

# **Entwicklung und Anwendung eines stoffstrombasierten Optimierungswerkzeugs zur Ablaufplanung von Rückbauprojekten**

Zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften  
(Dr.-Ing.)

von der KIT-Fakultät für Wirtschaftswissenschaften  
des Karlsruher Instituts für Technologie (KIT)

genehmigte  
Dissertation

von  
Marco Gehring, M.Sc.

Tag der mündlichen Prüfung: 11. Mai 2023  
Referent: Prof. Dr. Frank Schultmann  
Korreferent: Prof. Dr. Stefan Nickel



# Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als Wissenschaftlicher Mitarbeiter am Institut für Industriebetriebslehre und Industrielle Produktion (IIP) des Karlsruher Instituts für Technologie (KIT) in den Jahren von 2019 bis 2023. Insbesondere die Erkenntnisse aus dem Forschungsvorhaben „Entwicklung eines benutzerfreundlichen kostenoptimierenden Planungswerkzeugs für kerntechnische Rückbauprojekte unter der Berücksichtigung von Stoffströmen zur Ressourcenplanung“ (NukPlaRStoR, Förderkennzeichen 15S9414A) flossen in die Erstellung meiner Dissertation ein. Dieses Vorwort möchte ich dazu nutzen, allen Personen zu danken, die mich bei meinem Promotionsvorhaben unterstützt haben.

Mein besonderer Dank gilt Prof. Dr. Frank Schultmann für die Betreuung meines Promotionsvorhabens und für die Schaffung einer stets sehr angenehmen Arbeitsatmosphäre am IIP. Bei Prof. Dr. Stefan Nickel bedanke ich mich für die Übernahme des Korreferats. Den weiteren Mitgliedern des Prüfungskollegiums, Prof. Dr. Marc Wouters und Prof. Dr. Orestis Terzidis, möchte ich ebenfalls meinen Dank aussprechen.

Dem Bundesministerium für Bildung und Forschung bin ich für die finanzielle Förderung des oben genannten Forschungsvorhabens zu Dank verpflichtet. Ein großes Dankeschön richtet sich an meine Gruppenleiterin Dr. Rebekka Volk für die Einwerbung der Fördermittel und die fachliche Unterstützung bei meinem Promotionsvorhaben, welche stets in idealer Ausgewogenheit zwischen Anleitung und Freiheit bei der thematischen Ausgestaltung erfolgte. Danken möchte ich außerdem Dr. Felix Hübner, der einen wesentlichen Beitrag zur Konzeption des Forschungsvorhabens leistete. Meinem Kollegen Niklas Braun danke ich für die Durchsicht meiner Dissertation und die vielen inspirierenden Diskussionen. Meinen Gruppenkolleginnen und -kollegen – darunter insbesondere meinen Bürokollegen Christoph Stallkamp und Justus Steins – danke ich herzlich für die konstruktiven und unterhaltsamen Gespräche, die den Arbeitsalltag auch in herausfordernden Phasen sehr erleichtert haben. Selbiges gilt für Dr. Florian Diehlmann und Dr. Markus Lüttenberg, deren Büro eine geschätzte Anlaufstelle für Fragen und Diskussionen war. Mein Dank gilt darüber hinaus allen Mitgliedern des IIP für den außergewöhnlich kollegialen Umgang miteinander – die Begegnungen auf dem Flur, in der Kaffeeküche oder bei Doktorandenseminaren haben dazu beigetragen, dass ich mich gerne an meine Zeit am IIP zurückerinnern werde.

Herzlich danken möchte ich meinen Eltern, die mir meine Hochschulausbildung ermöglichten und mich jederzeit dabei unterstützt haben, den eingeschlagenen Weg weiterzugehen.

Karlsruhe, im Juni 2023  
Marco Gehring



# Kurzfassung

Um das umweltpolitische Ziel des Übergangs zu einer Kreislaufwirtschaft zu erreichen, wird im Bausektor statt eines konventionellen Abbruchs ein selektiver Rückbau angestrebt. Dabei wird eine Vermischung der bei der Demontage anfallenden Stoffe so weit wie möglich vermieden. Zur Erhöhung der Recyclingquoten ist darüber hinaus eine Aufbereitung der Stoffe erforderlich. Die Aufbereitung und der Transport (zusammengefasst als Verarbeitung bezeichnet) sowie die Lagerung der Stoffe auf der Baustelle unterliegen Kapazitätsbeschränkungen. Werden diese stoffstrombezogenen Kapazitätsbeschränkungen bei der Durchführung eines Rückbauprojekts überschritten, muss der Rückbau pausiert oder kurzfristig umgeplant werden. Um dies zu vermeiden, wird in dieser Arbeit ein Optimierungswerkzeug entwickelt, mit welchem die stoffstrombezogenen Kapazitätsbeschränkungen bei der Ablaufplanung eines Rückbauprojekts im Voraus berücksichtigt werden können. Damit soll Projektverantwortlichen die Erarbeitung eines realistischen, auf die Kapazitäten der Stoffverarbeitung abgestimmten Ablaufplans vereinfacht werden. Darüber hinaus soll das Optimierungswerkzeug Entscheidungen über die Beschaffung von Anlagen oder die Dimensionierung von Lagerflächen unterstützen. Der Rückbau kern-technischer Anlagen stellt aufgrund der Projektkomplexität und der mit den Stoffströmen verbundenen hohen Sicherheitsanforderungen ein geeignetes Anwendungsbeispiel dar.

Das Optimierungswerkzeug baut auf quantitativen Modellen und Methoden aus dem Forschungszweig der Projektablaufplanung (engl. project scheduling), einem Teilgebiet des Operations Research, auf. Zunächst wird dazu ein sogenanntes Schedulingproblem formuliert, welches die Suche nach Startzeitpunkten für (Projekt-)Vorgänge beschreibt, so dass die Projektdauer minimiert wird. Die Einhaltung von Vorrangbeziehungen zwischen den Vorgängen, Kapazitäten erneuerbarer Ressourcen (z. B. Personal, Maschinen) sowie Verarbeitungs- und Lagerkapazitäten bilden die Restriktionen des Schedulingproblems. Die Berücksichtigung der Verarbeitungs- und Lagerkapazitäten, welche den Rückbau verzögern können, stellt eine wesentliche Erweiterung gegenüber existierenden Modellen dar. Insbesondere aufgrund der Lager kann bereits der Rechenaufwand zur Bestimmung einer zulässigen Lösung exponentiell ansteigen. Daher werden vier neuartige heuristische Lösungsverfahren entwickelt, welche jeweils für bestimmte Arten von Probleminstanzen geeignet sind. Die Constraint- und die lineare Programmierung werden ebenfalls zur Lösung des Problems eingesetzt. Umfangreiche Rechenexperimente mit Testinstanzen zeigen, dass die neu entwickelten Lösungsverfahren und die Constraintprogrammierung geeignet sind, um für stoffstromintensive Projekte innerhalb praktikabler Rechenzeiten hochwertige Ablaufpläne zu berechnen. Es werden Empfehlungen erarbeitet, unter welchen Umständen welches konkrete Verfahren zum Einsatz kommen sollte.



# Abstract

To achieve the environmental policy goal of transitioning to a circular economy, selective dismantling is preferred in the construction sector instead of conventional demolition. That is, mixing of materials released during dismantling is avoided as much as possible. In addition, conditioning the materials is necessary to increase recycling rates. Conditioning and transporting (jointly referred to as processing) as well as storing materials at the construction site are subject to capacity restrictions. If these material flow-related capacity restrictions are exceeded during the execution of a dismantling project, the dismantling must be interrupted or rescheduled at short notice. To avoid this, an optimization tool is developed in this thesis, which allows for taking into account material flow-related capacity restrictions during the planning phase of a dismantling project. This should assist project managers with finding a realistic schedule that is compatible with capacities in material processing. In addition, the optimization tool aims at supporting decisions on procuring machines or sizing storage spaces. Dismantling nuclear facilities is a suitable application case due to the project complexity and the high safety requirements associated with the material flows.

The optimization tool is based on quantitative models and methods from the research field of project scheduling, a subdiscipline of Operations Research. First, a scheduling problem is formulated, which can be summarized as finding start times for (project) activities so that the project makespan is minimized. The constraints of the scheduling problem state that precedence relations between activities, capacities of renewable resources (e.g., staff, machines) as well as processing and storage capacities must be satisfied. Taking into account processing and storage capacities, which can delay the dismantling project, is a significant enhancement over existing models. In particular, due to the storage capacities, the computational effort to find a feasible solution can already increase exponentially. Therefore, four novel heuristic solution methods are developed, each suitable for specific types of problem instances. Constraint and linear programming are also used to solve the problem. Extensive computational experiments with test instances show that the novel solution methods and constraint programming are suitable for computing high-quality schedules for material flow-intensive projects within reasonable computation times. Recommendations are given regarding which specific method should be used under which circumstances.





# Inhaltsverzeichnis

<b>Vorwort</b>	<b>iii</b>
<b>Kurzfassung</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Abbildungsverzeichnis</b>	<b>xii</b>
<b>Tabellenverzeichnis</b>	<b>xv</b>
<b>Abkürzungs- und Symbolverzeichnis</b>	<b>xvii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Ausgangslage und Problemstellung . . . . .	1
1.2 Zielsetzung und Lösungsweg . . . . .	4
<b>2 Methodische Grundlagen</b>	<b>9</b>
2.1 Begriffe . . . . .	9
2.1.1 Projektplanung und Scheduling . . . . .	9
2.1.2 Stoffströme und Materialflüsse . . . . .	11
2.1.3 Probleme, Verfahren und Modelle . . . . .	12
2.2 Schedulingprobleme . . . . .	14
2.2.1 Allgemeines Schedulingproblem . . . . .	14
2.2.2 Scheduling unter Vorrangrestriktionen . . . . .	17
2.2.3 Scheduling unter Zeitrestriktionen . . . . .	19
2.2.4 Scheduling unter (erneuerbaren) Ressourcenrestriktionen . . . . .	22
2.2.5 Mehr-Modus-Scheduling . . . . .	26
2.2.6 Scheduling unter erneuerbaren und kumulativen Ressourcenrestriktionen . . . . .	29
2.2.7 Weitere Modelle und Restriktionen . . . . .	34
2.2.8 Zielsetzungen . . . . .	35
2.3 Lösungsverfahren . . . . .	38
2.3.1 Generierungsschemas . . . . .	39
2.3.2 Prioritätsregelbasierte Verfahren . . . . .	42
2.3.3 Metaheuristiken . . . . .	43
2.4 Stoffstrombasiertes Produktionsmanagement . . . . .	44
2.4.1 Begriffliche Abgrenzung . . . . .	44
2.4.2 Zusammenhang zwischen Projektplanung und Produktionsplanung . . . . .	45

2.4.3	Graphentheoretische Modelle aus dem Produktionsmanagement . . .	46
2.4.4	Vergenzen von Produktionsstrukturen und Stoffströmen . . . . .	50
<b>3</b>	<b>Stoffstrombasierte Planung von Rückbauprojekten</b>	<b>53</b>
3.1	Problemstellung . . . . .	53
3.2	Der Rückbau kerntechnischer Anlagen als Anwendungsbeispiel . . . . .	57
3.2.1	Freisetzung von Stoffströmen . . . . .	58
3.2.2	Verarbeitung von Stoffströmen . . . . .	60
3.2.3	Eignung zum Einsatz eines Optimierungswerkzeugs . . . . .	62
3.3	Modellanforderungen, -einschränkungen und -abgrenzungen . . . . .	63
3.4	Das Schedulingproblem unter Ressourcen- und Stoffstromrestriktionen (RM-CPSP) . . . . .	66
3.4.1	Stoffstrommodell . . . . .	66
3.4.2	Konzeptionelle Problemformulierung . . . . .	68
3.4.3	Modellierungs- und Lösungsansätze . . . . .	72
3.5	Existierende Ansätze in der Literatur . . . . .	75
3.5.1	Stoffstrommodell . . . . .	75
3.5.2	Schedulingproblem . . . . .	78
<b>4</b>	<b>Integrierte Planung von Rückbauprojekt und Stoffströmen</b>	<b>83</b>
4.1	Das Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen (RCPSP/M) . . . . .	84
4.2	Transformationsschemas . . . . .	91
4.2.1	Granulares Transformationsschema . . . . .	92
4.2.2	Aggregiertes Transformationsschema . . . . .	99
4.2.3	Strukturelle Eigenschaften von granularen und aggregierten Instanzen	108
4.2.4	Auswirkungen der Aggregation auf die Realitätstreue . . . . .	110
4.2.5	Weitere Transformationsschemas . . . . .	116
4.3	Problemcharakteristika . . . . .	119
4.3.1	Notwendigkeit der stoffstrominduzierten Zeitstruktur . . . . .	120
4.3.2	Kritischer Pfad . . . . .	122
4.4	Modellierungsaspekte . . . . .	124
4.4.1	Ein- und Auslagerungen in bzw. aus demselben Lager . . . . .	124
4.4.2	No-Wait-Stoffverarbeitung . . . . .	126
4.4.3	Mindestlagerdauern und Transportzeiten . . . . .	129
4.4.4	Festlegung der Stoffeinheiten . . . . .	129
4.5	Modellerweiterungen . . . . .	130
4.5.1	Zeitliche Höchstabstände und Höchstlagerdauern . . . . .	131
4.5.2	Zeitlich variable Ressourcenkapazitäten . . . . .	131
4.5.3	Homogene Stoffströme und Mindestbestände . . . . .	134
4.6	Überblick und Zusammenfassung . . . . .	135
<b>5</b>	<b>Lösungsverfahren</b>	<b>137</b>
5.1	Grundlagen zur Lösbarkeit von Schedulingproblemen mit kumulativen Ressourcen . . . . .	137

5.2	Existierende Ansätze in der Literatur . . . . .	140
5.2.1	Übersicht . . . . .	140
5.2.2	Branch-and-Bound-Verfahren von Neumann u. a. . . . .	144
5.2.3	Prioritätsregelbasiertes Verfahren von Schwindt u. a. . . . .	149
5.2.4	Übertragbarkeit der Ansätze zur Lösung des (RCPSP/M) . . . . .	151
5.3	Lösungsverfahren für granulare Instanzen des (RCPSP/M) . . . . .	153
5.3.1	Grundlegende Konzepte . . . . .	153
5.3.2	Generierungsschema nach einem Dekompositionsansatz . . . . .	154
5.4	Lösungsverfahren für aggregierte Instanzen des (RCPSP/M) . . . . .	163
5.4.1	Grundlegende Konzepte . . . . .	163
5.4.2	Generierungsschema nach einem Dekompositionsansatz . . . . .	168
5.4.3	Generierungsschema nach einem Integrationsansatz . . . . .	175
5.5	Lösungsverfahren für beliebige Instanzen des (RCPSP/M) . . . . .	183
5.5.1	Grundlegende Konzepte . . . . .	183
5.5.2	Generierungsschema nach einem Integrationsansatz . . . . .	187
5.6	Vergleich der Lösungsverfahren . . . . .	194
5.7	Erweiterungen der Lösungsverfahren . . . . .	197
5.8	Lösung mittels Constraintprogrammierung . . . . .	199
5.9	Lösung mittels linearer Programmierung . . . . .	202
5.10	Überblick und Zusammenfassung . . . . .	206
<b>6</b>	<b>Rechenexperimente</b> . . . . .	<b>209</b>
6.1	Testinstanzen . . . . .	209
6.1.1	Instanzparameter . . . . .	210
6.1.2	Verfahren zur Erzeugung von Testinstanzen . . . . .	211
6.1.3	Untere Schranken . . . . .	215
6.2	Testaufbau . . . . .	217
6.2.1	Verwendete Metaheuristiken . . . . .	218
6.2.2	Verwendete Testinstanzen . . . . .	224
6.2.3	Verwendete Lösungsstrategien . . . . .	226
6.3	Experimente mit Fokus auf den Stoffstrombereich . . . . .	229
6.3.1	Ablauf und Ergebnisübersicht . . . . .	229
6.3.2	Auswertung hinsichtlich der Transformationsschemas . . . . .	234
6.3.3	Auswertung hinsichtlich der Lösungsverfahren . . . . .	237
6.3.4	Auswertung hinsichtlich der Metaheuristiken . . . . .	244
6.3.5	Auswertung hinsichtlich der Freisetzungs- bzw. Nutzungsart . . . . .	245
6.3.6	Langzeitexperimente . . . . .	247
6.4	Experimente mit Fokus auf den Rückbaubereich . . . . .	248
6.4.1	Ablauf und Ergebnisübersicht . . . . .	249
6.4.2	Auswertung . . . . .	249
6.5	Experimente mit Fokus auf die Zielsetzung . . . . .	251
6.5.1	Ablauf und Ergebnisübersicht . . . . .	252
6.5.2	Auswertung . . . . .	252

<b>7 Zusammenfassung und Ausblick</b>	<b>257</b>
7.1 Zusammenfassung . . . . .	257
7.2 Kritische Würdigung und Ausblick . . . . .	261
<b>Literatur</b>	<b>265</b>

# Abbildungsverzeichnis

2.1	Beispiel für einen Vorgangsknotennetzplan mit zeitlichen Höchst- und Mindestabständen. . . . .	21
2.2	Gantt-Diagramme und Ressourcenprofile zu Beispiel 2.1. . . . .	32
2.3	Beispiel für einen Input-Output-Graphen (nach Fröhling 2006, S. 61). . . .	47
2.4	Beispiel für einen Gozintographen (nach Bloech u. a. 2014, S. 157). . . . .	48
2.5	Beispiel für ein Petri-Netz. . . . .	49
2.6	Darstellung a) eines Input-Output-Graphen zur Versorgung der Produktionsstellen 2 und 3 als b) State-Task-Netz mit verschiedenartigen Stoffströmen und als c) State-Task-Netz mit gleichartigen Stoffströmen (nach Kondili u. a. 1993, S. 213f). . . . .	50
2.7	Verschiedene Input-Output-Graphen mit Angabe des Vergenztyps der abgebildeten Produktionsstruktur (und der damit einhergehenden Stoffströme). . . . .	51
3.1	Gantt-Diagramm und erneuerbares Ressourcenprofil zu einem Projekt mit unbegrenzten erneuerbaren Ressourcenkapazitäten; Lösung des (TPSP). . . .	54
3.2	Gantt-Diagramm und erneuerbares Ressourcenprofil zu einem Projekt mit begrenzten erneuerbaren Ressourcenkapazitäten; Lösung des (RCPSp). . . .	55
3.3	Gantt-Diagramm und Auslastungsprofil zu einem Projekt mit unbegrenzten Lagerkapazitäten; Lösung des (RCPSp). . . . .	56
3.4	Gantt-Diagramm und Auslastungsprofil zu einem Projekt mit begrenzten Lagerkapazitäten; Lösung des in diesem Kapitel vorgestellten (RMCPSP). . .	57
3.5	Flussnetzwerk zu Beispiel 3.1. . . . .	69
3.6	Gantt-Diagramme und Auslastungsprofile: a) bei gleichmäßiger Stofffreisetzung während der Vorgangsausführung wie in dieser Arbeit; b) bei kontinuierlichen Stoffströmen nach Neumann, Schwindt und Trautmann (2005). . .	81
4.1	Gantt-Diagramme und Verlauf der kumulativen Ressourcennutzungen zu Beispiel 4.1. . . . .	89
4.2	Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.2. . . . .	98
4.3	Exemplarischer Vorgangsknotennetzplan einer Instanz $\overline{I}_{RCPSp/M-gra}$ ; Rückbauvorgang 1 und Stoffstromvorgänge $\overline{Succ}^o(1)$ entstammen dem Beispiel 4.2. . . . .	99
4.4	Exemplarischer Vorgangsknotennetzplan einer teilweise aggregierten Instanz $\overline{I}_{RCPSp/M-agg}$ (vgl. Abbildung 4.3). . . . .	102
4.5	Exemplarischer Vorgangsknotennetzplan einer vollständig aggregierten Instanz $\overline{I}_{RCPSp/M-agg}$ (vgl. Abbildung 4.4). . . . .	103

4.6	Gantt-Diagramme zu Beispiel 4.3: a) zeitliche Mindestabstände zwischen granularen Stoffstromvorgängen und ihrem Vorgänger; b) zeitlicher Mindestabstand zwischen aggregiertem Stoffstromvorgang und seinem Vorgänger. . . . .	106
4.7	Gantt-Diagramm zu Beispiel 4.3: Zeitliche Mindestabstände zwischen einem aggregierten Stoffstromvorgang und seinen Nachfolgern. . . . .	107
4.8	Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.3. . . . .	108
4.9	Exemplarischer Vorgangsknotennetzplan einer Instanz $\bar{I}_{\text{RCPSP}/\text{M-agg}}$ ; Rückbauvorgang 1 und Stoffstromvorgänge $\overline{\text{Succ}}^0(1)$ entstammen dem Beispiel 4.3. . . . .	109
4.10	Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.4: a) granulare Instanz; b) aggregierte Instanz. . . . .	112
4.11	Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.5: a) granulare Instanz; b) aggregierte Instanz. . . . .	113
4.12	Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.6: a) granulare Instanz; b) aggregierte Instanz. . . . .	114
4.13	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 4.7: a) granulare Instanz; b) aggregierte Instanz. . . . .	116
4.14	Gantt-Diagramme und erneuerbare Ressourcenprofile zu Beispiel 4.8: a) granulare Instanz; b) horizontal aggregierte Instanz. . . . .	118
4.15	Gantt-Diagramme und erneuerbare Ressourcenprofile zu Beispiel 4.9: a) granulare Instanz; b) vertikal aggregierte Instanz. . . . .	119
4.16	Vorgangsknotennetzplan, Gantt-Diagramm und kumulatives Ressourcenprofil zu Beispiel 4.10 mit stoffstrominduzierter Zeitstruktur. . . . .	121
4.17	Vorgangsknotennetzplan, Gantt-Diagramm und kumulatives Ressourcenprofil zu Beispiel 4.10 ohne stoffstrominduzierte Zeitstruktur. . . . .	122
4.18	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 4.13: a) Einlagerung zum Start und Auslagerung zum Ende; b) Einlagerung zum Ende und Auslagerung zum Start; c) gleichmäßige Ein- und Auslagerung während der Ausführung. . . . .	127
4.19	Überblick über die Inhalte der Kapitel 3 und 4. . . . .	135
5.1	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.1: a) nach Einplanung von Vorgang 1; b) nach Einplanung von Vorgang 2. . . . .	139
5.2	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.2 bei Nutzung zum Vorgangsstart bzw. -ende: a) initiale Ressourcenrelaxation; b) nach Auflösung des Konflikts. . . . .	146
5.3	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.3 bei gleichmäßiger Nutzung während der Vorgangsausführung: a) initiale Ressourcenrelaxation; b) nach Auflösung des Konflikts. . . . .	149
5.4	Flussdiagramm für SubScheduleGra. . . . .	158
5.5	Vorgangsknotennetzplan zu Beispiel 5.6. . . . .	161
5.6	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.6. . . . .	161
5.7	Vorgangsknotennetzplan zu Beispiel 5.7. . . . .	166

5.8	Vorgangsknotennetzplan zu Beispiel 5.8. . . . .	171
5.9	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.8. . . .	173
5.10	Vorgangsknotennetzplan zu Beispiel 5.9. . . . .	178
5.11	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.9. . . .	181
5.12	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.10. . . .	182
5.13	Gantt-Diagramme und Verlauf der kumulativen Ressourcennutzungen zu Beispiel 5.11. . . . .	188
5.14	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.12. . . .	192
5.15	Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.13 (Ver- anschaulichung der Unterschiede im kumulativen Ressourcenprofil der ori- ginalen Instanz und der Stufenvariante): a) granulare Instanz; b) aggregierte Instanz. . . . .	203
5.16	Überblick über die Inhalte der Kapitel 3, 4 und 5. . . . .	208
6.1	Veranschaulichung der betrachteten Lösungsstrategien. . . . .	228

# Tabellenverzeichnis

3.1	Eingeführte Notation zur Formulierung des (RMCPSP). . . . .	73
4.1	Eingeführte Notation zur Formulierung des (RCPSP/M). . . . .	91
5.1	Iterationen in Beispiel 5.8 (It.: Iteration). . . . .	172
5.2	Iterationen in Beispiel 5.9 (It.: Iteration). . . . .	180
5.3	Iterationen in Beispiel 5.12 (It.: Iteration). . . . .	193
5.4	Vergleich der vorgestellten Lösungsverfahren. . . . .	195
6.1	Wertebereiche der Instanzparameter. . . . .	224
6.2	Experimente mit Fokus auf den Stoffstrombereich: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%]. . . . .	232
6.3	Experimente mit Fokus auf den Stoffstrombereich: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%]. . . . .	233
6.4	Experimente mit Fokus auf den Stoffstrombereich: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%]; modellbereinigter Vergleich. . . . .	243
6.5	Experimente mit Fokus auf den Stoffstrombereich: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%]; modellbereinigter Vergleich. . . . .	243
6.6	Experimente mit Fokus auf den Rückbaubereich: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%]. . . . .	250
6.7	Experimente mit Fokus auf den Rückbaubereich: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%]. . . . .	250
6.8	Experimente mit Fokus auf die Zielfunktion: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%]. . . . .	253
6.9	Experimente mit Fokus auf die Zielfunktion: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%]. . . . .	253



# Abkürzungs- und Symbolverzeichnis

## Abkürzungen

AtG	Atomgesetz
BMUB	Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit
CS	Kaltstart (engl. cold start)
LB	untere Schranke (engl. lower bound)
Mg	Megagramm
MILP	gemischt-ganzzahliges lineares Programm (engl. mixed-integer linear program)
MLS	Multi-Start-Local-Search (-Metaheuristik)
MRCPSP	Mehr-Modus-Schedulingproblem
MS	Multi-Start (-Metaheuristik)
NP	nichtdeterministische Polynomialzeit
OR	Operations Research
PSP	allgemeines Schedulingproblem (engl. project scheduling problem)
RCPSP	Schedulingproblem unter erneuerbaren Ressourcenrestriktionen (engl. resource-constrained project scheduling problem)
RCPSP/c	Schedulingproblem unter erneuerbaren und kumulativen Ressourcenrestriktionen
RCPSP/M	Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen
RCPSP/max	Schedulingproblem unter erneuerbaren Ressourcenrestriktionen mit Zeitbeziehungen anstelle von Vorrangbeziehungen
RMCPSP	Schedulingproblem unter Ressourcen- und Stoffstromrestriktionen (engl. resource- and material-flow-constrained project scheduling problem)
SA	Simulated-Annealing (-Metaheuristik)
SGS	Generierungsschema (engl. schedule generation scheme)
StrlSchV	Strahlenschutzverordnung
TPSP	Schedulingproblem unter Vorrangrestriktionen (engl. temporal project scheduling problem)
TPSP/max	Schedulingproblem unter Zeitrestriktionen
u. d. R.	unter den Restriktionen
UBA	Umweltbundesamt
WS	Warmstart (engl. warm start)

## Symbole

### Verschiedene

$\square$	Ende eines Beispiels
$:=$	gleich per Definition oder Zuweisung eines Wertes
$\equiv$	identisch, d. h. für alle Variablenbelegungen gleich
$\cup$	Vereinigung von Mengen
$\cap$	Schnitt von Mengen
$\times$	kartesisches Produkt
$ $	Eigenschaft der Elemente einer Menge
$\forall$	Allquantor („für alle“)
$\exists$	Existenzquantor („es existiert ein“)
$\emptyset$	leere Menge
$()$	leere Folge
$ x $	Betrag von $x$ bzw. Kardinalität der Menge $x$ (d. h. Anzahl der Elemente in der Menge $X$ )
$[x]^+$	Maximum von 0 und $x$
$\lfloor x \rfloor$	abgerundeter Wert von $x$ (Abrundungsfunktion)
$\lceil x \rceil$	aufgerundeter Wert von $x$ (Aufrundungsfunktion)
$[x_1, x_2)$	Intervall von $x_1$ (inklusive) bis $x_2$ (exklusive)
$\text{rnd}()$	Zufallszahl im Intervall $[0, 1)$
$\text{round}(x)$	auf eine ganze Zahl gerundeter Wert von $x$
$\text{length}(\ell)$	Anzahl der Vorgänge in der Vorgangsfolge $\ell$
$\text{upstream}(p)$	Menge aller dem Verarbeitungsschritt $p$ vorgeschalteten Lager
$\text{units}(p)$	Anzahl der Stoffeinheiten, die Verarbeitungsschritt $p$ während des gesamten Rückbauprojekts durchlaufen

### Zahlen

0	fiktiver Startvorgang
1, 2, 3, ...	exemplarische (Rückbau-)Vorgänge (im entsprechenden Zusammenhang)
$1^o, 2^o, 3^o, \dots$	exemplarische Stoffstromvorgänge
$\hat{1}^o, \hat{2}^o, \hat{3}^o, \dots$	exemplarische aggregierte Stoffstromvorgänge
$1^o[\text{P1}]$	ein exemplarischer Stoffstromvorgang, der von Verarbeitungsschritt P1 abgeleitet wurde

### Lateinische Buchstaben

$A$	Teilmenge der (Rückbau-)Vorgänge
$\mathcal{A}^\alpha(S, t)$	Menge der zum Zeitpunkt $t$ bei gegebenem Schedule $S$ hinsichtlich der erneuerbaren Ressourcen aktiven (Rückbau-)Vorgänge

---

$\mathcal{A}^\alpha(S, M, t)$	Menge der zum Zeitpunkt $t$ bei gegebenem Schedule $S$ und gegebener Moduszuweisung $M$ hinsichtlich der erneuerbaren Ressourcen aktiven (Rückbau-)Vorgänge
$\mathcal{A}^\alpha(S^o, t)$	Menge der zum Zeitpunkt $t$ bei gegebenem Schedule $S^o$ hinsichtlich der erneuerbaren Ressourcen aktiven Stoffstromvorgänge
$\mathcal{A}^\beta(S, t)$	Menge der zum Zeitpunkt $t$ bei gegebenem Schedule $S$ hinsichtlich der nichterneuerbaren Ressourcen aktiven (Rückbau-)Vorgänge
$\mathcal{A}^V(S, t)$	Menge der zum Zeitpunkt $t$ bei gegebenem Schedule $S$ hinsichtlich der kumulativen Ressourcen aktiven (Rückbau-)Vorgänge
$\mathcal{A}^V(S^o, t)$	Menge der zum Zeitpunkt $t$ bei gegebenem Schedule $S^o$ hinsichtlich der kumulativen Ressourcen aktiven Stoffstromvorgänge
$act[i]$	Intervallvariable zum Rückbauvorgang $i$ im Constraintprogramm
$act[j]$	Intervallvariable zum Stoffstromvorgang $j$ im Constraintprogramm
$B$	Teilmenge der (Rückbau-)Vorgänge
$C1, C2, \dots$	exemplarische kumulative Ressourcen
$c_i$	mit dem Start des Vorgangs $i$ ausgelöster Zahlungsstrom
$c_k$	Kostenfaktor der Ressource $k$
$DUR$	Instanzzparameter (siehe Abschnitt 6.1.1)
$\bar{d}$	maximale Projektdauer
$d_i$	Dauer des (Rückbau-)Vorgangs $i$
$\bar{d}_i$	Fälligkeitstermin des Vorgangs $i$
$d_{im}$	Dauer des Vorgangs $i$ im Modus $m$
$d_j$	Dauer des Stoffstromvorgangs $j$
$d_p$	Dauer des Verarbeitungsschritts $p$ zur Verarbeitung einer Stoffeinheit
$d_{ii'}$	Länge des längsten Pfades zwischen den (Rückbau-)Vorgängen $i$ und $i'$
$d_{ij}$	Länge des längsten Pfades zwischen Rückbauvorgang $i$ und Stoffstromvorgang $j$
$d_{jj'}$	Länge des längsten Pfades zwischen den Stoffstromvorgängen $j$ und $j'$
$d_{jj'}^{flow}$	zeitlicher Mindestabstand zur stoffstrominduzierten Zeitbeziehung zwischen den Stoffstromvorgängen $j$ und $j'$
$d^{flow}$	Matrix aller zeitlichen Mindestabstände zu den stoffstrominduzierten Zeitbeziehungen
$d_{ii'}^{min}$	zeitlicher Mindestabstand zwischen Vorgang $i$ und Vorgang $i'$
$d_{ii'}^{max}$	zeitlicher Höchstabstand zwischen Vorgang $i$ und Vorgang $i'$
$d_{ij}^{rel}$	zeitlicher Mindestabstand zur Freisetzungsbeziehung zwischen Rückbauvorgang $i$ und Stoffstromvorgang $j$
$d^{rel}$	Matrix aller zeitlichen Mindestabstände zu den Freisetzungsbeziehungen
$dur_x$	Ausprägung des Instanzparameters $DUR$
$E$	Menge der Vorrangbeziehungen
$\mathcal{E}$	Menge der einplanbaren (Rückbau-)Vorgänge
$E^{flow}$	Menge der stoffstrominduzierten Zeitbeziehungen
$E^{rel}$	Menge der Freisetzungsbeziehungen
$E^{temp}$	Menge der Zeitbeziehungen
$ES_i$	frühestmöglicher Startzeitpunkt des (Rückbau-)Vorgangs $i$

$ES_j$	frühestmöglicher Startzeitpunkt des Stoffstromvorgangs $j$
ext	Platzhalter für max oder min
$f_{iw}^Y$	Anzahl der Stoffeinheiten, die Rückbauvorgang $i$ freisetzt und die Stoffstrompfad $w$ durchlaufen (= Stofffreisetzung)
$f_{iw}^{Y,start}$	Anzahl der Stoffeinheiten, die Rückbauvorgang $i$ zu seinem Start freisetzt und die Stoffstrompfad $w$ durchlaufen
$f_{iw}^{Y,lin}$	Anzahl der Stoffeinheiten, die Rückbauvorgang $i$ gleichmäßig während seiner Ausführung freisetzt und die Stoffstrompfad $w$ durchlaufen
$f_{iw}^{Y,end}$	Anzahl der Stoffeinheiten, die Rückbauvorgang $i$ zu seinem Ende freisetzt und die Stoffstrompfad $w$ durchlaufen
$f_{iw}^Y(S_i, t)$	Anzahl der durch Rückbauvorgang $i$ zum Zeitpunkt $t$ bei gegebenem Startzeitpunkt $S_i$ freigesetzten Stoffeinheiten, die Stoffstrompfad $w$ durchlaufen
$f_{ik}^Y(S_i, t)$	Anzahl der durch Rückbauvorgang $i$ zum Zeitpunkt $t$ bei gegebenem Startzeitpunkt $S_i$ in Lager bzw. kumulative Ressource $k$ freigesetzten Stoffeinheiten
$f_{ik}^Y(\cdot, \infty)$	gesamte Anzahl der durch Rückbauvorgang $i$ in Lager bzw. kumulative Ressource $k$ freigesetzten Stoffeinheiten
$f_k^Y(S, t)$	Anzahl der durch alle Rückbauvorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S$ in Lager bzw. kumulative Ressource $k$ freigesetzten Stoffeinheiten
$I$	Anzahl der (Rückbau-)Vorgänge
$I + 1$	fiktiver End-(Rückbau-)Vorgang
$\mathcal{I}_x$	Instanz des Schedulingproblems $x$
$INV$	Instanzparameter (siehe Abschnitt 6.1.1)
$i, i', i_1, \dots$	Vorgänge (im Grundlagenkapitel) bzw. Rückbauvorgänge
$i^*$	der in einer Iteration einzuplanende (Rückbau-)Vorgang
$(J + 1)^o$	fiktiver End-Stoffstromvorgang
$j, j', j_1, \dots$	Stoffstromvorgänge
$k, k', k_1, \dots$	Ressourcen bzw. Lager
$k_x(w)$	$x$ -tes Lager in Stoffstrompfad $w$
$L1, L2, \dots$	exemplarische Lager
$LB_{PSPLIB}$	PSPLIB-basierte untere Schranke für den Zielfunktionswert
$LB_{material}$	stoffstrombasierte untere Schranke für den Zielfunktionswert
$LS_i$	spätestmöglicher Startzeitpunkt des (Rückbau-)Vorgangs $i$
$LS_j$	spätestmöglicher Startzeitpunkt des Stoffstromvorgangs $j$
$l, l', l_1, \dots$	Vorgänge, die sowohl Rückbau- als auch Stoffstromvorgänge sein können
$l^+$	relevanter Vorgänger
$l^*$	der in einer Iteration einzuplanende Vorgang
$\ell$	Vorgangsfolge
$M_i$	Moduszuweisung des Vorgangs $i$ (im Grundlagenkapitel) oder Stoffstromstruktur des Rückbauvorgangs $i$
$M$	Vektor aller Moduszuweisungen (= Moduszuweisung)
$\mathcal{M}_i$	Modusmenge des Vorgangs $i$

---

$MA1, \dots$	exemplarische erneuerbare Ressourcen (zur Abgrenzung von PSPLIB-Ressourcen)
$m, m_0, \dots$	Modi
$N$	Vorgangsknotennetzplan mit Vorrangbeziehungen
$N^{temp}$	Vorgangsknotennetzplan mit Zeitbeziehungen
$N^{temp+}$	um die Kante $(I + 1, 0)$ ergänzter Vorgangsknotennetzplan $N^{temp}$
$N1, N2, \dots$	exemplarische nichterneuerbare Ressourcen
$NC$	Instanzparameter (siehe Abschnitt 6.1.1)
$NREL$	Instanzparameter (siehe Abschnitt 6.1.1)
$O(x)$	O-Notation zur Angabe der oberen Schranke $x$ für die Laufzeitkomplexität eines Algorithmus
$\mathcal{P}$	Menge der Verarbeitungsschritte
$P1, P2, \dots$	exemplarische Verarbeitungsschritte
$Pred(i)$	Menge der (Rückbau-)Vorgänge, die direkte Vorgänger des (Rückbau-)Vorgangs $i$ sind
$Pred(j)$	Menge der Rückbauvorgänge, die direkte Vorgänger des Stoffstromvorgangs $j$ sind
$Pred^{temp}(i)$	Menge der Vorgänge, die direkte Vorgänger des Vorgangs $i$ sind (bei Vorhandensein von Zeitbeziehungen)
$Pred^o(j)$	Menge der Stoffstromvorgänge, die direkte Vorgänger des Stoffstromvorgangs $j$ sind
$\overline{Pred}(i)$	Menge der (Rückbau-)Vorgänge, die transitive Vorgänger des (Rückbau-)Vorgangs $i$
$PREL$	Instanzparameter (siehe Abschnitt 6.1.1)
$p, p', p_1, \dots$	Verarbeitungsschritte
$p_x(w)$	$x$ -ter Verarbeitungsschritt in Stoffstrompfad $w$
$\mathbb{R}_{\geq 0}$	Menge der positiven reellen Zahlen inklusive null
$R1, R2, \dots$	exemplarische erneuerbare Ressourcen
$RF$	Instanzparameter (siehe Abschnitt 6.1.1)
$RFP$	Instanzparameter (siehe Abschnitt 6.1.1)
$RS$	Instanzparameter (siehe Abschnitt 6.1.1)
$\mathcal{R}^\alpha$	Menge der erneuerbaren Ressourcen
$R_k^\alpha$	Kapazität der erneuerbaren Ressource $k$
$r_{ik}^\alpha$	Nutzung der erneuerbaren Ressource $k$ durch (Rückbau-)Vorgang $i$
$r_{imk}^\alpha$	Nutzung der erneuerbaren Ressource $k$ durch Vorgang $i$ in Modus $m$
$r_{jk}^\alpha$	Nutzung der erneuerbaren Ressource $k$ durch Stoffstromvorgang $j$
$r_{pk}^\alpha$	Nutzung der erneuerbaren Ressource $k$ durch Verarbeitungsschritt $p$
$r_k^\alpha(S, t)$	Anzahl der durch alle (Rückbau-)Vorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S$ genutzten Einheiten der erneuerbaren Ressource $k$
$r_k^\alpha(S, M, t)$	Anzahl der durch alle Vorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S$ und gegebener Moduszuweisung $M$ genutzten Einheiten der erneuerbaren Ressource $k$
$r_k^\alpha(S^o, t)$	Anzahl der durch alle Stoffstromvorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S^o$ genutzten Einheiten der erneuerbaren Ressource $k$

$\mathcal{R}^\beta$	Menge der nichterneuerbaren Ressourcen
$R_k^\beta$	Kapazität der nichterneuerbaren Ressource $k$
$r_{imk}^\beta$	Nutzung der nichterneuerbaren Ressource $k$ durch Vorgang $i$ in Modus $m$
$r_k^\beta(S, M, t)$	Anzahl der durch alle Vorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S$ und gegebener Moduszuweisung $M$ genutzten Einheiten der nichterneuerbaren Ressource $k$
$\mathcal{R}^Y$	Menge der Lager bzw. kumulativen Ressourcen
$\mathcal{R}^Y(l)$	Menge aller kumulativen Ressourcen, in die Vorgang $l$ Stoffeinheiten freisetzt bzw. einlagert
$\mathcal{R}^{Y+}(l)$	Menge der kumulativen Ressourcen, die Vorgang $l$ positiv nutzt
$\mathcal{R}^{Y-}(l)$	Menge der kumulativen Ressourcen, die Vorgang $l$ negativ nutzt
$\bar{R}_k^Y \in \mathbb{Z}$	Kapazität (= Höchstbestand) des Lagers bzw. der kumulativen Ressource $k$
$R_k^Y \in \mathbb{Z}$	Kapazität (= Höchstbestand) des Lagers bzw. der kumulativen Ressource $k$ (falls kein expliziter Mindestbestand gegeben ist, wird auf die Überstreichung verzichtet)
$\underline{R}_k^Y \in \mathbb{Z}$	Mindestbestand des Lagers bzw. der kumulativen Ressource $k$
$r_{ik}^Y$	Nutzung der kumulativen Ressource $k$ durch Vorgang $i$
$r_{ik}^{Y,start}$	Nutzung der kumulativen Ressource $k$ durch Vorgang $i$ zu seinem Start
$r_{ik}^{Y,lin}$	gleichmäßige Nutzung der kumulativen Ressource $k$ durch Vorgang $i$ während seiner Ausführung
$r_{ik}^{Y,end}$	Nutzung der kumulativen Ressource $k$ durch Vorgang $i$ zu seinem Ende
$r_{ik}^Y(S_i, t)$	Anzahl der durch Vorgang $i$ zum Zeitpunkt $t$ bei gegebenem Startzeitpunkt $S_i$ genutzten Einheiten der kumulativen Ressource $k$
$r_{ik}^Y(\cdot, \infty)$	gesamte Anzahl der durch Vorgang $i$ genutzten Einheiten der kumulativen Ressource $k$
$r_{jk}^Y(S_j^o, t)$	Anzahl der durch Stoffstromvorgang $j$ zum Zeitpunkt $t$ bei gegebenem Startzeitpunkt $S_j^o$ genutzten Einheiten der kumulativen Ressource $k$
$r_{jk}^Y(\cdot, \infty)$	gesamte Anzahl der durch Stoffstromvorgang $j$ genutzten Einheiten der kumulativen Ressource $k$
$r_k^Y(S, t)$	Anzahl der durch alle Vorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S$ genutzten Einheiten der kumulativen Ressource $k$
$r_k^Y(S^o, t)$	Anzahl der durch alle Stoffstromvorgänge zum Zeitpunkt $t$ bei gegebenem Schedule $S^o$ genutzten Einheiten der kumulativen Ressource $k$
$r_{jk}^{Y+,start}$	Maximum von 0 und der Nutzung der kumulativen Ressource $k$ durch Stoffstromvorgang $j$ zu seinem Start
$r_{jk}^{Y-,start}$	Minimum von 0 und der Nutzung der kumulativen Ressource $k$ durch Stoffstromvorgang $j$ zu seinem Start
$r_{jk}^{Y+,lin}$	Maximum von 0 und der gleichmäßigen Nutzung der kumulativen Ressource $k$ durch Stoffstromvorgang $j$ während seiner Ausführung
$r_{jk}^{Y-,lin}$	Minimum von 0 und der gleichmäßigen Nutzung der kumulativen Ressource $k$ durch Stoffstromvorgang $j$ während seiner Ausführung

---

$r_{jk}^{Y+,end}$	Maximum von 0 und der Nutzung der kumulativen Ressource $k$ durch Stoffstromvorgang $j$ zu seinem Ende
$r_{jk}^{Y-,end}$	Minimum von 0 und der Nutzung der kumulativen Ressource $k$ durch Stoffstromvorgang $j$ zu seinem Ende
$r_{jk}^{Y+}(S_j^o, t)$	Anzahl der durch Stoffstromvorgang $j$ zum Zeitpunkt $t$ bei gegebenem Startzeitpunkt $S_j^o$ positiv genutzten Einheiten der kumulativen Ressource $k$
$r_{jk}^{Y-}(S_j^o, t)$	Anzahl der durch Stoffstromvorgang $j$ zum Zeitpunkt $t$ bei gegebenem Startzeitpunkt $S_j^o$ negativ genutzten Einheiten der kumulativen Ressource $k$
$r_{jk}^{Y+}(\cdot, \infty)$	gesamte Anzahl der durch Stoffstromvorgang $j$ positiv genutzten Einheiten der kumulativen Ressource $k$
$r_{jk}^{Y-}(\cdot, \infty)$	gesamte Anzahl der durch Stoffstromvorgang $j$ negativ genutzten Einheiten der kumulativen Ressource $k$
$r^+, r^-$	Hilfsvariablen für positive und negative Nutzungen
$S_i$	Startzeitpunkt des (Rückbau-)Vorgangs $i$
$S$	Vektor aller Startzeitpunkte der (Rückbau-)Vorgänge (= Schedule)
$S_{old}$	Hilfsvariable für einen zuvor berechneten Startzeitpunkt
$S_j^o$	Startzeitpunkt des Stoffstromvorgangs $j$
$S^o$	Vektor aller Startzeitpunkte der Stoffstromvorgänge
$(S, S^o)$	Schedule der Rückbau- und Stoffstromvorgänge
$S^M$	Schedule für alle Rückbau- und Stoffstromvorgänge einer Stoffstromstruktur
$\mathcal{S}$	Menge aller zulässigen Schedules
$Succ(i)$	Menge der (Rückbau-)Vorgänge, die direkte Nachfolger des (Rückbau-)Vorgangs $i$ sind
$Succ^{temp}(i)$	Menge der Vorgänge, die direkte Nachfolger des Vorgangs $i$ sind (bei Vorhandensein von Zeitbeziehungen)
$Succ^o(i)$	Menge der Stoffstromvorgänge, die direkte Nachfolger des Rückbauvorgangs $i$ sind
$Succ^o(j)$	Menge der Stoffstromvorgänge, die direkte Nachfolger des Stoffstromvorgangs $j$ sind
$Succ^-(l)$	Menge der Vorgänge, die relevante Nachfolger des Vorgangs $l$ sind
$\overline{Succ^-}$	alle Mengen der relevanten Nachfolger
$\overline{Succ}(i)$	Menge der (Rückbau-)Vorgänge, die transitive Nachfolger des (Rückbau-)Vorgangs $i$ sind
$\overline{Succ^o}(i)$	Menge der Stoffstromvorgänge, die transitive Nachfolger des Rückbauvorgangs $i$ sind
$\overline{Succ^o}(j)$	Menge der Stoffstromvorgänge, die transitive Nachfolger des Stoffstromvorgangs $j$ sind
$s$	Senke
$st$	Stelle in einem Petri-Netz
$T$	obere Schranke für die Projektdauer
$\mathcal{T}_1, \mathcal{T}_2$	Mengen von Testinstanzen
$t, t', t_1, \dots$	Zeitpunkte
$t^c$	Zeitpunkt, zu dem ein Ressourcenkonflikt auftritt

$tr$	Transition in einem Petri-Netz
$u$	Hilfsvariable zur Zählung von Stoffeinheiten
$ur_{lk}^{y+}$	Hilfsvariablen für diejenige positive Nutzung der kumulativen Ressource $k$ durch Vorgang $l$ , zu deren Ausgleich kein relevanter Nachfolger gefunden werden konnte
$V$	Menge der (Rückbau-)Vorgänge
$V^o$	Menge der Stoffstromvorgänge
$\hat{V}^o$	aggregierbare Teilmenge der Stoffstromvorgänge
$\mathcal{W}$	Menge der Stoffstrompfade
$W1, W2, \dots$	exemplarische Stoffstrompfade
$w$	Stoffstrompfad
$w_i$	Gewicht des Vorgangs $i$ (im Zusammenhang mit Zielsetzungen)
$x$	Entscheidungsvariable oder allgemeiner Platzhalter
$x_{it}$	Puls-Startvariable, die angibt, ob Rückbauvorgang $i$ zum Zeitpunkt $t$ startet oder nicht
$x_{jt}$	Puls-Startvariable, die angibt, ob Stoffstromvorgang $j$ zum Zeitpunkt $t$ startet oder nicht
$Y_k$	Niveau, ab welchem die Nutzung der Ressource $k$ mit Kosten verbunden ist
$\mathbb{Z}_{\geq 0}$	Menge der positiven ganzen Zahlen inklusive null
$z$	Zielfunktionswert
$z$	Zielfunktionswert in Abhängigkeit von Entscheidungsvariable $x$

### Griechische Buchstaben

$\Gamma_k^Y(t)$	zum Zeitpunkt $t$ verfügbare Lagerkapazität
$\delta_{ii'}$	Zeitabstand zwischen Vorgang $i$ und Vorgang $i'$
$\delta$	Matrix aller Zeitabstände
$\delta_{jj'}^{flow}$	Zeitabstand zur stoffstrominduzierten Zeitbeziehung zwischen den Stoffstromvorgängen $j$ und $j'$
$\delta_{ij}^{rel}$	Zeitabstand zur Freisetzungsbeziehung zwischen Rückbauvorgang $i$ und Stoffstromvorgang $j$
$\lambda_{lk}(S^o)$	Hilfsfunktion zur Formulierung der relevanten kumulativen Ressourcenrestriktionen (5.21)
$\mu$	Stoffstromvorgangsfolge
$\nu$	Diskontfaktor
$\pi$	Prioritätsregel
$\pi(i)$	Priorität des (Rückbau-)Vorgangs $i$
$\tau$	Zeitpunkt
$\Phi_k^\alpha(t)$	Anzahl der zum Zeitpunkt $t$ im Stoffstrombereich genutzten Einheiten der erneuerbaren Ressource $k$
$\chi_l(S^o)$	Hilfsfunktion zur Formulierung der relevanten kumulativen Ressourcenrestriktionen (5.14)



# 1 Einleitung

## 1.1 Ausgangslage und Problemstellung

Die Europäische Union hat in ihrer im Jahr 2020 beschlossenen Verordnung über die Einrichtung eines Rahmens zur Erleichterung nachhaltiger Investitionen (der sogenannten „EU-Taxonomie“) definiert, dass Wirtschaftstätigkeiten unter anderem als ökologisch nachhaltig gelten, falls sie einen „wesentlichen Beitrag zum Übergang zur Kreislaufwirtschaft leisten“ (Europäische Union 2020, Art. 13, Abs. 1). In Absatz 1g wird weiter spezifiziert, dass ein solcher wesentlicher Beitrag unter anderem vorliegt, wenn durch die Wirtschaftstätigkeit „die Abfallerzeugung, einschließlich der Erzeugung von Abfall [...] beim Bau und Abriss von Gebäuden, vermieden oder verringert“ wird. Der Übergang zur Kreislaufwirtschaft stellt eines der sechs in der Verordnung genannten Umweltziele dar. Darüber hinaus ist „der Übergang zur einer sicheren, klimaneutralen, klimaresilienten, ressourceneffizienten und stärker kreislauforientierten Wirtschaft [...] von zentraler Bedeutung für die Sicherung der langfristigen Wettbewerbsfähigkeit der Wirtschaft in der Union“ (Europäische Union 2020, Einleitung, Abs. 4).

Mit Blick auf die in der Verordnung angesprochene Abfallerzeugung beim Bau und Abriss von Gebäuden lässt sich gemäß Daten des Statistischen Bundesamtes feststellen, dass Bau- und Abbruchabfälle in Deutschland den Großteil des Abfallaufkommens ausmachen – im Jahr 2020 mit 229,4 Mio. Tonnen ca. 55 % (Statistisches Bundesamt 2020). Unter den Bau- und Abbruchabfällen nimmt der Bodenaushub laut Kreislaufwirtschaft Bau (2021, S. 6) mit einem ca. 60-prozentigen Anteil die dominierende Position ein. Die verbleibenden Bau- und Abbruchabfälle setzen sich aus Straßenaufbruch (ca. 6 %), Bauschutt (ca. 27 %) und Baustellenabfällen (ca. 6 %) zusammen. Unter dem Begriff des Bauschutts werden diejenigen Baustellenabfälle zusammengefasst, die aus dem Abbruch von Gebäuden, Tiefbauten und Ingenieurbauwerken stammen und typischerweise sehr unterschiedlich zusammengesetzt sein können (Müller 2018, S. 23).

Das hohe Abfallaufkommen im Bausektor macht deutlich, dass Bemühungen zur Verringerung, zur Aufbereitung und zum Recycling hier besonders bedeutsam sind, um das von der EU ausgegebene Umweltziel des Übergangs zu einer Kreislaufwirtschaft zu erreichen. Dies wird auch durch das Umweltbundesamt unterstrichen: „Der Bestand an Gebäuden und Infrastrukturen ist mit rund 28 Milliarden Tonnen (Stand 2010 [...]) inzwischen ein bedeutendes, menschengemachtes Rohstofflager, das nach Nutzungsende wieder dem Recycling zugeführt werden kann“ (Umweltbundesamt 2021). Um möglichst viele Baustoffe recyceln zu können, wird statt eines Abbruchs, welcher die Entfernung eines Bauwerks ohne Rück-

sicht auf die verbauten Materialien bezeichnet, ein selektiver Rückbau angestrebt (Müller 2018, S. 19). Ziel hierbei ist, die Vermischung von Materialien größtmöglich zu vermeiden. Obwohl beim selektiven Rückbau bereits eine hohe Sortiertiefe erreicht werden kann, ist üblicherweise eine Aufbereitung der Bauabfälle erforderlich, um daraus neuen Recycling-Baustoff mit bestimmten Eigenschaften zu erzeugen (Müller 2018, S. 19 und 61). Je nach Aufbereitungstechnologie und verfügbarem Ausgangsmaterial lassen sich unterschiedliche Qualitätsstufen für die erzeugten Recycling-Baustoffe erreichen. In den meisten Fällen handelt es sich bei einer Aufbereitung um die Anwendung verschiedener Verfahren zum Zerkleinern, Klassieren oder Sortieren von Bauabfällen (Müller 2018, S. 125). Darüber hinaus sind auch weitere Aufbereitungsschritte möglich, beispielsweise die Entfernung umweltrelevanter Kontaminationen in Altschotter mittels biologischen, chemischen oder thermischen Verfahren (Müller 2018, S. 273). Die Aufbereitung kann sowohl direkt auf der Baustelle mit mobilen Anlagen oder zentralisiert mit stationären Anlagen erfolgen (Müller 2018, S. 125). Zwischen einzelnen Aufbereitungsschritten ist es erforderlich, Material zu lagern und zu transportieren. In aller Regel unterliegen die Aufbereitung, der Transport und die Lagerung entsprechenden Kapazitätsbeschränkungen, da beispielsweise die Anzahl der verfügbaren Anlagen oder der verfügbare Platz begrenzt sind. Insbesondere bei städtischen Rückbauprojekten sind beengte Platzverhältnisse zu beobachten, wie Rosen (2021, S. 134f) exemplarisch anhand eines im Jahr 2016 durchgeführten Rückbauprojekts in der Frankfurter Innenstadt zeigt. Beim dortigen Rückbau eines Geschäftshauses mussten demontierte Materialien in das zweite Untergeschoss verbracht, manuell in Container verladen und über einen Lieferantenaufzug zum Abtransport hochgefahren werden. Die Abtransporte konnten nur in den frühen Morgen- und späten Abendstunden durchgeführt werden. Obwohl es sich bei diesem Beispiel um ein einzigartiges Projekt handelt, lässt sich auch generell feststellen, dass Lagerflächen in Rückbauprojekten üblicherweise nur in beschränktem Ausmaß zur Verfügung stehen. Aus Sicht des Projektmanagements ist es daher erforderlich, „den Materialanfall so zu steuern, dass die Anzahl gleichzeitig benötigter Container oder Zwischenlagerflächen mit dem Platzangebot in Einklang gebracht werden kann“ (Schultmann 1998, S. 3).

Unter den Rückbauprojekten nimmt der Rückbau kerntechnischer Anlagen aufgrund der Größe der rückzubauenden Objekte und der hohen strahlenschutzrechtlichen Sicherheitsanforderungen eine besondere Stellung ein (Hübner 2019, S. 11). Beispielsweise beträgt nach Angaben der EnBW Kernkraft GmbH (2018, S. 25) die Gesamtmasse des sich in der Stilllegung befindlichen Kernkraftwerks Philippsburg, Block 2, ca. 782 500 Tonnen. Davon sind ca. 15 600 Tonnen den radioaktiven Reststoffen zuzuordnen, die gemäß den §§ 31–42 der Strahlenschutzverordnung (StrlSchV) nur nach erfolgreicher Freimessung, d. h. nach Sicherstellung der radioaktiven Kontaminierungs- und Aktivierungsfreiheit, in den Stoffkreislauf zurückgeführt werden dürfen. Ansonsten müssen sie der Endlagerung zugeführt werden. Im Sinne einer kreislauforientierten Wirtschaft und vor dem Hintergrund knapper Zwischen- und Endlagerkapazitäten ist es „ein ökologisch und ökonomisch vordringliches Gebot“, möglichst viele der radioaktiven Reststoffe freimessen zu können (Thierfeldt und Schartmann 2009, S. 44). Um dieses Ziel zu erreichen, werden bei der Verarbeitung der anfallenden Reststoffe zahlreiche Dekontaminationsverfahren, wie beispielsweise Waschen, Bürsten, Abfräsen, Sandstrahlen, etc. eingesetzt. Neben der Dekontamination

und der Freimessung müssen die Reststoffe auch zerlegt, Vormessungen unterzogen, in Endlagerbehältern konditioniert, innerhalb der Anlage transportiert und aus der Anlage ausgeschleust werden. Zur Durchführung all dieser Schritte werden Arbeitskräfte und Werkzeuge bzw. Anlagen benötigt, die beispielsweise aufgrund mangelnder Verfügbarkeit auf dem Markt, aus Kostengründen oder wegen Platzmangels innerhalb der Anlagen nicht unbegrenzt zur Verfügung stehen. Auch Lagerflächen zur Pufferung der Reststoffe auf ihrem Weg vom Abbauort bis hin zur Freimessung und der Ausschleusung aus der Anlage sind begrenzt, wie beispielsweise Brusa u. a. (2002, S. 7) anhand eines Grundrisses des italienischen Kernkraftwerks Trino zeigen.

Die sowohl in konventionellen Rückbauprojekten als auch beim Rückbau kerntechnischer Anlagen vorhandenen Kapazitätsbeschränkungen im Umgang mit Reststoffen unterstreichen die Notwendigkeit, die Verarbeitung von Reststoffen bereits in der Ablaufplanung eines Rückbauprojekts zu berücksichtigen. Denn sobald im Verlauf eines Projekts die Verarbeitungs- und Lagerkapazitäten nicht mehr ausreichen, um weitere Stoffe aufzunehmen, muss mit der Demontage pausiert werden. Dies erfordert kurzfristige Umplanungen, die angesichts bereits vergangener, unumkehrbar getroffener Entscheidungen zu suboptimalen Ergebnissen führen können. Beispielsweise kann dann eine Verschiebung eines kritischen Vorgangs erforderlich werden, weil andere, unkritische Vorgänge bereits begonnen wurden. Dadurch verlängert und verteuert sich das Rückbauprojekt. Ist hingegen im Voraus schon bekannt, dass innerhalb eines bestimmten Zeitraums eine Überlastung der Reststoffverarbeitung zu erwarten ist, kann die Demontage so gesteuert werden, dass sich die Freisetzung besonders aufbereitungsintensiver Reststoffe auf andere Zeiten verlagert und stattdessen kritische Vorgänge bevorzugt ausgeführt werden. Dass die Reststoffverarbeitung im Management von kerntechnischen Rückbauprojekten berücksichtigt werden soll, bestätigt die Nuclear Energy Agency (2022): „There is a need for integrated project management to support decommissioning projects (with reasonable safety focus) taking into account experience from operation maintenance, radiation protection, engineering and waste management.“

Eine besondere Herausforderung bei der Integration von Stoffströmen in die Ablaufplanung eines Rückbauprojekts stellt die vermeintliche Unvereinbarkeit der Planungsebenen dar. Während der Rückbau eines Gebäudes, einer Anlage oder einer Infrastruktur üblicherweise die klassischen Merkmale eines Projekts erfüllt, weist die vom Repetitionstyp einer Serienfertigung ähnelnde Stoffverarbeitung eher produktionswirtschaftliche Merkmale auf. In der Literatur werden die Projektplanung und die Produktionsplanung meist getrennt voneinander betrachtet, sodass die Übertragbarkeit der entwickelten Modelle auf die stoffstrombasierte Rückbauplanung begrenzt ist. Des Weiteren stellt die planungstechnische Einbindung der begrenzten Lagerkapazitäten eine Herausforderung dar, da Lager aus mathematischer Sicht infolge der zeitlich auseinanderfallenden Ein- und Auslagerungsvorgänge die Anwendung quantitativer Verfahren erschweren. Somit existieren bisher keine Methoden, mit denen die durch die Reststoffverarbeitung induzierten Restriktionen in vollumfänglicher Weise auf die Rückbauplanung abgebildet werden können. Vorangegangene Arbeiten, die sich mit der Ablaufplanung von Rückbauprojekten befassen, berücksichtigen stoffstrombezogene Restriktionen nicht bzw. nur mit einschränkenden Annahmen (z.

B. Schultmann 1998; Bartels 2009; Volk 2017; Hübner 2019). Wie Gehring, Winkler u. a. (2021) belegen, bieten auch die in der Projektplanung häufig eingesetzten Softwareprodukte nicht den erforderlichen Funktionsumfang, um alle relevanten stoffstrombezogenen Restriktionen (insbesondere beschränkte Lagerkapazitäten) abzubilden.

### 1.2 Zielsetzung und Lösungsweg

Ziel dieser Arbeit ist die Entwicklung und die Anwendung eines stoffstrombasierten Optimierungswerkzeugs zur Ablaufplanung von Rückbauprojekten.

Unter dem Begriff der *Ablaufplanung* werden die beiden Planungsbereiche der Termin- und Kapazitätsplanung eingeordnet. Hierzu hat sich in den vergangenen Jahrzehnten ein umfangreicher Forschungszeitweig herausgebildet, der quantitative Ansätze zur Behandlung entsprechender Fragestellungen bereitstellt. Diese Arbeit soll auf derartigen Ansätzen aufbauen und sie weiterentwickeln. Die in der Praxis üblichen und auch in gängigen Ansätzen betrachteten Modellbestandteile, wie Vorrangbeziehungen oder begrenzt verfügbare Ressourcen, sollen einbezogen werden.

Unter einem *Optimierungswerkzeug* wird im Rahmen dieser Arbeit eine Kombination aus einem mathematischen Modell, auf Basis dessen ein Optimierungsproblem formuliert wird, und einem oder mehreren algorithmisch beschriebenen Verfahren zur Lösung dieses Optimierungsproblems verstanden. Die Umsetzung und Anwendung des Optimierungswerkzeugs erfolgt rechnergestützt. Um die Absicht der Optimierung, d. h. die Minimierung oder Maximierung eines Optimierungsziels, sprachlich zu unterstreichen, wird bewusst nicht von einem Planungswerkzeug gesprochen. Als Optimierungsziel wird die Minimierung der Projektdauer gewählt, da davon ausgegangen wird, dass sich die Projektdauer weitgehend proportional zu den Projektkosten verhält und sie somit ein adäquater Stellvertreter für die aus ökonomischer Sicht zentrale Zielsetzung der Kostenminimierung ist. Ein Großteil der in der Literatur behandelten Modelle zur optimierenden Projektablaufplanung fokussiert sich auf die Projektdauer als Zielsetzung (Węglarz u. a. 2011, S. 128). Der Verzicht auf eine kostenbasierte Zielsetzung ist auch praktischen Gründen geschuldet. So sind laut Rosen (2021, 54f) Kostenschätzungen zu Rückbauprojekten mangels Datenbasis mit großen Unsicherheiten behaftet, wohingegen der Zeitfaktor besser messbar ist. Dass die Minimierung der Projektdauer mit dem Ziel der Kostenminimierung in Einklang steht, bestätigen Thierfeldt und Scharmann (2009, S. 34) für den Rückbau kerntechnischer Anlagen: „Da bis zur Hälfte der Gesamtkosten eines Projekts [zum Rückbau kerntechnischer Anlagen] auf Personalkosten und sonstige Kosten, die direkt mit der Abwicklungsdauer verbunden sind, entfallen können, werden alle Faktoren vermieden, die ein Projekt verzögern könnten.“

Die Verwendung des handwerklich konnotierten Begriffs „Werkzeug“ verdeutlicht, dass die praktische Anwendbarkeit der entwickelten Modelle und Methoden beachtet werden soll. Dies geht mit der Absicht dieser Arbeit einher, die Inhalte möglichst anwendungsnah zu vermitteln, ohne dass fachspezifische Vorkenntnisse erforderlich sind. Komplexe Formalismen sollen mit Beispielen veranschaulicht werden. Der Bezug zur Anwendung soll

regelmäßig hergestellt werden. Hierzu eignet sich aufgrund der Komplexität seiner Rahmenbedingungen insbesondere der Rückbau kerntechnischer Anlagen. Dennoch soll das Optimierungswerkzeug auf eine Weise entwickelt werden, die es nicht ausschließlich auf den kerntechnischen Anwendungsfall einschränkt, sondern für jede Art von Rückbauprojekt einsetzbar macht. Da zur Lösung eines Optimierungsproblems nicht notwendigerweise ein einzelner Ansatz ausreicht bzw. besser abschneidet als alle anderen Ansätze, kann das Optimierungswerkzeug auch eine Sammlung mehrerer Lösungsansätze umfassen. Dann ist jedoch im Sinne der praktischen Anwendbarkeit sicherzustellen, dass Empfehlungen abgegeben werden, unter welchen Umständen welcher konkrete Ansatz zum Einsatz kommen soll. Es sei angemerkt, dass trotz der angestrebten anwendungsnahen Darstellung ein Verständnis von mathematischen Grundlagen und Konventionen, beispielsweise zur Mengenlehre oder zur Formulierung von Pseudocodes, vorausgesetzt wird.

Die Bezeichnung des Optimierungswerkzeugs mit dem Attribut *stoffstrombasiert* stellt eine Verbindung zu der im vorangegangenen Abschnitt skizzierten Problemstellung der stoffstrombezogenen Restriktionen in einem Rückbauprojekt her. Das zu entwickelnde Optimierungswerkzeug soll nicht ausschließlich zur Termin- und Kapazitätsplanung der Demontage dienen, sondern auch die Kapazitätsengpässe bei der Verarbeitung und Lagerung anfallender Stoffe berücksichtigen. Dadurch soll Projektverantwortlichen ein Werkzeug an die Hand gegeben werden, welches es ihnen ermöglicht, die rückkoppelnden Auswirkungen der Stoffverarbeitung auf den Ablauf des Rückbauprojekts einzuschätzen. Dies soll dazu beitragen, die Demontage besser mit vorhandenen Aufbereitungs-, Lager- oder Transportkapazitäten synchronisieren und damit einen realistischeren Ablaufplan ausarbeiten zu können. Ein weiterer Mehrwert soll für Projektverantwortliche darin bestehen, mittels Variation von Parametern methodisch fundiert abschätzen zu können, wie sich Veränderungen in den Rahmenbedingungen auf den Projektablauf auswirken. Beispielsweise sollen sich folgende Fragen beantworten lassen:

- Welche Beschleunigung des Projektablaufs lässt sich erzielen, wenn in eine zusätzliche mobile Aufbereitungsanlage investiert wird?
- Können die vorhandenen Lagerflächen weiter reduziert werden, ohne die Projektausführung zu verzögern?
- Welche Auswirkungen hat ein temporärer Ausfall einer Dekontaminationsanlage auf das Erreichen der kommenden Meilensteine?

Diese exemplarischen Fragen veranschaulichen, dass das stoffstrombasierte Optimierungswerkzeug sowohl bei strategischen als auch taktischen bzw. operativen Entscheidungen im Zusammenhang mit Investitionen oder dem Baustellenlayout unterstützen soll.

Angaben zu den mengen- oder volumenmäßig anfallenden Stoffen, den Verarbeitungspfaden der Stoffe inklusive dazwischengeschalteter Lagerflächen, den Verarbeitungs- und Lagerkapazitäten sowie den Demontage- und Verarbeitungsdauern werden als gegeben betrachtet und bilden die Mindestmenge an erforderlichen Eingabedaten für das Optimierungswerkzeug. Darüber hinaus sollen zeitliche Abhängigkeiten und weitere Ressourcen, wie benötigte Maschinen, Werkzeuge oder Fachkräfte miteinbezogen werden können.

Technische Konzepte zur Ermittlung der Stoffmengen, der Demontage- und Verarbeitungsdauern oder anderer Parameter werden nicht als Bestandteil des Optimierungswerkzeugs betrachtet und sind daher nicht Gegenstand dieser Arbeit.

Unter dem Begriff der *Rückbauprojekte* werden Projekte zusammengefasst, die sich auf jedwede Art von kleineren bis hin zu großen, komplexen und stoffstromintensiven Gebäuden, Anlagen oder Infrastrukturen beziehen. Das zu entwickelnde Optimierungswerkzeug soll hinsichtlich der Größenordnung der Anwendung möglichst keine Einschränkungen vorgeben. Bei der Formulierung eines Modells soll auf Allgemeingültigkeit geachtet werden, um unter Berücksichtigung der Handhabbarkeit und Lösbarkeit eine größtmögliche Bandbreite an Restriktionen aus der Rückbaupraxis abbilden zu können. Im Hinblick auf die rechnergestützte Umsetzung des Optimierungswerkzeugs soll darauf geachtet werden, dass die entwickelten Berechnungsmethoden auch für kleinteilige und stoffstromintensive Anwendungsfälle in angemessener Rechenzeit qualitativ hochwertige Ergebnisse liefern.

Zusammengefasst soll das zu entwickelnde Optimierungswerkzeug die Antwort auf folgende Forschungsfrage liefern:

*Wie kann ein Ablaufplan für ein Rückbauprojekt unter Berücksichtigung von Ressourcenrestriktionen und Stoffströmen ausgearbeitet werden, der die Projektdauer minimiert?*

Zum Erreichen des formulierten Ziels wird folgendermaßen vorgegangen:

Um die methodischen Inhalte dieser Arbeit auch ohne einschlägige Vorkenntnisse verstehen zu können, startet Kapitel 2 mit der Definition grundlegender Begriffe. Anschließend werden verschiedene in der Literatur betrachtete Optimierungsprobleme vorgestellt, welche einen Bezug zur Projektablaufplanung aufweisen. Dabei wird auch auf existierende Lösungsansätze eingegangen. Das Kapitel schließt mit thematisch relevanten Grundlagen zum stoffstrombasierten Produktionsmanagement.

Kapitel 3 fokussiert sich auf die konzeptionelle Formulierung des mathematischen Problems, welches den Kern des stoffstrombasierten Optimierungswerkzeugs bildet. Dazu wird die betrachtete Problemstellung zunächst verbal beschrieben und anhand exemplarischer Ablaufpläne illustriert. Der Rückbau kerntechnischer Anlagen dient, wie bereits erläutert, als Anwendungsfall. Daher enthält Kapitel 3 einen Abschnitt, der sich ausführlich mit den technischen und gesetzlichen Rahmenbedingungen des Rückbaus kerntechnischer Anlagen befasst und basierend darauf ihre Eignung zum Einsatz eines Optimierungswerkzeugs ableitet. Anschließend werden aus anwendungsübergreifender Perspektive Modellanforderungen, -einschränkungen und -abgrenzungen erörtert, bevor das Problem ausformuliert wird. Ein Vergleich mit existierenden Ansätzen in der Literatur zeigt die vorhandene Forschungslücke bei der stoffstrombasierten Ablaufplanung von Rückbauprojekten auf.

Da die konzeptionelle Formulierung des mathematischen Problems die stoffstrombezogenen Restriktionen nur auf abstrakte Weise berücksichtigt, befasst sich Kapitel 4 mit einer Konkretisierung des Modells. Dazu wird der Stoffverarbeitung eine zeitliche Dimension

zugewiesen, woraufhin ein integriertes Modell zur Planung von Rückbauprojekt und Stoffströmen formuliert wird. Da es für diese Zuweisung der zeitlichen Dimension keine idealtypische Herangehensweise gibt, werden verschiedene Ausgestaltungsmöglichkeiten betrachtet und algorithmisch als sogenannte Transformationsschemas ausformuliert. Ein Transformationsschema ist somit als Hilfsmittel bei der Ermittlung der Eingabedaten für das Modell zur integrierten Planung von Rückbauprojekt und Stoffströmen zu verstehen. Es stellt das Bindeglied zwischen der realen Problemstellung und dem quantitativ lösbaren mathematischen Problem dar. Auf Charakteristika des mathematischen Problems, auf Modellierungsaspekte und auf Modellerweiterungen wird in Kapitel 4 ebenfalls ausführlich eingegangen. Das Kapitel schließt mit einem Überblick und einer Zusammenfassung bisher erarbeiteter Inhalte.

Kapitel 5 befasst sich mit Lösungsverfahren für das in Kapitel 4 formulierte Problem. Zunächst werden existierende Ansätze aus der Literatur untersucht, um festzustellen, dass es der Entwicklung neuer, problemspezifischer Lösungsverfahren bedarf. Anschließend werden vier Lösungsverfahren für unterschiedliche Arten der Problemzusammensetzung ausgearbeitet und miteinander verglichen. Ergänzt werden diese neu entwickelten Lösungsverfahren um die Formulierung eines Constraint- und eines linearen Programms. Diese Programme können unter Verwendung kommerziell verfügbarer Solver ausgeführt werden. Das Kapitel schließt mit einem Überblick und einer Zusammenfassung bisher erarbeiteter Inhalte.

Umfangreiche Rechenexperimente mit wechselndem Fokus auf die Stoffströme, das Rückbauprojekt und die Zielsetzung demonstrieren in Kapitel 6 die Anwendung der vorgeschlagenen Lösungsverfahren anhand von Testinstanzen. In der Auswertung der Rechenergebnisse werden Empfehlungen dazu abgegeben, welche Lösungsverfahren unter welchen Bedingungen eingesetzt werden sollten.

Eine abschließende Zusammenfassung und eine kritische Würdigung mit Ausblick folgen in Kapitel 7.

Es sei darauf hingewiesen, dass bei der Ausarbeitung des Optimierungswerkzeugs auf Erläuterungen zur softwaretechnischen Implementierung der entwickelten Modelle und Lösungsverfahren verzichtet wird, obwohl eine solche Implementierung zur experimentellen Anwendung selbstverständlich erforderlich ist und umgesetzt wurde. Damit folgt diese Arbeit der gängigen Praxis, die Implementierung separat von methodischen Inhalten zu dokumentieren, wie vergleichbare Arbeiten zeigen (z. B. Bartels 2009; Volk 2017). Eine solche Dokumentation wurde in Form eines Handbuchs angefertigt.

Teile der nachfolgenden Inhalte (insbesondere Teile der Kapitel 3, 4 und 5) wurden bereits in Gehring, Volk u. a. (2022) veröffentlicht. Sie werden in dieser Arbeit verfeinert, mit Beispielen hinterlegt und umfangreich erweitert. Da sie vom Autor dieser Arbeit entwickelt wurden, sind an den entsprechenden Stellen keine Referenzen eingefügt. In den beiden Abschnitten mit einem Überblick über erarbeitete Inhalte wird jeweils darauf eingegangen, wo Überschneidungen vorhanden sind.





## 2 Methodische Grundlagen

In Abschnitt 2.1 werden grundlegende Begriffe definiert und zueinander in Beziehung gebracht. Um einfache und klare Formulierungen zu ermöglichen, werden begriffliche Konventionen für den weiteren Verlauf der Arbeit etabliert. Abschnitt 2.2 befasst sich mit verschiedenen in der Literatur diskutierten Optimierungsproblemen, die relevant für die Entwicklung des Optimierungswerkzeugs sind. Ihrer thematischen Zugehörigkeit entsprechend werden diese Probleme als Schedulingprobleme bezeichnet. Zunächst wird das allgemeine Schedulingproblem vorgestellt, welches daraufhin schrittweise spezifiziert und erweitert wird. Der Fokus liegt auf der Problemformulierung und der Vermittlung dazu erforderlicher Konzepte und Notationen. Auf Ansätze zur Lösung der Schedulingprobleme wird in Abschnitt 2.3 eingegangen. Für Details wird auf weiterführende Kapitel dieser Arbeit oder entsprechende Literatur verwiesen. Abschnitt 2.4 befasst sich mit Grundlagen zum stoffstrombasierten Produktionsmanagement. Nach einer begrifflichen Abgrenzung wird dargelegt, inwiefern die Projektplanung und die Produktionsplanung in Zusammenhang zueinander stehen. Basierend darauf werden graphentheoretische Modelle aus dem Produktionsmanagement vorgestellt und auf ihre Eignung zur Abbildung von Stoffströmen untersucht. Dies bildet eine Grundlage für die im weiteren Verlauf der Arbeit vorgenommene Einbindung von Stoffströmen in die Ablaufplanung von Rückbauprojekten. Abschließend erfolgt eine Klassifikation von Stoffströmen anhand ihrer Struktur oder genauer, der sogenannten Vergenztypen.

### 2.1 Begriffe

#### 2.1.1 Projektplanung und Scheduling

Ein *Projekt* besteht nach DIN ISO 21500 „aus einer einzigartigen Gruppe von Prozessen, die auf eine Zielsetzung ausgerichtete, koordinierte und gesteuerte Vorgänge mit Beginn- und Fertigstellungsterminen umfassen“ (DIN Deutsches Institut für Normung 2016, S. 10). Die *Projektplanung* kann als eine Phase im Projektlebenszyklus betrachtet werden, in der die detaillierte Planung des Projektablaufs erfolgt (Zimmermann u. a. 2006, S. 4). Weitere Phasen sind demnach die Projektkonzeption, die Projektspezifikation und die Projektrealisierung. Die DIN ISO 21500 definiert die Projektplanung hingegen als Gruppe von Projektmanagementprozessen, die in verschiedenen Phasen des Projektlebenszyklus zur Anwendung kommen und zur Entwicklung von Planungsdetails dienen (DIN Deutsches Institut für Normung 2016, S. 16-19).

Gemäß obiger Definition ist die Durchführung eines Projekts mit Zielen verbunden, wie beispielsweise der Minimierung der Projektdauer oder der Minimierung der Kosten. Werden diese Ziele bei der Projektplanung systematisch berücksichtigt, liegt für die zu treffenden Entscheidungen eine Absicht zur *Optimierung* zugrunde. Das interdisziplinäre Feld des *Operations Research (OR)* befasst sich mit der Unterstützung von Entscheidungen mit Hilfe von Erkenntnissen und Werkzeugen aus der Mathematik, der Wirtschaftswissenschaften und der Informatik, wobei häufig ebenfalls eine Absicht zur *Optimierung* zugrunde liegt (Briskorn 2020, S. 2, vgl. Abschnitt 2.1.3). In einem Teilgebiet des Operations Research werden verschiedene Fragestellungen der *Ablaufplanung* untersucht, wobei das Teilgebiet oft dem englischsprachigen Begriff entsprechend als *Scheduling* bezeichnet wird. Anwendungsfälle finden sich im industriellen Kontext, wo beispielsweise die Zuordnung von Produktionsaufträgen zu verschiedenen Maschinen zu planen ist (Jaehn und Pesch 2019, S. 9). Die Ablaufplanung kann in weitere Planungsbereiche unterteilt werden, wie in die *Terminplanung* (oder *Zeitplanung*) und die *Kapazitätsplanung* (oder *Ressourcenplanung*), wobei eine klare Abgrenzung nicht möglich ist und gängige Methoden aus dem Operations Research in der Regel mehrere Planungsbereiche integriert behandeln (Schulzmann 1998, S. 45 und S. 50). Die *Projektabschlussplanung* oder das *Projektscheduling* fasst diejenigen Modelle und Methoden zusammen, die für die Termin- und/oder Kapazitätsplanung eines Projekts entwickelt wurden. Je nach Ausgestaltung können beispielsweise auch Aspekte aus der Kostenplanung oder, bei Berücksichtigung von Unsicherheiten, des Risikomanagements einfließen (vgl. z. B. Hübner 2019, S. 138). Im Gegensatz zu industriellen Anwendungen, die üblicherweise wiederkehrender Art sind, ist ein Projekt einzigartig und auf einen vorgegebenen Zeitraum begrenzt (Zimmermann u. a. 2006, S. 2). Die Grenzen der verschiedenen Anwendungsgebiete der Projektabschlussplanung sind allerdings fließend. Beispielsweise behandeln Schwindt und Trautmann (2000) eine Anwendung aus der industriellen Produktionsplanung mit Methoden der Projektabschlussplanung. In Abschnitt 2.4.2 wird erläutert, unter welchen Voraussetzungen diese Übertragung aus der Produktionsplanung in die Projektabschlussplanung möglich ist. Wenn im weiteren Verlauf dieser Arbeit von Scheduling gesprochen wird, bezieht sich dies vor dem Hintergrund des behandelten Themas stets auf das Teilgebiet des Projektschulings.

Das Scheduling wird in der OR-Literatur teilweise mit der Projektplanung gleichgesetzt (z. B. Zimmermann u. a. 2006). Aus Sicht des Projektmanagements umfasst die Projektplanung allerdings Themengruppen, die über den gewöhnlichen Untersuchungsrahmen des Scheduling hinausgehen, wie beispielsweise Qualitätsplanung oder Kommunikationsplanung (DIN Deutsches Institut für Normung 2016, S. 17f). Das Scheduling ist aus dieser Perspektive lediglich ein Forschungsfeld mit dem Ziel, bestimmte Themengruppen der Projektplanung methodisch zu unterstützen. Für die vorliegende Arbeit wird folgende begriffliche Konvention etabliert: Der Begriff des Scheduling wird im mathematisch-methodischen Kontext verwendet und deutet stets auf einen Zusammenhang zum OR hin. Der Begriff der Projektplanung wird im anwendungsorientierten Kontext verwendet und deutet auf einen Zusammenhang zum ganzheitlichen Projektmanagement hin.

### 2.1.2 Stoffströme und Materialflüsse

Ein *Stoffstrom* kann allgemein als „Weg eines Stoffes von seiner Gewinnung als Rohstoff über die verschiedenen Stufen der Veredelung bis zur Stufe der Endprodukte, den Gebrauch/Verbrauch des Produktes, ggf. seine Wiederverwendung/Verwertung bis zu seiner Entsorgung“ verstanden werden (Bundestag 1994b, S. 301; zitiert nach Bundestag 1994a, S. 337). Zur Einbindung von Stoffströmen in die Projektplanung, wie sie im Rahmen dieser Arbeit erfolgt, ist die genannte Definition jedoch zu weit gefasst. Je nach Systemgrenze kann unter einem Stoffstrom auch nur ein Teil des Weges verstanden werden, den ein Stoff während seines Lebenszyklus zurücklegt. Für Rückbauprojekte umfasst dies in der Regel nur der Weg eines Stoffes ab dem Zeitpunkt seiner Demontage bis zu seiner Wiederverwendung, Verwertung oder Entsorgung. Statt von der Demontage wird in dieser Arbeit von der *Freisetzung* eines Stoffstroms gesprochen, womit das Ereignis beschrieben wird, ab dem ein Stoff seine ursprüngliche Funktion nicht mehr erfüllt. Nach der Freisetzung kann ein Stoff direkt wiederverwendet, verwertet oder entsorgt werden. Alternativ kann ein Stoff zunächst verschiedene Aufbereitungsprozesse durchlaufen, um beispielsweise Werkstoffverbunde zu trennen, qualitativ hochwertige Sekundärrohstoffe zu gewinnen oder eine Entsorgung vorzubereiten (Schultmann 2003, S. 70). Diese Aufbereitungsprozesse werden als *Verarbeitungsschritte* bezeichnet. Das Ende eines hinsichtlich der Systemgrenzen relevanten Stoffstroms wird als (*Stoffstrom-*)*Senke* bezeichnet. Die Senke kann entweder mit dem Ende des gesamten Stoffstroms zusammenfallen und beschreibt dann das Ereignis, ab welchem ein Stoff als wiederverwendet, verwertet oder entsorgt gilt. Falls das Ende des gesamten Stoffstroms jedoch nicht mehr relevant ist, weil beispielsweise Stoffströme aus einer innerbetrieblichen Perspektive betrachtet werden und ein externer Dienstleister für die Entsorgung verantwortlich ist, entspricht die Senke demjenigen Ereignis, zu dem der Stoffstrom an den Dienstleister übergeben wird.

Die genannten Definitionen zu Stoffströmen entstammen einer produktionswirtschaftlichen Perspektive. Davon zu unterscheiden ist die Rolle von Stoffen aus logistischer Perspektive. Zur begrifflichen Abgrenzung bietet es sich an, im logistischen Zusammenhang von Materialflüssen zu sprechen. Diese Abgrenzung wird im Rahmen der vorliegenden Arbeit konsistent eingehalten, hat sich in der Literatur allerdings nicht derart etabliert. Insbesondere in der englischsprachigen Literatur werden sowohl Stoffe als auch Materialien mit „material“ übersetzt. Ein *Materialfluss* beschreibt „diskrete Objekte [...], die sich in regelmäßigen oder in unregelmäßigen zeitlichen Abständen über Transportwege oder Förderstrecken bewegen“ (Arnold und Furmans 2009, S. 1). In diesem Zusammenhang wird der Begriff der Senke gebraucht, um das Ende eines Transportweges oder Förderstrecke zu beschreiben. Die *Materialflusslehre* befasst sich mit der Modellierung der Materialflüsse und der Beantwortung von Fragestellungen zum Zustand, zum Durchsatz oder der Zuverlässigkeit von Materialflusssystemen (vgl. Arnold und Furmans 2009, S. 2f). Ein wesentliches Unterscheidungskriterium zwischen Stoffströmen und Materialflüssen im Sinne dieser Arbeit ist die räumliche und zeitliche Granularität, mit welcher die Stoffe modelliert werden. Während für Stoffströme die exakten Standorte (z. B. in Form von Koordinaten) und Wege durch ein Gebäude oder eine Maschine nicht relevant sind, müssen diese für Materialflüsse in der Regel detailliert nachvollzogen werden. Und

während die zeitliche Dimension von Stoffströmen eher in groben Einheiten, wie Minuten, Stunden oder Tage bemessen wird, können bei der Planung von Materialflüssen Sekunden oder Sekundenbruchteile ausschlaggebend sein. Die Unterschiede in der Granularität sind im Wesentlichen darauf zurückzuführen, dass Materialflüsse als primäres Untersuchungsobjekt dienen, wohingegen Stoffströme lediglich eine Art „Rahmenbedingung“ einer anderweitigen Problemstellung beschreiben. Beispielsweise wird mit Hilfe der Warteschlangentheorie primär untersucht, wie sich Materialflüsse in einem Wartesystem verhalten (vgl. Arnold und Furmans 2009, S. 112ff). Im Gegensatz dazu befasst sich die Produktionsablaufplanung primär mit der Planung herzustellender Produkte unter Berücksichtigung der Stücklisten und Arbeitspläne, die Informationen zu den hervorgerufenen Stoffströmen bereitstellen (vgl. z. B. Günther 1992).

Materialflüsse sind nicht Gegenstand dieser Arbeit, da das zu entwickelnde Optimierungswerkzeug auf der Ebene der Projektplanung angesiedelt ist und keine logistischen Fragestellungen beantwortet. In Abschnitt 2.4.2 wird die Rolle von Stoffströmen im produktionswirtschaftlichen Umfeld ausführlich beleuchtet.

### 2.1.3 Probleme, Verfahren und Modelle

Das *Operations Research (OR)* wurde in Abschnitt 2.1.1 bereits als interdisziplinäres Feld zur Unterstützung von Entscheidungen vorgestellt. Briskorn (2020, S. 4) vergleicht das Treffen einer Entscheidung mit dem Beantworten einer Frage. Die Frage, welche die Gesamtheit aller Informationen zur Entscheidungssituation umfasst, wird als *Problem* bezeichnet. Im OR werden nach Art der Ausgabe zwei Arten von *mathematischen Problemen* unterschieden (Briskorn 2020, S. 6): *Entscheidungsprobleme* beantworten eine binäre Frage der Art „ja/nein“. *Optimierungsprobleme* beantworten die Frage nach der optimalen Ausprägung einer *Zielsetzung*. Im Folgenden werden die Bestandteile mathematischer Probleme angelehnt an Briskorn (2020, S. 5f) erläutert. Die zur Beantwortung der Frage zu bestimmenden Informationen werden mittels *Entscheidungsvariablen* quantifiziert, deren mögliche Ausprägungen als *Lösungsraum* bezeichnet werden. Der Begriff der Entscheidung bezieht sich hierbei nicht auf die Beantwortung der Frage, sondern auf den Prozess, der zur Beantwortung führt. Deshalb sind Entscheidungsvariablen sowohl in Entscheidungsproblemen als auch Optimierungsproblemen vorhanden. Die Zielsetzung lässt sich als eine Funktion der Entscheidungsvariablen, der sogenannten *Zielfunktion*, mit einer Angabe zur Präferenz hoher oder niedriger Werte, abbilden. In Entscheidungsproblemen sind Zielsetzungen irrelevant, da es sich hierbei um das Unterscheidungskriterium zu Optimierungsproblemen handelt. Eine konkrete Ausprägung der Informationen zur Beantwortung der Frage – formal also eine Belegung der Entscheidungsvariablen mit Werten – wird als *Lösung* bezeichnet. Der dazugehörigen Funktionswert der Zielfunktion ist der *Zielfunktionswert*. Eine Lösung eines Optimierungsproblems ist eine *optimale Lösung*, eine *exakte Lösung* oder ein *Optimum*, falls es keine andere Lösung gibt, deren Zielfunktionswert besser ist. Wenn eine Lösung den Gegebenheiten der Entscheidungssituation widerspricht, wird sie als *unzulässig* bezeichnet. Andernfalls wird eine Lösung implizit als *zulässig* betrachtet. Die Gegebenheiten der Entscheidungssituation werden quantitativ

über *Restriktionen* ausgedrückt, die formale Beziehungen zwischen verschiedenen Entscheidungsvariablen vorschreiben (z. B. Entscheidungsvariable  $y$  muss immer doppelt so groß wie Entscheidungsvariable  $x$  sein). Der Prozess zur Bestimmung einer Lösung wird als das *Lösen* bezeichnet. In dieser Arbeit wird unter dem Lösen eines Optimierungsproblems nicht implizit das exakte Lösen, d. h. die Bestimmung einer optimalen Lösung, verstanden, sondern die Bestimmung einer zulässigen Lösung.

Neben den Entscheidungsvariablen und der Zielsetzung besteht ein Problem aus *Problemparametern*, die die Datenstruktur zu sämtlichen gegebenen Informationen einer Entscheidungssituation bilden. In der Literatur wird hier oft auch lediglich von Parametern gesprochen (z. B. Briskorn 2020, S. 28). In dieser Arbeit wird zur besseren Abgrenzung von anderen Parametern (Instanzparameter und Verfahrensparameter; siehe Kapitel 6) jedoch der spezifischere Begriff der Problemparameter verwendet. Eine Belegung aller Problemparameter mit Werten beschreibt eine konkrete Ausprägung eines Problems und wird deshalb als *(Problem-)Instanz* bezeichnet. Die als gegeben betrachteten Problemparameter sind möglicherweise ebenfalls Ergebnis einer zuvor getroffenen Entscheidung, die allerdings nicht in den Untersuchungsrahmen des behandelten Problems fällt. Da im weiteren Verlauf der Arbeit eine klare sprachliche Abgrenzung erforderlich ist, werden die außerhalb des Untersuchungsrahmens eines Problems getroffenen Entscheidungen als *modellexogen* bezeichnet. Im Gegensatz dazu werden die innerhalb des Untersuchungsrahmens eines Problems getroffenen Entscheidungen als *modellendogen* bezeichnet.

Probleminstanzen können Werte enthalten, die ein Problem *unlösbar* machen, weil sie Restriktionen unabhängig von der zu treffenden Entscheidung verletzen (z. B. ein Vorgang  $i$  soll nach einem Vorgang  $i'$  starten und Vorgang  $i'$  soll nach Vorgang  $i$  starten). Solche Probleminstanzen werden, analog zu Lösungen, die Restriktionen verletzen, als *unzulässig* bezeichnet. Davon abgesehen ist es auch möglich, dass ein Problem *generell unlösbar* ist, wenn sich Restriktionen unabhängig von der zu treffenden Entscheidung und unabhängig von der Probleminstanz widersprechen (z. B. Entscheidungsvariable  $x$  muss immer doppelt so groß wie Entscheidungsvariable  $y$  sein und  $x$  muss immer kleiner als  $y$  sein).

Da die Durchführung eines Projekts mit Zielen verbunden ist (vgl. Abschnitt 2.1.1), werden im Folgenden nur noch Optimierungsprobleme betrachtet. Der Begriff des Problems wird somit als Synonym zum Optimierungsproblem gebraucht. Zusammengefasst ist ein mathematisches Problem definiert durch eine Zielsetzung bestehend aus Zielfunktion und Angabe zur Präferenz hoher oder niedriger Werte, Entscheidungsvariablen, Problemparametern und Restriktionen. Eine Probleminstanz ist die Eingabe zu einem Problem und besteht aus einer Belegung aller Problemparameter. Eine Lösung ist die Ausgabe zu einem Problem und besteht aus einer Belegung aller Entscheidungsvariablen und einem Zielfunktionswert. Es wird angenommen, dass die Zurverfügungstellung der Eingabe und die damit verbundenen modellexogenen Entscheidungen, sowie die Auswertung und Umsetzung der Ausgabe der Verantwortlichkeit einer oder mehrerer natürlicher Personen unterliegen. Diese Personen werden im Folgenden als *Projektverantwortliche* bezeichnet.

Im OR werden Methoden zum Lösen von Problemen erforscht, sogenannte *Lösungsverfahren*. Unterschieden werden dabei *optimale* oder *exakte Lösungsverfahren*, die eine optimale

Lösung liefern, *Approximationsverfahren*, die Lösungen nahe am Optimum liefern (ausgedrückt in einem bestimmten Verhältnis) und *Heuristiken*, die eine Lösung liefern, „deren Wert potentiell beliebig schlecht“ im Vergleich zum Optimum sein kann (Briskorn 2020, S. 10). Letztere können allerdings dennoch so konstruiert werden, dass sie für alle oder für praktisch relevante Instanzen sehr gute Lösungen bei vergleichsweise geringem Rechenaufwand liefern (Briskorn 2020, S. 11). Eine präzise und eindeutige Beschreibung eines Verfahrens oder Verfahrensbestandteils ist ein *Algorithmus* (Briskorn 2020, S. 11).

Die Überführung eines realen Problems in ein formales Problem geht mit einer Abstrahierung der Wirklichkeit einher (Briskorn 2020, S. 15ff). Dieser Prozess wird als *Modellierung* bezeichnet und dessen Ergebnis als (*Optimierungs-*)*Modell*. Als Modellierung kann außerdem die Formalisierung eines spezifischen Anwendungsfalls verstanden werden, dessen Ergebnis dann die Probleminstanz ist. Nach Briskorn (2020, S. 16) ist ein Modell allgemein eine zielorientierte Abbildung der Eigenschaften des Originals. Für das OR sind *mathematische Modelle* relevant, die als Gesamtheit der einem Problem zugrundeliegenden Strukturen und Konzepte verstanden werden können. Vereinfacht ausgedrückt sind mathematische Modelle eine Sprache und Probleme im Sinne des OR konkrete Aussagen, die sich dieser Sprache bedienen. Die Problemformulierung geht demnach mit einer Modellbeschreibung einher.

Wie bereits erläutert, befasst sich ein Teilgebiet des OR mit dem Scheduling. In dieses Teilgebiet lässt sich auch die vorliegende Arbeit einordnen. Daneben gibt es zahlreiche weitere Teilgebiete, die sich beispielsweise mit der Optimierung von Wegen, Touren, Netzwerkflüssen, Packungen, Zuschnitten, Produktionsprogrammen, etc. befassen.

## 2.2 Schedulingprobleme

In diesem Abschnitt werden verschiedene Schedulingprobleme sowie die zugrundeliegenden Modellbestandteile erläutert und schrittweise erweitert. Bei einem *Schedulingproblem* handelt es sich um ein mathematisches Problem, welches sich mit einer Fragestellung aus dem Bereich der Projektablaufplanung befasst.

### 2.2.1 Allgemeines Schedulingproblem

Um den zeitlichen Ablauf eines Projekts zu planen, ist zunächst eine Unterteilung der auszuführenden Arbeiten in (*Projekt-*)*Vorgänge* erforderlich. Die DIN ISO 21500 beschreibt einen Vorgang als „festgelegte Arbeitsaufgabe (Arbeitspaket) im Rahmen eines Plans, [die] erforderlich für die Vollendung eines Projekts [ist]“ (DIN Deutsches Institut für Normung 2016, S. 7). Dem Definieren von Vorgängen geht die Erstellung eines Projektstrukturplans voraus, der die auszuführenden Arbeiten hierarchisch gliedert (DIN Deutsches Institut für Normung 2016, S. 27). Die Vorgänge werden basierend auf der untersten Ebene dieser Hierarchie abgeleitet. Hinsichtlich des Untersuchungsrahmens eines Schedulingproblems handelt es sich beim Definieren der Vorgänge um eine modellexogene Entscheidung.

Projektverantwortliche müssen sich dabei insbesondere mit dem erforderlichen Detaillierungsgrad befassen, da Vorgänge je nach Arbeitsaufgabe, die sie beschreiben, theoretisch beliebig oft aufgespalten werden können. Die Entscheidung zum Detaillierungsgrad hängt von verschiedenen Faktoren ab, beispielsweise inwiefern eine zeitliche Unterbrechung der Ausführung von Vorgängen möglich ist, inwiefern sich die logische Beziehung zu anderen Vorgängen abbilden lässt oder inwiefern zur Ausführung benötigte Ressourcen eindeutig zugeordnet werden können.

Zur Problemformulierung werden die Vorgänge eines Projekts mit natürlichen Zahlen durchnummeriert, wobei die gewählte Reihenfolge die Allgemeinheit des Modells nicht beschränkt. Ein Projekt ist somit in die Vorgänge  $i = 1, \dots, I$  unterteilt. Die Menge der Vorgänge ist ein Problemparameter und wird als  $V := \{1, \dots, I\}$  bezeichnet. Neben der Variable  $i$  werden im weiteren Verlauf der Arbeit auch  $i', i_1, i_2, \dots$  verwendet, um Vorgänge zu indizieren. Problemparameter  $I$  entspricht der Anzahl an Vorgängen im Projekt, wobei  $I \geq 1$  gilt. In natürlicher Sprache lässt sich das *allgemeine Schedulingproblem (PSP)* (engl. project scheduling problem) nun folgendermaßen beschreiben:

<p style="margin: 0;">Wie plane ich ein Projekt, welches in die Vorgänge  <math>i = 1, \dots, I</math> unterteilt wurde, sodass eine gegebene          Zielsetzung erreicht wird und alle Restriktionen          eingehalten werden?</p>	}	(PSP)
--	---	-------

Zimmermann u. a. (2006, S. 39) definieren Vorgänge im Sinne des Scheduling als „zeitbeanspruchende Geschehen mit ausgezeichneten Start- und Endereignissen, bei deren Ausführung in der Regel Ressourcen (z. B. Arbeitskräfte, Maschinen, Material) ge- oder verbraucht und Kosten verursacht werden“. Die durch einen Vorgang  $i$  bei seiner Ausführung beanspruchte Zeit wird als (*Vorgangs-*)*Dauer*  $d_i$  bezeichnet. Es gilt  $d_i \in \mathbb{R}_{\geq 0}$  für alle  $i \in V$ . Bei den Vorgangsdauern handelt es sich um Problemparameter, d. h., sie werden modellexogen von Projektverantwortlichen festgelegt. Ein *Ereignis* „kennzeichnet das Erreichen eines bestimmten Projektzustands“ (Zimmermann u. a. 2006, S. 41). Neben den Start- und Endereignissen jedes Vorgangs (kurz auch einfach als Start und Ende des Vorgangs bezeichnet) sind der *Projektstart* und das *Projektende* zwei weitere relevante Ereignisse. Grundsätzlich kann jeder Zeitpunkt während der Projektausführung als Ereignis betrachtet werden. Ereignisse unterscheiden sich von Vorgängen, da sie nicht zeitbeanspruchender, sondern augenblicklicher Art sind. Der zeitliche Abstand zwischen Projektstart und Projektende ist die *Projektdauer*.

Unter der „Planung eines Projektes“, wie im (PSP) formuliert, wird im Scheduling die zeitliche Anordnung oder *Terminierung* der Vorgänge verstanden. Gesucht sind somit Angaben zum Startzeitpunkt, zum Endzeitpunkt und zum Ausführungszeitraum jedes Vorgangs. Da Startzeitpunkt, Endzeitpunkt und Ausführungszeitraum in einer festen formalen Beziehung zueinander stehen, ist es ausreichend, lediglich den Startzeitpunkt  $S_i$  jedes Vorgangs  $i = 1, \dots, I$  als Entscheidungsvariable zu definieren. Der Endzeitpunkt entspricht der Summe aus Startzeitpunkt  $S_i$  und Dauer  $d_i$ . Der Ausführungszeitraum entspricht dem Intervall von Startzeitpunkt bis Endzeitpunkt, d. h.  $[S_i, S_i + d_i)$ . Es ist zu beachten, dass wie in allen gängigen Problemformulierungen der Endzeitpunkt selbst nicht mehr zum

Ausführungszeitraum gezählt wird. Dies ist damit begründet, dass per Konvention folgende eindeutige Beziehung zwischen diskreten Zeitpunkten und kontinuierlichen Zeiträumen definiert wird (Neumann, Schwindt und Zimmermann 2003, S. 10): Der Zeitraum zwischen zwei aufeinanderfolgenden Zeitpunkten  $t$  und  $t + 1$  entspricht dem halboffenen Zeitintervall  $[t, t + 1)$ . Ein Vektor  $S = (S_i)_{i \in V}$  aller mit Werten belegten Startzeitpunkte wird als *Schedule* bezeichnet. Da das allgemeine Schedulingproblem neben den Startzeitpunkten keine weiteren Entscheidungsvariablen enthält, ist ein *Schedule* gleichzeitig eine Lösung des Schedulingproblems. Beide Begriffe werden in diesem Fall synonym verwendet. Der dem *Schedule* entsprechende deutschsprachige Begriff des *Ablaufplans* wird nur noch verwendet, sofern kein methodischer Kontext vorliegt.

Dauern und Startzeitpunkte können, sofern es einen gemeinsamen Teiler gibt, als ganze Zahlen ausgedrückt werden, indem alle Zeitangaben auf ein ganzzahliges Vielfaches ihres größten gemeinsamen Teilers hochskaliert werden. Diese *Diskretisierung* der Zeit ist zur Anwendung vieler Lösungsverfahren erforderlich und entspricht in der Regel auch der Realität, wo Zeit mit ganzzahligen Einheiten, wie Tagen oder Stunden, bemessen wird (vgl. Neumann, Schwindt und Zimmermann 2003, S. 4). Für die Problemformulierungen in dieser Arbeit wird die Zeit dennoch naturgemäß als kontinuierlich betrachtet. Zur Messung von zeitlichen Abständen wird der nicht näher spezifizierte Begriff der *Periode* verwendet. Ohne Beschränkung der Allgemeinheit können die Startzeitpunkte als positive reelle Zahlen ausgedrückt werden, weshalb  $S_i \geq 0$  für alle  $i \in V$  gilt. Das Zeitintervall  $[0, \infty)$ , in dem sich die Startzeitpunkte bewegen dürfen, wird als *Zeithorizont* bezeichnet.

Die Zielsetzung wird im allgemeinen Schedulingproblem durch eine generische Zielfunktion  $z(S)$  ausgedrückt, wobei niedrigere Werte präferiert werden. Es handelt sich folglich um ein *Minimierungsproblem*. Maximierungsprobleme können in Minimierungsprobleme transformiert werden, da das Minimieren einer Funktion äquivalent zum Maximieren der negativen Funktion ist (Briskorn 2020, S. 87). Die Restriktionen werden ebenfalls generisch formuliert: Gegeben sei eine Menge  $\mathcal{S}$  aller zulässigen Schedules. Für eine (zulässige) Lösung  $S$  des Problems muss  $S \in \mathcal{S}$  gelten (in Anlehnung an Franck u. a. 2001). Die eingeführte Notation ermöglicht nun eine formale Beschreibung des (PSP):

$$\left. \begin{array}{ll} \min_S & z(S) \\ \text{u. d. R.} & S \in \mathcal{S} \end{array} \right\} \quad (\text{PSP})$$

Die Abkürzung „u. d. R.“ bedeutet „unter den Restriktionen“.

Einige Literaturquellen verwenden auch eine Formulierung, die die Endzeitpunkte der Vorgänge ebenfalls als Entscheidungsvariable in das Schedulingproblem integrieren (z. B. Schwindt und Trautmann 2000; Carlier, Moukrim und Xu 2009). Diese Formulierung würde die Notation von Restriktionen im Zusammenhang mit Stoffströmen etwas flexibler gestalten, da sie auf zusätzliche Ereignisse, d. h. die Ereignisse der Endzeitpunkte von Vorgängen, erweitert wurde. Dennoch wird in dieser Arbeit die ausschließlich auf den Startzeitpunkten basierende Formulierung (PSP) verwendet, da sie weiter verbreitet und intuitiver ist. Beide Formulierungen lassen sich ineinander transformieren.



In den nachfolgenden Abschnitten wird das allgemeine Schedulingproblem (PSP) konkretisiert, indem die generische Restriktion  $S \in \mathcal{S}$  durch spezifische Restriktionen ersetzt wird. Diese spezifischen Restriktionen werden von Rahmenbedingungen im zu planenden Projekt abgeleitet. Die generische Zielfunktion wird jedoch beibehalten und erst in Abschnitt 2.2.8 spezifiziert. Damit wird demonstriert, dass sich die Problemformulierungen leicht auf verschiedene Zielsetzungen übertragen lassen. Lediglich zur Angabe einer Komplexitätsklasse für die Optimierungsprobleme wird eine konkrete Zielsetzung benötigt. Hierfür wird von der Minimierung der Projektdauer ausgegangen, da es sich dabei um eine der am häufigsten in der Literatur betrachteten Zielsetzungen handelt (vgl. Abschnitt 2.2.8).

### 2.2.2 Scheduling unter Vorrangrestriktionen

Aus technischen, logischen oder ablauforganisatorischen Gründen bestehen zwischen Vorgängen *Vorrangbeziehungen* (Zimmermann u. a. 2006, S. 42ff). Die Menge aller Vorrangbeziehungen  $E$  ist eine binäre Relation auf  $V$ , d. h.  $E \subset V^2$ . Eine Vorrangbeziehung  $(i, i') \in E$  besagt, dass Vorgang  $i'$  frühestens zum Endzeitpunkt  $(S_i + d_i)$  des Vorgangs  $i$  starten darf, wobei  $i \neq i'$ . Da Vorrangbeziehungen feste zeitliche Reihenfolgen für Vorgänge induzieren, können für jeden Vorgang  $i \in V$  folgende beiden Mengen definiert werden: Die Menge der (*direkten*) *Vorgänger*

$$\text{Pred}(i) := \{i' \in V \mid \exists(i', i) \in E\} \quad (2.1)$$

und analog die Menge der (*direkten*) *Nachfolger*

$$\text{Succ}(i) := \{i' \in V \mid \exists(i, i') \in E\}. \quad (2.2)$$

Die als Menge  $E$  gegebenen Vorrangbeziehungen implizieren *Vorrangrestriktionen* für das Scheduling, die als Ungleichungen

$$S_{i'} \geq S_i + d_i \quad (\forall(i, i') \in E) \quad (2.3)$$

formuliert werden. Ein Schedule  $S$ , für den die Ungleichungen (2.3) erfüllt sind, wird als *vorrangzulässig* bezeichnet.

Für eine Instanz des Schedulingproblems unter Vorrangrestriktionen werden per Konvention zwei zusätzliche Vorgänge 0 und  $I + 1$  zur Menge  $V$  hinzugefügt. Sie werden als *fiktiver Start-* bzw. *Endvorgang* bezeichnet, da sie keine Dauer haben ( $d_0 = d_{I+1} := 0$ ). Aus diesem Grund handelt es sich strenggenommen bei den fiktiven Vorgängen um Ereignisse, die allerdings aus Formulierungsgründen dennoch als Vorgänge modelliert werden (vgl. Zimmermann u. a. 2006, S. 42). Die Vorgänge  $i = 1, \dots, I$  werden zur expliziten Abgrenzung als *reale Vorgänge* bezeichnet. Der Vektor  $S$  umfasst nun neben den Startzeitpunkten der realen Vorgänge auch die Startzeitpunkte der fiktiven Vorgänge:  $S := (S_0, S_1, \dots, S_I, S_{I+1})$ . Die fiktiven Vorgänge haben lediglich methodische Relevanz. Sie sind ein Modellbestandteil, der den Projektstart als  $S_0$  und das Projektende als  $S_{I+1}$  für Problemformulierungen und Algorithmen zugänglich macht. Dazu muss allerdings sichergestellt werden, dass

Vorgang 0 in jedem Fall der früheste Vorgang des Projekts und Vorgang  $I + 1$  in jedem Fall der späteste Vorgang des Projekts ist. Dies erfolgt über zusätzliche, fiktive Vorrangbeziehungen, die bei der Erstellung der Problem Instanz auf folgende Art generiert werden: Für jeden Vorgang  $i$  mit  $Pred(i) = \emptyset$  wird eine Vorrangbeziehung  $(0, i)$  zu  $E$  hinzugefügt. Für jeden Vorgang  $i$  mit  $Succ(i) = \emptyset$ , wird eine Vorrangbeziehung  $(i, I + 1)$  zu  $E$  hinzugefügt.

Das allgemeine Schedulingproblem (PSP) kann nun als *Schedulingproblem unter Vorrangrestriktionen* (TPSP) (engl. temporal project scheduling problem) spezifiziert werden, indem die Ungleichungen (2.3) als Restriktionen aufgefasst werden. Darüber hinaus wird  $S_i \geq 0$  explizit als Restriktion für alle Vorgänge aufgenommen, wobei der Projektstart  $S_0$  ohne Beschränkung der Allgemeinheit auf den Zeitpunkt 0 fixiert wird. Die formale Beschreibung des (TPSP) lautet somit:

$$\left. \begin{array}{ll} \min_S & z(S) \\ \text{u. d. R.} & S_{i'} \geq S_i + d_i \quad (\forall (i, i') \in E); \\ & S_0 = 0; \\ & S_i \geq 0 \quad (i = 1, \dots, I + 1). \end{array} \right\} \quad (\text{TPSP})$$

Bei der Menge  $E$  und den Dauern  $d_i$  für alle Vorgänge  $i = 0, \dots, I + 1$  handelt es sich um Problemparameter. Das (TPSP) befasst sich ausschließlich mit dem Planungsbereich der Terminplanung.

Instanzen des (TPSP) können in Form eines (*Vorgangsknoten-*)netzplans  $N = (V, E)$  dargestellt werden, wobei es sich hierbei um einen *bewerteten Digraphen* handelt. Die Vorgänge  $V$  werden als Knoten abgebildet und die Vorrangbeziehungen  $(i, i') \in E$  werden als gerichtete und mit  $d_i$  gewichtete Kanten von  $i$  nach  $i'$  abgebildet. Knoten und Kanten spiegeln die zugehörigen Vorgänge bzw. Vorrangbeziehungen, weshalb die identische Notation verwendet wird. Die Länge eines Pfades von Knoten  $i$  zu Knoten  $i'$  in  $N$  induziert einen zeitlichen Mindestabstand zwischen den Startzeitpunkten der Vorgänge  $i$  und  $i'$  (Neumann und Schwindt 2002, S. 8). Der längste Pfad zwischen dem fiktiven Startvorgang und dem fiktiven Endvorgang wird als *kritischer Pfad* bezeichnet. Wenn in  $N$  ein Pfad von einem Knoten  $i$  zu einem anderen Knoten  $i'$  existiert, ist  $i'$  ein *transitiver Nachfolger* von  $i$ . Formal lässt sich die Menge  $\overline{Succ}(i)$  der transitiven Nachfolger eines Vorgangs  $i$  rekursiv mit

$$\overline{Succ}(i) := \begin{cases} \bigcup_{i' \in Succ(i)} (\{i'\} \cup \overline{Succ}(i')) & \text{falls } Succ(i) \neq \emptyset, \\ \emptyset & \text{sonst} \end{cases} \quad (2.4)$$

definieren und berechnen. Analog dazu ist die Menge  $\overline{Pred}(i)$  der transitiven Vorgänger definiert. Wenn nicht explizit von transitiven Vorgängern oder Nachfolgern gesprochen wird, sind stets die direkten Vorgänger  $Pred(i)$  bzw. Nachfolger  $Succ(i)$  gemeint.

Die *Metra-Potential-Methode* befasst sich mit der Modellierung und Lösung des (TPSP) mit Hilfe von Vorgangsknotennetzplänen (Zimmermann u. a. 2006, Abschnitt 1.4.3), falls die Projektdauer minimiert werden soll. Als Alternative dient die *Kritischer-Pfad-Methode*, wobei hierfür jedoch *Vorgangskantennetzpläne* verwendet werden (Kelley 1961). Für weitere Details und Angaben zu geeigneten Algorithmen wird auf den nächsten Abschnitt verwiesen, welcher eine Verallgemeinerung der Vorrangbeziehungen behandelt.

### 2.2.3 Scheduling unter Zeitrestriktionen

Das Modell der *Zeitbeziehungen* verallgemeinert die im vorangegangenen Abschnitt eingeführten Vorrangbeziehungen. Eine Zeitbeziehung zwischen zwei Vorgängen  $i, i' \in V$  wird durch eine Anordnungsbeziehung  $(i, i')$  und einen Zeitabstand  $\delta_{ii'} \in \mathbb{R}$  spezifiziert (Zimmermann u. a. 2006, S. 42). Es gilt  $i \neq i'$ . Die Menge aller Anordnungsbeziehungen  $E^{temp}$  ist eine binäre Relation auf  $V$ , d. h.  $E^{temp} \subset V^2$ . Die Zeitabstände werden als Matrix  $\delta := (\delta_{ii'})_{i, i' \in V; i \neq i'}$  zusammengefasst. Zimmermann u. a. (2006, S. 42) unterscheiden für die Anordnungsbeziehungen vier verschiedene *Verknüpfungstypen*, die angeben, ob eine Anordnungsbeziehung jeweils den Start oder das Ende des Vorgangs  $i$  mit dem Start oder dem Ende des Vorgangs  $i'$  verknüpft: Ende-Start-, Start-Start-, Start-Ende- und Ende-Ende-Beziehungen. Da sich die verschiedenen Verknüpfungstypen ineinander transformieren lassen (vgl. Bartusch u. a. 1988; Zimmermann u. a. 2006, S. 43f), ist es ausreichend, die Start-Start-Beziehungen als standardmäßigen Verknüpfungstyp für alle Anordnungsbeziehungen anzunehmen. Start-Start-Beziehungen lassen sich am besten mit den verwendeten Entscheidungsvariablen  $S_i$  ausdrücken.

Die als Tupel  $(E^{temp}, \delta)$  gegebenen Zeitbeziehungen implizieren *Zeitrestriktion* für das Scheduling, die als Ungleichungen

$$S_{i'} \geq S_i + \delta_{ii'} \quad (\forall (i, i') \in E^{temp}) \quad (2.5)$$

formuliert werden. Ein Schedule, der alle Zeitrestriktionen einhält, wird als *zeitzulässig* bezeichnet. Die Ungleichungen (2.5) ermöglichen es, nach folgendem Schema beliebige zeitliche Mindest- oder Höchstabstände zwischen den Vorgängen zu modellieren (Neumann, Schwindt und Zimmermann 2003, S. 7):

- Wenn zwischen zwei Vorgängen  $i, i' \in V$  (genauer gesagt, zwischen den Startzeitpunkten zweier Vorgänge) ein zeitlicher Mindestabstand  $d_{ii'}^{min}$  existiert, wird eine Anordnungsbeziehung  $(i, i')$  zur Menge der Anordnungsbeziehungen  $E^{temp}$  hinzugefügt und  $\delta_{ii'} := d_{ii'}^{min}$  gesetzt. Die entsprechende Ungleichung (2.5) bewirkt, dass der Startzeitpunkt  $S_{i'}$  mindestens so groß sein muss, wie die Summe aus dem Startzeitpunkt  $S_i$  und  $\delta_{ii'}$ . Falls zwischen den Vorgängen  $i$  und  $i'$  mehrere zeitliche Mindestabstände existieren, wird nur der größte dieser zeitlichen Mindestabstände herangezogen.
- Wenn zwischen zwei Vorgängen  $i, i' \in V$  ein zeitlicher Höchstabstand  $d_{ii'}^{max}$  existiert, wird eine Anordnungsbeziehung  $(i', i)$  zur Menge der Anordnungsbeziehungen  $E^{temp}$  hinzugefügt und  $\delta_{i'i} := -d_{ii'}^{max}$  gesetzt. Die entsprechende Ungleichung (2.5) bewirkt, dass der Startzeitpunkt  $S_{i'}$  höchstens so groß sein darf, wie die Summe aus dem Startzeitpunkt  $S_i$  und  $|\delta_{i'i}|$ . Falls zwischen den Vorgängen  $i$  und  $i'$  mehrere zeitliche Höchstabstände bestehen, wird nur der kleinste dieser zeitlichen Höchstabstände herangezogen.

Bei der Erzeugung der fiktiven Vorgänge 0 und  $I + 1$  werden fiktive Zeitbeziehungen nach folgendem Schema erzeugt (Neumann, Schwindt und Zimmermann 2003, S. 2f): Für jeden Vorgang  $i$ , für den es keinen anderen Vorgang  $i'$  mit zeitlichem Mindestabstand  $d_{i'i}^{min}$  gibt,

wird ein zeitlicher Mindestabstand  $d_{0i}^{min} := 0$  eingeführt. Für jeden Vorgang  $i$ , für den es keinen anderen Vorgang  $i'$  mit zeitlichem Mindestabstand  $d_{ii'}^{min}$  gibt, wird ein zeitlicher Mindestabstand  $d_{i,I+1}^{min} := d_i$  eingeführt. Letzterer stellt sicher, dass das Projektende nicht vor dem Abschluss des Vorgangs  $i$  liegen kann. Es ist bei Zeitbeziehungen außerdem möglich, dass für einen Vorgang  $i$  zwar Vorgänge  $i'$  mit zeitlichem Mindestabstand  $d_{ii'}^{min}$  existieren, aber  $d_{ii'}^{min} + d_{i'} < d_i$  für alle  $i'$  ist. Um dennoch sicherzustellen, dass das Projektende nicht vor dem Abschluss des Vorgangs  $i$  liegen kann, muss in solchen Fällen ebenfalls ein zeitlicher Mindestabstand  $d_{i,I+1}^{min} := d_i$  eingeführt werden.

Zeitbeziehungen ermöglichen die Modellierung vieler weiterer zeitlicher Vorgaben eines Projekts: Durch Einführung eines zeitlichen Höchstabstands  $d_{0,I+1}^{max}$  kann eine *maximale Projektdauer* modelliert werden. Außerdem ist eine Modellierung von Zeitfenstern oder von fixen Startzeitpunkten für Vorgänge sowie von fixen zeitlichen Abständen zwischen verschiedenen Vorgängen möglich (siehe Zimmermann u. a. 2006, S. 58). Vorrangbeziehungen  $(i, i') \in E$  erfordern per Definition einen zeitlichen Mindestabstand  $d_{ii'}^{min} := d_i$ . Somit können sie problemlos als Zeitbeziehungen ausgedrückt werden und es gilt dann  $E \subseteq E^{temp}$ .

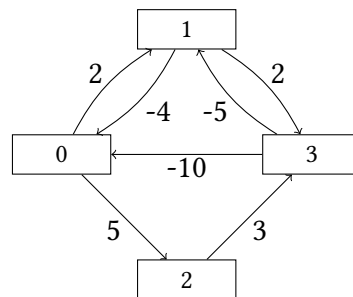
Das Schedulingproblem unter Vorrangrestriktionen (TPSP) kann nun als *Schedulingproblem unter Zeitrestriktionen (TPSP/max)* generalisiert werden, indem Ungleichungen (2.5) an die Stelle der Vorrangrestriktionen treten. Das „max“ in der Problembezeichnung stammt von den „maximum time lags“, d. h. den zeitlichen Höchstabständen, die durch Zeitbeziehungen abgebildet werden können und ist angelehnt an die Bezeichnung in der Schedulingliteratur (z. B. Neumann und Schwindt 1997). Die formale Beschreibung des (TPSP/max) lautet:

$$\left. \begin{array}{ll} \min_S & z(S) \\ \text{u. d. R.} & S_{i'} \geq S_i + \delta_{ii'} \quad (\forall (i, i') \in E^{temp}); \\ & S_0 = 0; \\ & S_i \geq 0 \quad (i = 1, \dots, I + 1). \end{array} \right\} \quad (\text{TPSP/max})$$

Bei der Menge der Anordnungsbeziehungen  $E^{temp}$  und den Zeitabständen  $\delta$  handelt es sich um Problemparameter, die von den zeitlichen Mindest- und Höchstabständen abgeleitet werden. Das (TPSP/max) befasst sich ausschließlich mit dem Planungsbereich der Terminplanung.

Instanzen des (TPSP/max) können in Form eines Vorgangsknotennetzplans  $N^{temp} = (V, E^{temp}, \delta)$  dargestellt werden. Jeder Vorgang  $i \in V$  wird als Knoten abgebildet und jede Anordnungsbeziehung  $(i, i') \in E^{temp}$  wird als gerichtete und mit  $\delta_{ii'}$  bewertete Kante von  $i$  nach  $i'$  abgebildet. Knoten und bewertete Kanten spiegeln direkt die zugehörigen Vorgänge bzw. Zeitbeziehungen, weshalb die identische Notation verwendet wird. Die Netzpläne dienen der Veranschaulichung der Zeitstruktur einer Instanz des (TPSP/max). Wie auch in  $N$ , d. h. dem Netzplan für Schedulingprobleme mit Vorrangrestriktionen, induziert ein Pfad positiver Länge von Knoten  $i$  zu Knoten  $i'$  in  $N^{temp}$  einen zeitlichen Mindestabstand zwischen den Startzeitpunkten der Vorgänge  $i$  und  $i'$  (Neumann und Schwindt 2002, S. 8). Umgekehrt induziert ein Pfad negativer Länge von Knoten  $i$  zu Knoten  $i'$  einen zeitlichen Höchstabstand zwischen den Startzeitpunkten der Vorgänge  $i'$  und  $i$  (Neumann

und Schwindt 2002, S. 8). Ein Pfad, dessen Start- und Endknoten übereinstimmen, wird als *Zyklus* bezeichnet (Zimmermann u. a. 2006, S. 61). Wenn in  $N$  oder  $N^{temp}$  ein Zyklus positiver Länge existiert, würde das für jeden Vorgang im Zyklus bedeuten, dass er in Bezug auf sich selbst mit einem zeitlichen Mindestabstand starten muss. Dies ist ein Widerspruch, weshalb *Zyklen positiver Länge* in einem Vorgangsknotennetzplan zu erkennen geben, dass die Instanz unzulässig ist (Neumann und Schwindt 2002, S. 10). Abbildung 2.1 zeigt einen beispielhaften Netzplan eines Projekts mit  $I = 2$  realen Vorgängen. Der längste Pfad ist  $(0, 2, 3)$ , weshalb es sich hierbei um den kritischen Pfad handelt. Seiner Länge entsprechend beträgt die kritische Projektdauer acht Perioden. Zwischen den fiktiven Vorgängen 0 und 3 besteht ein zeitlicher Höchstabstand, der gleichzeitig eine maximale Projektdauer von zehn Perioden vorgibt. Der Start von Vorgang 1 ist auf das Zeitfenster  $[3, 4]$  begrenzt, da der Pfad  $(0, 2, 3, 1)$  einen frühestmöglichen Startzeitpunkt von drei und der Pfad  $(1, 0)$  einen spätestmöglichen Startzeitpunkt von vier induziert. Es handelt sich um eine zulässige Instanz, da keine Zyklen positiver Länge vorliegen.



**Abbildung 2.1:** Beispiel für einen Vorgangsknotennetzplan mit zeitlichen Höchst- und Mindestabständen.

Die in den Gleichungen (2.1) und (2.2) definierten Mengen der direkten Vorgänger und Nachfolger lassen sich nicht ohne Weiteres auf das Konzept der Zeitbeziehungen übertragen, da  $N^{temp}$  Zyklen negativer Länge enthalten darf (Neumann, Schwindt und Zimmermann 2003, S. 8). Im Gegensatz dazu enthält  $N$  keine Zyklen, sofern die entsprechende Instanz des (TPSP) zulässig ist. Um eine Übertragbarkeit der in Abschnitt 2.3 und den nachfolgenden Kapiteln vorgestellten Lösungsverfahren auf Schedulingprobleme unter Zeitrestriktionen zu ermöglichen, wird die Menge der Vorgänger bei Vorhandensein von Zeitbeziehungen als

$$Pred^{temp}(i) := \{i' \in V \text{ mit } i' \neq i \mid d_{i'i} > 0 \vee (d_{i'i} = 0 \wedge d_{ii'} < 0)\} \quad (2.6)$$

und die Menge der Nachfolger als

$$Succ^{temp}(i) := \{i' \in V \text{ mit } i' \neq i \mid d_{ii'} > 0 \vee (d_{ii'} = 0 \wedge d_{i'i} < 0)\} \quad (2.7)$$

definiert. Dabei bezeichnet  $d_{ii'}$  die Länge des längsten Pfades von  $i$  zu  $i'$  im Netzplan  $N^{temp+} := (V, E^{temp} \cup \{(I+1, 0)\}, \delta)$  (Neumann, Schwindt und Zimmermann 2003, S. 11) mit

$$\delta_{I+1,0} := \begin{cases} \text{maximale Projektdauer} & \text{falls vorgegeben,} \\ \text{kritische Projektdauer} & \text{sonst.} \end{cases} \quad (2.8)$$

Der Netzplan  $N^{temp+}$  entspricht somit dem um die Kante  $(I+1, 0)$  ergänzten Netzplan  $N^{temp}$ . Die Gleichungen (2.6) und (2.7) sind an das Konzept der „Distanzordnung“ in Neumann, Schwindt und Zimmermann (2003, S. 17f) angelehnt und begründen sich damit, dass sie eine strenge Ordnung der Vorgänge beschreiben (zur Definition einer strengen Ordnung siehe Neumann, Schwindt und Zimmermann 2003, S. 16). Strenge Ordnungen werden für die in Abschnitt 2.3.1 beschriebenen Generierungsschemas benötigt. Intuitiv werden die Gleichungen (2.6) und (2.7) auch dadurch nachvollziehbar, dass das Vorhandensein eines zeitlichen Höchstabstands zwischen zwei Vorgängen  $i$  und  $i'$  alleine noch keine Aussage darüber ermöglicht, ob Vorgang  $i$  oder Vorgang  $i'$  zuerst startet. Deshalb impliziert dies auch keine Vorgänger-Nachfolger-Beziehung. Da der exemplarische Netzplan in Abbildung 2.1 bereits eine Kante  $(I + 1, 0)$  enthält, kann er als  $N^{temp+}$  betrachtet werden. Die Längen der längsten Pfade betragen beispielsweise  $d_{0,1} = 3$ ,  $d_{2,1} = -2$ ,  $d_{3,1} = -5$ ,  $d_{1,0} = -4$ ,  $d_{1,2} = 1$ ,  $d_{1,3} = 4$ . Das bedeutet,  $Pred^{temp}(1) = \{0\}$  und  $Succ^{temp}(1) = \{2, 3\}$ . Es sei darauf hingewiesen, dass eine Unterscheidung zwischen direkten und transitiven Vorgängern bzw. Nachfolgern bei Zeitbeziehungen obsolet ist. In den Gleichungen (2.6) und (2.7) sind bereits alle Vorgänge einbezogen, unabhängig davon, ob sie direkt über eine einzelne Anordnungsbeziehung oder transitiv über mehrere Anordnungsbeziehungen mit dem betrachteten Vorgang  $i$  verknüpft sind.

Die Metra-Potential-Methode ist zur Modellierung und Lösung des (TPSP/max) geeignet, falls die Projektdauer minimiert werden soll. Für eine ausführliche Erläuterung dazu wird auf Zimmermann u. a. (2006, Abschnitt 1.4.3) verwiesen. Die darin beschriebenen Algorithmen sind auch für das im vorangegangenen Abschnitt vorgestellte Problem (TPSP) geeignet, da Zeitbeziehungen eine Verallgemeinerung der Vorrangbeziehungen sind. Die Menge aller optimalen Lösungen kann in Polynomialzeit bestimmt werden, indem die frühestmöglichen und die spätestmöglichen Startzeitpunkte aller Vorgänge bestimmt werden. Die Zeitkomplexität des in Zimmermann u. a. (2006, S. 69) gezeigten Algorithmus ist laut Neumann, Schwindt und Zimmermann (2003, S. 17)  $\mathcal{O}(|V|^3)$ . Die Kritischer-Pfad-Methode ist aufgrund der in den verwendeten Vorgangskantennetzplänen nicht abbildbaren Zeitbeziehungen ungeeignet (Zimmermann u. a. 2006, S. 73). Zimmermann u. a. (2006, Kapitel 2) befassen sich ausführlich mit Methoden zum Scheduling unter Zeitrestriktionen für weitere Zielfunktionen neben der Projektdauerminimierung.

### 2.2.4 Scheduling unter (erneuerbaren) Ressourcenrestriktionen

In den bisher vorgestellten Schedulingproblemen unter Vorrang- bzw. Zeitrestriktionen wurde davon ausgegangen, dass zur Ausführung eines Projekts eine ausreichende Anzahl an *Ressourcen* zur Verfügung steht. In der Realität sind die Ressourcen knapp, was dazu führen kann, dass Vorgänge um Ressourcen konkurrieren und gegebenenfalls Zuteilungen notwendig werden. Das nachfolgend vorgestellte Modell ermöglicht es, diese Zuteilungen als modellendogene Entscheidung in das Schedulingproblem aufzunehmen und damit die Terminplanung um den Planungsbereich der Kapazitätsplanung zu erweitern. In der Schedulingliteratur werden verschiedene Arten von Ressourcen unterschieden, wobei sich dieser Abschnitt auf die sogenannten *erneuerbaren Ressourcen* beschränkt. Da es sich um

den sowohl in der Literatur als auch (vermutlich) in praktischen Anwendungen um den am meisten verbreiteten Ressourcentyp handelt, wird oft nicht explizit das Adjektiv „erneuerbar“ vorangestellt (z. B. Kolisch 1996b, S. 320). Aus diesem Grund ist es im Titel dieses Abschnitts in Klammern gesetzt. Erneuerbare Ressourcen sind dadurch definiert, dass sie von Vorgängen während ihres Ausführungszeitraums beansprucht werden können und nicht für andere Vorgänge zur Verfügung stehen. Nach Abschluss eines beanspruchenden Vorgangs werden alle durch ihn beanspruchten erneuerbaren Ressourcen freigegeben und stehen wieder in ursprünglicher Höhe für andere Vorgänge zur Verfügung (Zimmermann u. a. 2006, S. 51). Typische Beispiele sind Personalressourcen, Werkzeuge oder Maschinen.

Bei den erneuerbaren Ressourcen handelt es sich um einen Modellbestandteil, der wie folgt formalisiert werden kann: Zur Ausführung eines Projekts steht eine Menge an erneuerbaren Ressourcen  $\mathcal{R}^\alpha$  zur Verfügung. Erneuerbare Ressourcen müssen von Projektverantwortlichen zunächst mit einer geeigneten Einheit quantifiziert werden, z. B. Anzahl der Fachkräfte oder benötigte elektrische Leistung in Megawatt. Dann können die Projektverantwortlichen die *Ressourcennutzungen* der Vorgänge bestimmen, die für jeden Vorgang  $i \in V$  und jede Ressource  $k \in \mathcal{R}^\alpha$  als  $r_{ik}^\alpha$  angegeben werden. Jede Ressourcennutzung ist per Definition konstant über den gesamten Ausführungszeitraum eines Vorgangs  $i$  und beträgt somit  $r_{ik}^\alpha$  Einheiten zu jedem Zeitpunkt zwischen Start (inklusive) und Ende (exklusive) von  $i$ . Per Konvention wird von ganzzahligen Ressourceneinheiten ausgegangen, d. h.  $r_{ik}^\alpha \in \mathbb{Z}_{\geq 0}$ . Für die fiktiven Vorgänge gilt  $r_{0k}^\alpha = r_{I+1,k}^\alpha := 0$  für alle  $k \in \mathcal{R}^\alpha$ . Neben den Ressourcennutzungen können die Projektverantwortlichen für jede Ressource  $k \in \mathcal{R}^\alpha$  eine *Ressourcenkapazität*  $R_k^\alpha \in \mathbb{Z}_{\geq 0}$  festlegen. Falls eine Ressource unbeschränkt ist, wird  $R_k^\alpha := \infty$  gesetzt. Bei  $\mathcal{R}^\alpha$ ,  $r_{ik}^\alpha$  und  $R_k^\alpha$  handelt es sich um Problemparameter. Im Folgenden wird davon ausgegangen, dass Ressourcenkapazitäten  $R_k^\alpha$  konstant über den Zeithorizont sind. Zeitlich variable Ressourcenkapazitäten können modelliert werden, indem künstliche Vorgänge – sogenannte *Dummyvorgänge* – erzeugt und auf einen Zeitraum fixiert werden (Bartusch u. a. 1988; Zimmermann u. a. 2006, S. 198). Die Ressourcennutzung eines Dummyvorgangs wird so gewählt, dass er eine bestimmte Ressourcenkapazität blockiert und diese zur Ausführung der realen Vorgänge dadurch nicht mehr zur Verfügung steht. Es ist zu beachten, dass eine Fixierung von Vorgängen mittels Restriktionen nur mit den Zeitrestriktionen möglich ist.

Für einen gegebenen Schedule  $S$  umfasst die *aktive Menge* bezüglich der erneuerbaren Ressourcen

$$\mathcal{A}^\alpha(S, t) := \{i \in V \mid S_i \leq t < S_i + d_i\} \quad (2.9)$$

die Menge aller Vorgänge  $i \in V$ , die sich zu einem Zeitpunkt  $t$  in Ausführung befinden. Hierbei wird formal definiert, dass der Endzeitpunkt  $(S_i + d_i)$  eines Vorgangs  $i$  nicht mehr zum Ausführungszeitraum gezählt wird (vgl. Abschnitt 2.2.1) und Vorgang  $i$  zum Endzeitpunkt somit auch keine erneuerbaren Ressourcen mehr nutzt. Die Anzahl der durch alle Vorgänge zu einem Zeitpunkt  $t$  genutzten Einheiten einer Ressource  $k \in \mathcal{R}^\alpha$  ergibt sich bei gegebenem Schedule  $S$  zu

$$r_k^\alpha(S, t) := \sum_{i \in \mathcal{A}^\alpha(S, t)} r_{ik}^\alpha \quad (2.10)$$

(Zimmermann u. a. 2006, S. 118). Die Ressourcenkapazitäten implizieren (*erneuerbare*) *Ressourcenrestriktionen* für das Scheduling, die als Ungleichungen

$$r_k^\alpha(S, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0) \quad (2.11)$$

formuliert werden. Ein Schedule, der alle Ressourcenrestriktionen einhält, wird als *ressourcenzulässig* bezeichnet. Für einen gegebenen Schedule  $S$  und alle Zeitpunkte  $t \geq 0$  bezeichnet  $r_k^\alpha(S, \cdot)$  das *erneuerbare Ressourcenprofil* der erneuerbaren Ressource  $k$ . Das erneuerbare Ressourcenprofil ist eine rechtsseitig stetige Treppenfunktion (Neumann, Schwindt und Zimmermann 2003, S. 26).

Schedulingprobleme unter Ressourcenrestriktionen werden in der englischsprachigen Literatur als *Resource-Constrained Project Scheduling Problem* oder kurz als RCPSP bezeichnet. Es ist nicht klar definiert, welche Modellbestandteile das RCPSP umfasst. Stattdessen handelt es sich dabei um eine Klasse von Schedulingproblemen. Im weiteren Sinne kann unter einem RCPSP jedes Schedulingproblem verstanden werden, welches Restriktionen zur zeitlichen Abfolge der Vorgänge und zu beliebigen Ressourcentypen enthält (für eine Auflistung verschiedener Ressourcentypen siehe Węglarz u. a. 2011). Demnach gehören auch die in den nachfolgenden Abschnitten vorgestellten Schedulingprobleme in die RCPSP-Klasse. Im engeren Sinne handelt es sich bei einem RCPSP um ein Schedulingproblem, welches Vorrangrestriktionen (auch wenn in der Problembezeichnung nur von Ressourcen gesprochen wird) und erneuerbare Ressourcenrestriktionen umfasst, sowie als Zielsetzung die Minimierung der Projektdauer verfolgt (siehe z. B. Kolisch 2015). Brucker u. a. (1999) bezeichnet letzteres Schedulingproblem in ihrer Übersichtsarbeit als „Kernproblem“ der RCPSP-Klasse. Diese Arbeit orientiert sich an der enger ausgelegten Definition, schränkt jedoch die Zielsetzung formal zunächst nicht ein, sondern verwendet weiterhin die generische Zielfunktion  $z(S)$ . Das *Schedulingproblem unter Ressourcenrestriktionen (RCPSP)* enthält demnach die Vorrangrestriktionen (2.3) und die erneuerbaren Ressourcenrestriktionen (2.11). Es ist eine Generalisierung des Schedulingproblems unter Vorrangrestriktionen (TPSP). Die formale Beschreibung des (RCPSP) lautet:

$$\left. \begin{array}{ll} \min_S & z(S) \\ \text{u. d. R.} & S_{i'} \geq S_i + d_i \quad (\forall (i, i') \in E); \\ & r_k^\alpha(S, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0); \\ & S_0 = 0; \\ & S_i \geq 0 \quad (i = 1, \dots, I + 1). \end{array} \right\} \quad (\text{RCPSP})$$

Eine Instanz des (RCPSP) wird als  $\mathcal{I}_{\text{RCPSP}}$  bezeichnet. Bei der Formulierung des (RCPSP) handelt es sich um eine sogenannte *konzeptionelle Formulierung*, da  $r_k^\alpha(S, t)$  eine Funktion der Entscheidungsvariablen ist, deren Zuordnungsvorschrift nicht in die Formulierung integriert ist (Talbot und Patterson 1978, S. 1165). Konzeptionelle Formulierungen dienen der Problembeschreibung. Im Gegensatz dazu zielen beispielsweise *lineare Formulierungen* (auch *lineare Programme* genannt) auf die Lösung des Problems ab. Die erste lineare Formulierung des (RCPSP) wurde von Pritsker u. a. (1969) vorgestellt. Für alternative lineare Formulierungen wird auf Artigues u. a. (2015) verwiesen. Eine lineare Formulierung



zu der in dieser Arbeit betrachteten Problemstellung wird in Abschnitt 5.9 vorgestellt. Sie enthält auch die Vorrangrestriktionen und erneuerbaren Ressourcenrestriktionen des (RCPSp).

Beim Kernproblem der RCPSp-Klasse, d. h. dem (RCPSp) mit Minimierung der Projektdauer, handelt es sich um ein NP-schweres Problem (Blazewicz u. a. 1983). Für Probleme der Komplexitätsklasse NP (nichtdeterministische Polynomialzeit) gilt informell, dass sie nur mit exponentiellem Aufwand algorithmisch optimal gelöst werden können (Kripfganz und Perl 2020, S. 23). *NP-schwere Probleme* werden definiert als solche Probleme, auf die alle Probleme in NP in Polynomialzeit reduziert werden können. Sie sind daher mindestens so schwer lösbar wie Probleme in NP. So existieren für das (RCPSp) Instanzen mit nur 30 Vorgängen, zu denen bisher keine optimale Lösung gefunden werden konnte (Coelho und Vanhoucke 2018). Eine zulässige Lösung kann für das (RCPSp) allerdings in Polynomialzeit bestimmt werden, indem Vorgänge nacheinander zu vorrang- und ressourcenzulässigen Zeitpunkten eingeplant werden (Kolisch 2015, S. 10f). Dazu geeignete Algorithmen werden in Abschnitt 2.3.1 behandelt. Es gibt auch zahlreiche Instanzen des (RCPSp), die relativ schnell und einfach optimal gelöst werden können (Coelho und Vanhoucke 2020, S. 2). Die Einstufung des Problems als NP-schwer beschreibt daher nur das im ungünstigsten Fall zu erwartende Laufzeitverhalten. In vielen praktischen Fällen können jedoch gute bis optimale Lösungen gefunden werden.

Durch Ersetzen der Vorrangrestriktionen (2.3) durch Zeitrestriktionen (2.5) kann das (RCPSp) zum (RCPSp/max) generalisiert werden:

$$\left. \begin{array}{ll} \min_S & z(S) \\ \text{u. d. R.} & S_{i'} \geq S_i + \delta_{ii'} \quad (\forall (i, i') \in E^{temp}); \\ & r_k^\alpha(S, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0); \\ & S_0 = 0; \\ & S_i \geq 0 \quad (i = 1, \dots, I + 1). \end{array} \right\} \quad (\text{RCPSp/max})$$

Eine Instanz des (RCPSp/max) wird als  $\mathcal{I}_{\text{RCPSp/max}}$  bezeichnet. Grundlegende Studien zum (RCPSp/max) wurden von Bartusch u. a. (1988) und Elmaghraby und Kamburowski (1992) vorgestellt. Letztere sprechen von „generalisierten Vorrangbeziehungen“, weshalb das (RCPSp/max) auch als RCPSp mit generalisierten Vorrangbeziehungen bezeichnet wird. Für das (RCPSp/max) ist bereits das Entscheidungsproblem zur Existenz einer zulässigen Lösung NP-vollständig (Bartusch u. a. 1988; Brucker u. a. 1999, S. 7), d. h., die Berechnung einer zulässigen Lösung verursacht exponentiellen Rechenaufwand. *NP-vollständige Probleme* sind NP-schwere Probleme, für die zusätzlich bekannt ist, dass sie auch in NP liegen (Kripfganz und Perl 2020, S. 23). Das Optimierungsproblem (RCPSp/max) ist bei Minimierung der Projektdauer NP-schwer, da es sich um eine Generalisierung des (RCPSp) handelt.

Eine ausführliche Darlegung verschiedener exakter oder heuristischer Lösungsverfahren für Probleme der RCPSp-Klasse ist in Neumann, Schwindt und Zimmermann (2003) enthalten. Da es sich bei diesen Schedulingproblemen um ein sehr intensiv erforschtes

Themengebiet handelt, existiert in der Literatur eine Vielzahl von Lösungsverfahren und Erweiterungen. Die Arbeiten von Brucker u. a. (1999) und Habibi u. a. (2018) vermitteln einen Überblick über relevante Publikationen. Pellerin u. a. (2020) stellen in ihrer Übersichtsarbeit verschiedene heuristische Lösungsansätze vor. Ein konkretes Lösungsverfahren, welches sich aus anderen leistungsstarken Verfahren in der Literatur zusammensetzt, ist beispielsweise in Coelho und Vanhoucke (2018) zu finden.

### 2.2.5 Mehr-Modus-Scheduling

Den bisher vorgestellten Schedulingproblemen liegt die Annahme zugrunde, dass Projektverantwortliche für alle Vorgänge im Voraus die Dauern sowie die Ressourcennutzungen festlegen. In der Praxis ist es möglich, dass für einen Vorgang mehrere *Ausführungsalternativen* existieren, die mit unterschiedlichen Dauern und Ressourcennutzungen einhergehen. Beispielsweise, wenn bestimmte Vorgänge entweder mit internem oder externem Personal ausgeführt werden können. Falls derartige Ausführungsalternativen existieren, mussten Projektverantwortliche bisher modellexogen eine Entscheidung treffen, welche Ausführungsalternative als Eingabe für das Schedulingproblem herangezogen wird. Im *Mehr-Modus-Scheduling* wird das Modell deshalb dahingehend erweitert, die Entscheidung zu Ausführungsalternativen modellendogen in das Schedulingproblem einzubinden. Bei einem *Modus* handelt es sich dabei um eine Ausführungsalternative eines Vorgangs, zu der die Dauer und die Ressourcennutzungen gegeben sind. Formal wird für jeden Vorgang  $i \in V$  eine *Modusmenge*  $\mathcal{M}_i$  definiert. Wenn Vorgang  $i$  in einem Modus  $m \in \mathcal{M}_i$  ausgeführt wird, dauert er  $d_{im} \in \mathbb{R}_{\geq 0}$  Perioden und nutzt während seiner Ausführung  $r_{imk}^\alpha \in \mathbb{Z}_{\geq 0}$  Einheiten jeder erneuerbaren Ressource  $k \in \mathcal{R}^\alpha$ . Im Vergleich zum sogenannten *Ein-Modus-Fall*, der in den vorangegangenen Abschnitten betrachtet wurde, sind die Dauern und die Ressourcennutzungen im *Mehr-Modus-Fall* auf die Dimension der Modusmenge erweitert, wie am zusätzlichen Index  $m$  erkennbar. Ohne Beschränkung der Allgemeinheit kann  $|\mathcal{M}_i| \geq 1$  für jeden Vorgang  $i \in V$  gefordert werden. Wenn für einen Vorgang  $i$  keine Ausführungsalternativen existieren, wird ein einzelner Modus  $m_0$  eingeführt, für dessen Dauer  $d_{im_0} := d_i$  und Ressourcennutzungen  $r_{im_0k}^\alpha := r_{ik}^\alpha$  für alle  $k \in \mathcal{R}^\alpha$  gilt. Für fiktive Vorgänge gilt  $|\mathcal{M}_0| = |\mathcal{M}_{I+1}| := 1$ . Die Modusmengen sind Problemparameter.

Neben den Entscheidungen über die Startzeitpunkte  $S := (S_0, \dots, S_{I+1})$  der Vorgänge sind im *Mehr-Modus-Schedulingproblem* auch Entscheidungen zur Moduswahl zu treffen. Dazu wird für jeden Vorgang  $i \in V$  die *Moduszuweisung*  $M_i$  als neue Entscheidungsvariable definiert. Für sie muss

$$M_i \in \mathcal{M}_i \tag{2.12}$$

gelten. Ein Vektor  $M := (M_0, \dots, M_{I+1})$  mit Entscheidungen zu allen Moduszuweisungen wird ebenfalls als Moduszuweisung bezeichnet. Somit ist eine Lösung des Mehr-Modus-Schedulingproblems ein Tupel  $(S, M)$  aus Schedule  $S$  und Moduszuweisung  $M$ .

Das Konzept der Modi motiviert den zusätzlichen Ressourcentyp der *nichtererneuerbaren Ressourcen*. Nichterneuerbare Ressource werden auch Verbrauchsressourcen genannt.

Vorgänge beanspruchen nichterneuerbare Ressourcen per Definition zum Start ihres Ausführungszeitraums und geben sie danach nicht mehr frei. Im Gegensatz zu erneuerbaren Ressourcen stehen nichterneuerbare Ressourcen somit für andere Vorgänge nicht mehr zu Verfügung. Typische Beispiele sind Finanzmittel oder Rohstoffe. Die Menge der nichterneuerbaren Ressourcen wird als  $\mathcal{R}^\beta$  bezeichnet. Von jeder nichterneuerbaren Ressource  $k \in \mathcal{R}^\beta$  steht eine Ressourcenkapazität  $R_k^\beta \in \mathbb{Z}_{\geq 0}$  zur Verfügung. Die Ressourcennutzung einer nichterneuerbaren Ressource  $k \in \mathcal{R}^\beta$  durch einen Vorgang  $i \in V$  bei Ausführung in Modus  $m \in \mathcal{M}_i$  beträgt  $r_{imk}^\beta \in \mathbb{Z}_{\geq 0}$  Einheiten. Bei  $\mathcal{R}^\beta$ ,  $r_{imk}^\beta$  und  $R_k^\beta$  handelt es sich um Problemparameter. Es wird analog zu den erneuerbaren Ressourcen davon ausgegangen, dass die nichterneuerbaren Ressourcenkapazitäten  $R_k^\beta$  konstant über den Zeithorizont sind. Variable zeitliche Ressourcenkapazitäten sind bei nichterneuerbaren Ressourcen nur mit der Bedingung sinnvoll, dass sie über die Zeit nicht abfallen dürfen. Eine zeitlich abfallende Ressourcenkapazität würde bedeuten, dass zu einem späteren Zeitpunkt weniger Einheiten zur Verfügung stehen. Diese Einheiten könnten allerdings auch nicht zu früheren Zeitpunkten genutzt werden, da sie entsprechend der modelltheoretischen Natur der nichterneuerbaren Ressourcen von Vorgängen bis zum Ende des Zeithorizonts beansprucht werden. Eine Abbildung variabler Ressourcenkapazitäten mit Dummyvorgängen ist deshalb auch nicht möglich.

Für einen gegebenen Schedule  $S$  umfasst die *aktive Menge* bezüglich der nichterneuerbaren Ressourcen

$$\mathcal{A}^\beta(S, t) := \{i \in V \mid 0 \leq S_i \leq t\} \quad (2.13)$$

die Menge aller Vorgänge  $i \in V$ , deren Ausführung bis einschließlich Zeitpunkt  $t$  gestartet wurde. Die Anzahl der durch alle Vorgänge zu einem Zeitpunkt  $t$  genutzten Einheiten einer nichterneuerbaren Ressource  $k \in \mathcal{R}^\beta$  ergibt sich bei gegebenem Schedule  $S$  und gegebener Moduszuweisung  $M$  zu

$$r_k^\beta(S, M, t) := \sum_{i \in \mathcal{A}^\beta(S, t)} r_{iM_i k}^\beta. \quad (2.14)$$

Die Funktion  $r_k^\beta(S, M, t)$  ist monoton steigend in  $t$ , da die Menge  $\mathcal{A}^\beta(S, t)$  für jedes  $t$  stets alle Vorgänge enthält, die sie auch für ein  $t' < t$  enthält. Intuitiv lässt sich das auch damit erklären, dass nichterneuerbare Ressourcen nicht mehr freigegeben werden und die Nutzung zum Projektende deshalb ihr Maximum annimmt. Es ist somit ausreichend, lediglich für das Projektende  $S_{I+1}$  die Einhaltung der nichterneuerbaren Ressourcenkapazität sicherzustellen. Da  $\mathcal{A}^\beta(S, S_{I+1}) \equiv V$  gilt (zum Projektende wurden alle Vorgänge gestartet), werden die *nichterneuerbaren Ressourcenrestriktionen* als Ungleichungen

$$\sum_{i \in V} r_{iM_i k}^\beta \leq R_k^\beta \quad (\forall k \in \mathcal{R}^\beta) \quad (2.15)$$

formuliert. Für eine gegebene Lösung  $(S, M)$  und alle Zeitpunkte  $t \geq 0$  bezeichnet  $r_k^\beta(S, M, \cdot)$  das *nichterneuerbare Ressourcenprofil* der nichterneuerbaren Ressource  $k$ . Das nichterneuerbare Ressourcenprofil ist eine rechtsseitig stetige und monoton steigende Treppenfunktion.

Das Schedulingproblem unter erneuerbaren Ressourcenrestriktionen (RCPS) kann nun als *Mehr-Modus-Schedulingproblem (MRCPS)* generalisiert werden, indem Bedingung (2.12) und Ungleichungen (2.15) als Restriktionen hinzugefügt werden. Außerdem müssen die in Abschnitt 2.2.4 definierten formalen Zusammenhänge an den Mehr-Modus-Fall angepasst werden; d. h. für die aktive Menge bezüglich der erneuerbaren Ressourcen

$$\mathcal{A}^\alpha(S, M, t) := \{i \in V \mid S_i \leq t < S_i + d_{iM_i}\}, \quad (2.16)$$

für die Anzahl der zum Zeitpunkt  $t$  genutzten Einheiten einer erneuerbaren Ressource

$$r_k^\alpha(S, M, t) := \sum_{i \in \mathcal{A}^\alpha(S, M, t)} r_{iM_i k}^\alpha \quad (2.17)$$

und für die erneuerbaren Ressourcenrestriktionen

$$r_k^\alpha(S, M, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0). \quad (2.18)$$

Die formale Beschreibung des (MRCPS) lautet zusammengefasst:

$$\left. \begin{array}{ll} \min_{S, M} & z(S, M) \\ \text{u. d. R.} & S_{i'} \geq S_i + d_{iM_i} \quad (\forall (i, i') \in E); \\ & r_k^\alpha(S, M, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0); \\ & \sum_{i \in V} r_{iM_i k}^\beta \leq R_k^\beta \quad (\forall k \in \mathcal{R}^\beta); \\ & S_0 = 0; \\ & S_i \geq 0 \quad (i = 1, \dots, I + 1); \\ & M_i \in \mathcal{M}_i \quad (\forall i \in V). \end{array} \right\} \quad (\text{MRCPS})$$

Das Mehr-Modus-Schedulingproblem wurde erstmalig von Elmaghraby (1977) betrachtet. Beim (MRCPS) handelt es sich um eine konzeptionelle Formulierung, die an van Peteghem und Vanhoucke (2014) angelehnt ist. Die erste lineare Formulierung des Mehr-Modus-Schedulingproblems wurde von Talbot (1982) vorgestellt. Durch Ersetzen der Vorrangrestriktionen durch Zeitrestriktionen (2.5) kann das (MRCPS) auf das Modell der Zeitbeziehungen generalisiert werden, welches erstmalig von Reyck und Herroelen (1999) behandelt wurde.

Die nichterneuerbaren Ressourcenrestriktionen (2.15) machen deutlich, weshalb nicht-erneuerbare Ressourcen nur im Mehr-Modus-Fall ein sinnvoller Modellbestandteil sind (Neumann, Schwindt und Zimmermann 2003, S. 160). Im Ein-Modus-Fall würden sich die Ungleichungen (2.15) zu  $\sum_{i \in V} r_{ik}^\beta \leq R_k^\beta$  reduzieren und wären damit entscheidungsunabhängig. Es könnte dann bereits im Voraus geprüft werden, ob die nichterneuerbaren Ressourcen zur Durchführung des Projekts ausreichen. Alternativen stünden innerhalb des Untersuchungsrahmens des Schedulingproblems ohnehin nicht zur Verfügung.

Als Generalisierung des (RCPS) handelt es sich beim (MRCPS) ebenfalls um ein NP-schweres Problem, falls die Projektdauer minimiert werden soll. Grundsätzlich kann das

(MRCPSP) in ein Moduszuweisungsproblem und ein (RCPSP) aufgeteilt werden (Neumann, Schwindt und Zimmermann 2003, S. 161). Da das Entscheidungsproblem zur Existenz einer zulässigen Moduszuweisung NP-vollständig ist (Neumann, Schwindt und Zimmermann 2003, S. 165), ist für Instanzen mit mehr als einer nichterneuerbaren Ressource auch das Entscheidungsproblem zur Existenz einer zulässigen Lösung des (MRCPSP) NP-vollständig (Kolisch 1995; zitiert nach Węglarz u. a. 2011, S. 62). Die Lösung von Mehr-Modus-Schedulingproblemen wurde intensiv erforscht, woraus zahlreiche exakte (z. B. Talbot 1982; Patterson u. a. 1989; Demeulemeester und Herroelen 1992; Sprecher, Hartmann u. a. 1997) und heuristische Lösungsverfahren (z. B. Boctor 1993; Mika u. a. 2008; Coelho und Vanhoucke 2011; Geiger 2017) hervorgegangen sind. Schnell und Hartl (2016) stellen ein Lösungsverfahren für das Mehr-Modus-Schedulingproblem mit Zeitbeziehungen vor. Für eine Übersichtsarbeit zu Mehr-Modus-Schedulingproblemen wird auf Węglarz u. a. (2011) verwiesen, wo auch zahlreiche Erweiterungen des Problems aufgelistet sind. Auf einige Erweiterungen wird auch in Abschnitt 2.2.7 eingegangen. Für einen experimentellen Vergleich verschiedener Lösungsverfahren wird auf van Peteghem und Vanhoucke (2014) verwiesen.

### 2.2.6 Scheduling unter erneuerbaren und kumulativen Ressourcenrestriktionen

In diesem Abschnitt wird vom Ein-Modus-Fall ausgegangen. Eine Generalisierung auf den Mehr-Modus-Fall ist mit den im vorangegangenen Abschnitt 2.2.5 vorgestellten Konzepten möglich.

Bei *kumulativen Ressourcen* handelt es sich um einen weiteren Ressourcentyp, der von Schwindt (1999) und Neumann und Schwindt (2002) erstmalig vorgestellt wurde. Die kumulativen Ressourcen sind unter anderem dazu geeignet, Lagerflächen bzw. -raum im Rahmen des Scheduling zu modellieren. Während bei erneuerbaren und nichterneuerbaren Ressourcen per Definition vorgegeben ist, zu welchem Zeitpunkt während der Ausführung eines Vorgangs die Ressourcenbeanspruchung und, im Fall von erneuerbaren Ressourcen, die Freigabe erfolgt, ist dies bei kumulativen Ressourcen flexibel. Somit ist es bei einer kumulativen Ressource beispielsweise möglich, dass sie durch einen Vorgang  $i$  beansprucht und durch einen anderen Vorgang  $i'$  wieder freigegeben wird. Auch in ihrer Höhe sind die Ressourcenbeanspruchungen und -freigaben bei kumulativen Ressourcen flexibel. So kann ein Vorgang  $i$  eine kumulative Ressource beanspruchen, die von zwei anderen Vorgängen  $i'$  und  $i''$  jeweils hälftig wieder freigegeben wird.

Die Menge der kumulativen Ressourcen wird als  $\mathcal{R}^Y$  bezeichnet. Motiviert durch die Anwendung in der Lagerhaltung ist für eine kumulative Ressource  $k \in \mathcal{R}^Y$  nicht nur die Ressourcenkapazität  $\bar{R}_k^Y \in \mathbb{Z}$  als *Höchstbestand* relevant, sondern auch eine Untergrenze  $\underline{R}_k^Y \in \mathbb{Z}$  als *Mindestbestand*. Die Beanspruchung und Freigabe einer kumulativen Ressource  $k \in \mathcal{R}^Y$  durch einen Vorgang  $i \in V$  wird als Ressourcennutzung  $r_{ik}^Y$  bezeichnet, wobei eine positive Ressourcennutzung einer Beanspruchung entspricht und eine negative Ressourcennutzung einer Freigabe. Im Kontext der Lagerhaltung entspricht die Beanspru-

chung einer kumulativen Ressource einer *Einlagerung* (Lagerplatz wird beansprucht). Die Freigabe einer kumulativen Ressource entspricht einer *Auslagerung* (Lagerplatz wird freigegeben). Obwohl im weiteren Verlauf dieser Arbeit kumulative Ressourcen ausschließlich zur Modellierung von Lagern verwendet werden, sind sie zur Abbildung vieler weiterer praktischer Problemstellungen, wie beispielsweise knapper Energiereserven oder knapper finanzieller Mittel, geeignet. Kyriakidis u. a. (2012) zeigen, dass mit kumulativen Ressourcen auch verschiedene logische Verknüpfungen zwischen Vorgängen ausgedrückt werden können. Beispielsweise kann modelliert werden, dass mindestens zwei Vorgänge aus einer Teilmenge  $\tilde{V} \subset V$  ausgeführt werden müssen, bevor ein anderer Vorgang  $i \notin \tilde{V}$  starten kann.

Um die flexible zeitliche Verteilung der kumulativen Ressourcennutzungen zu formalisieren, kann  $r_{ik}^Y$  als Funktion  $[0, d_i] \rightarrow \mathbb{Z}$  definiert werden, die jedem Zeitpunkt während der Ausführung des Vorgangs  $i$  seine skalare Ressourcennutzung zuordnet. Es wird ersichtlich, dass die kumulativen Ressourcennutzungen im Gegensatz zu den erneuerbaren und nichterneuerbaren Ressourcennutzungen um eine zeitliche Dimension erweitert sind. Die Funktionsnotation für  $r_{ik}^Y$  führt zur größtmöglichen Generalisierung des Konzepts der kumulativen Ressourcen, resultiert jedoch in einer aufwendigen Modellierung und Problemformulierung. Daher werden in der Literatur üblicherweise einschränkende Annahmen zur zeitlichen Verteilung der kumulativen Ressourcennutzungen getroffen: Schwindt und Trautmann (2000, S. 513) und Neumann, Schwindt und Zimmermann (2003, S. 129) definieren, dass einlagernde Vorgänge kumulative Ressourcen stets zu ihrem Ende und auslagernde Vorgänge kumulative Ressourcen stets zu ihrem Start nutzen. Bartels u. a. (2011, S. 412) gehen vom umgekehrten Fall aus: Einlagernde Vorgänge nutzen kumulative Ressourcen stets zu ihrem Start und auslagernde Vorgänge nutzen kumulative Ressourcen stets zu ihrem Ende. Sourd und Rogerie (2005, S. 511), Neumann, Schwindt und Trautmann (2005, S. 499) und Neumann, Schwindt und Zimmermann (2003, S. 138ff) betrachten eine Problemstellung, in welcher alle kumulative Ressourcen gleichmäßig während der Vorgangsausführung genutzt werden. Diese Modellierung lässt sich beispielsweise mit der chemischen Industrie motivieren, wo Flüssigkeiten von Vorgängen gleichmäßig in Tanks ein- bzw. aus Tanks ausgeleitet werden (Sourd und Rogerie 2005). In beiden Beiträgen wird von kumulativen Ressourcen gesprochen, in die mit „konstanten Raten ein- und ausgelagert wird“.

In dieser Arbeit sollen kumulative Ressourcen möglichst allgemein gehalten werden. Für jeden Vorgang  $i \in V$  und jede kumulative Ressource  $k \in \mathcal{R}^Y$  soll individuell festgelegt werden können, ob  $i$  die Ressource  $k$  zu seinem Start, zu seinem Ende oder gleichmäßig während seiner Ausführung nutzt. Dazu wird die kumulative Ressourcennutzung  $r_{ik}^Y$  durch den dreidimensionalen Vektor  $(r_{ik}^{Y,start}, r_{ik}^{Y,lin}, r_{ik}^{Y,end})$  ersetzt. Dabei bezeichnet  $r_{ik}^{Y,start} \in \mathbb{Z}$  die Ressourcennutzung zum Start von  $i$ ,  $r_{ik}^{Y,lin} \in \mathbb{Z}$  die gleichmäßige Ressourcennutzung während der Ausführung von  $i$  und  $r_{ik}^{Y,end} \in \mathbb{Z}$  die Ressourcennutzung zum Ende von  $i$ . Zusammengefasst wird hierbei von drei *Nutzungsarten* gesprochen. Die Bezeichnung „lin“ in der Notation für die gleichmäßige Nutzung während der Vorgangsausführung ist eine Abkürzung für „linear“. Sie wurde vom linearen Verlauf des kumulativen Ressourcenprofils

abgeleitet, der aus dieser Nutzungsart folgt (siehe unten). Bei  $r_{ik}^{y,start}$ ,  $r_{ik}^{y,lin}$  und  $r_{ik}^{y,end}$  handelt es sich um Problemparameter. Es ist ohne Einschränkung möglich, dass sich die Vorzeichen der Nutzungsarten unterscheiden. Beispielsweise kann ein Vorgang zu seinem Start in eine kumulative Ressource  $k$  einlagern und zu seinem Ende wieder aus derselben kumulativen Ressource  $k$  auslagern.

Die Anzahl der durch einen Vorgang  $i \in V$  mit gegebenem Startzeitpunkt  $S_i$  zu einem Zeitpunkt  $t \geq S_i$  genutzten Einheiten einer kumulativen Ressource  $k \in \mathcal{R}^Y$  ergibt sich zu

$$r_{ik}^y(S_i, t) := \begin{cases} r_{ik}^{y,start} & \text{falls } t = S_i, \\ r_{ik}^{y,start} + (t - S_i) r_{ik}^{y,lin} / d_i & \text{falls } S_i < t < S_i + d_i, \\ r_{ik}^{y,start} + r_{ik}^{y,lin} + r_{jk}^{y,end} & \text{falls } t \geq S_i + d_i. \end{cases} \quad (2.19)$$

Falls  $t = S_i$ , ist die Nutzung zum Vorgangsstart erfolgt. Falls  $S_i < t < S_i + d_i$ , sind die Nutzung zum Vorgangsstart und die anteilige Nutzung während der Vorgangsausführung erfolgt. Letztere ergibt sich aus der Multiplikation der seit Vorgangsstart vergangenen Zeit  $(t - S_i)$  und der Ressourcennutzung pro Periode  $r_{ik}^{y,lin} / d_i$  (vgl. Neumann, Schwindt und Trautmann 2005, S. 499). Falls  $t \geq S_i + d_i$ , sind alle drei Nutzungsarten erfolgt. Die gesamte Anzahl der durch Vorgang  $i$  genutzten Einheiten einer kumulativen Ressource  $k$  wird als  $r_{ik}^y(\cdot, \infty)$  bezeichnet. Da dieser Wert unabhängig vom Startzeitpunkt  $S_i$  ist, wird in der Notation ein  $\cdot$  verwendet. Es gilt  $r_{ik}^y(\cdot, \infty) \equiv r_{ik}^{y,start} + r_{ik}^{y,lin} + r_{jk}^{y,end}$ . Ein Vorgang  $i$  wird bezüglich einer kumulativen Ressource  $k$  als einlagernd bezeichnet, falls  $r_{ik}^y(\cdot, \infty) > 0$ , d. h., falls Vorgang  $i$  Ressource  $k$  auch nach seinem Ende positiv nutzt. Umgekehrt wird er als auslagernd bezeichnet, falls  $r_{ik}^y(\cdot, \infty) < 0$ , d. h., falls Vorgang  $i$  Ressource  $k$  auch nach seinem Ende negativ nutzt.

Für einen gegebenen Schedule  $S$  umfasst die *aktive Menge* bezüglich der kumulativen Ressourcen

$$\mathcal{A}^Y(S, t) := \{i \in V \mid 0 \leq S_i \leq t\} \quad (2.20)$$

die Menge aller Vorgänge  $i \in V$ , deren Ausführung bis einschließlich Zeitpunkt  $t$  gestartet wurde. Die Anzahl der durch alle Vorgänge zu einem Zeitpunkt  $t$  genutzten Einheiten einer kumulativen Ressource  $k \in \mathcal{R}^Y$  ergibt sich bei gegebenem Schedule  $S$  zu

$$r_k^y(S, t) := \sum_{i \in \mathcal{A}^Y(S, t)} r_{ik}^y(S_i, t). \quad (2.21)$$

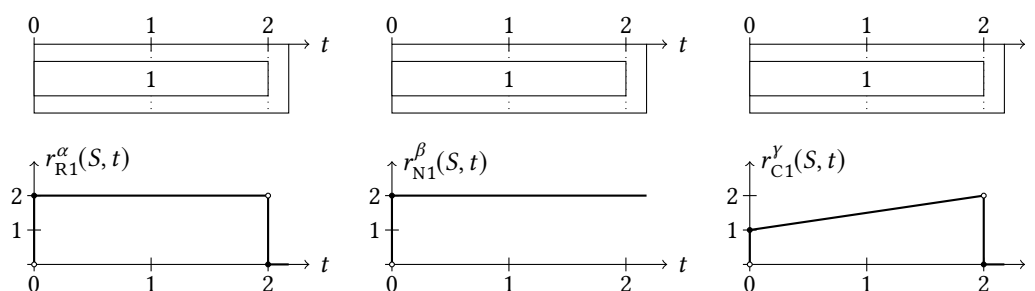
Die Höchst- und Mindestbestände implizieren *kumulative Ressourcenrestriktionen* für das Scheduling, die als Ungleichungen

$$\underline{R}_k^y \leq r_k^y(S, t) \leq \overline{R}_k^y \quad (\forall k \in \mathcal{R}^Y, t \geq 0) \quad (2.22)$$

formuliert werden (vgl. Neumann, Schwindt und Zimmermann 2003, S. 129). Für einen gegebenen Schedule  $S$  und alle Zeitpunkte  $t \geq 0$  bezeichnet  $r_k^y(S, \cdot)$  das *kumulative Ressourcenprofil* der kumulativen Ressource  $k$ .

Unter Verwendung der eingeführten Nutzungsarten lassen sich kumulative Ressourcen auch als Generalisierung von erneuerbaren und nichterneuerbaren Ressourcen betrachten: Sei  $k \in \mathcal{R}^\alpha$  eine erneuerbare Ressource, die von Vorgang  $i$  mit  $r_{ik}^\alpha$  Einheiten genutzt wird. Dann kann  $k$  auch als kumulative Ressource  $k \in \mathcal{R}^Y$  modelliert werden mit  $r_{ik}^{Y,start} := r_{ik}^\alpha$  und  $r_{ik}^{Y,end} := -r_{ik}^\alpha$ . Sei  $k \in \mathcal{R}^\beta$  eine nichterneuerbare Ressource, die von Vorgang  $i$  mit  $r_{ik}^\beta$  Einheiten genutzt wird. Dann kann  $k$  auch als kumulative Ressource  $k \in \mathcal{R}^Y$  modelliert werden mit  $r_{ik}^{Y,start} := r_{ik}^\beta$ . Beispiel 2.1 veranschaulicht exemplarische Ressourcenprofile für die drei verschiedenen Ressourcentypen.

**Beispiel 2.1:** Ein Vorgang 1 mit Dauer  $d_1 = 2$  nutzt eine erneuerbare Ressource R1, eine nichterneuerbare Ressource N1 und eine kumulative Ressource C1. Dabei gilt  $r_{1,R1}^\alpha = r_{1,N1}^\beta = 2$ ,  $r_{1,C1}^{Y,start} := 1$ ,  $r_{1,C1}^{Y,lin} := 1$  und  $r_{1,C1}^{Y,end} := -2$ . Abbildung 2.2 zeigt die Ressourcenprofile von R1, N1 und C1 für  $S_1 = 0$ . Auf die Angabe eines Modus wird bei den nichterneuerbaren Ressourcen verzichtet.  $\square$



**Abbildung 2.2:** Gantt-Diagramme und Ressourcenprofile zu Beispiel 2.1.

Obwohl demnach die kumulativen Ressourcen als einzige Ressource im Modell ausreichen würden, um auch die erneuerbaren und nichterneuerbaren Ressourcen abzubilden, werden die drei Ressourcentypen in dieser Arbeit dennoch weiterhin getrennt behandelt. Dies folgt erstens aus der Historie der Schedulingprobleme unter Ressourcenrestriktionen, die traditionell auf dem erneuerbaren Ressourcentyp basieren. Deshalb entfällt auf diesen Typ auch ein Großteil der Literatur. Zweitens ist es im Hinblick auf das Lösungsverfahren sinnvoll, praktische Sachverhalte mit differenzierten Modellierungskonzepten abzubilden, um die jeweils am besten geeigneten Algorithmen verwenden zu können. Und drittens ist auch aus anwendungsorientierter Perspektive eine Trennung intuitiver. Projektverantwortliche können damit bei der Zusammenstellung einer Probleminstanz auf passgenaue Ressourcentypen zugreifen.

Es sei angemerkt, dass das Konzept der kumulativen Ressourcen auch beliebige weitere Nutzungsarten erlauben würde. Beispielsweise könnte auch eine Nutzungsart eingeführt werden, die die kumulative Ressourcennutzung nach der halben Ausführungszeit eines Vorgangs beschreibt. Dazu müsste lediglich Gleichung (2.19) angepasst werden. Da es sich bei der Nutzung zum Vorgangsstart, der gleichmäßigen Nutzung während der Vorgangsausführung sowie der Nutzung zum Vorgangsende sowohl um die praktisch plausiblen als



auch um die in der Literatur behandelten Nutzungsarten handelt, werden diese jedoch als ausreichend betrachtet.

Das (RCPSP) kann nun als *Schedulingproblem unter erneuerbaren und kumulativen Ressourcenrestriktionen (RCPSP/c)* generalisiert werden, indem Ungleichungen (2.22) als Restriktionen hinzugefügt werden. Die formale Beschreibung des (RCPSP/c) lautet:

$$\begin{array}{ll}
 \min_S & z(S) \\
 \text{u. d. R.} & \left. \begin{array}{ll}
 S_{i'} \geq S_i + d_i & (\forall (i, i') \in E); \\
 r_k^\alpha(S, t) \leq R_k^\alpha & (\forall k \in \mathcal{R}^\alpha, t \geq 0); \\
 \underline{R}_k^\gamma \leq r_k^\gamma(S, t) \leq \overline{R}_k^\gamma & (\forall k \in \mathcal{R}^\gamma, t \geq 0); \\
 S_0 = 0; \\
 S_i \geq 0 & (i = 1, \dots, I + 1).
 \end{array} \right\} \quad (\text{RCPSP/c})
 \end{array}$$

Eine Instanz des (RCPSP/c) wird als  $\mathcal{I}_{\text{RCPSP/c}}$  bezeichnet. Die nichterneuerbaren Ressourcen sind aus den in Abschnitt 2.2.5 erläuterten Gründen im hier betrachteten Ein-Modus-Fall nicht enthalten. Durch Ersetzen der Vorrangrestriktionen durch Zeitrestriktionen (2.5) kann das (RCPSP/c) auf das Modell der Zeitbeziehungen erweitert werden. Beim (RCPSP/c) handelt es sich um eine konzeptionelle Formulierung.

Kumulative Ressourcen werden auch als Lagerressourcen (engl. storage resources) (z. B. Carlier und Moukrim 2015), als Reservoirs (z. B. Laborie 2003) oder als Ressourcen, die konsumiert und produziert werden (z. B. Carlier, Moukrim und Xu 2009), bezeichnet. Es sei außerdem darauf hingewiesen, dass in manchen Publikationen erneuerbare Ressourcen mit einer Kapazität  $R_k^\alpha > 1$  als kumulative Ressourcen bezeichnet werden (z. B. Baptiste u. a. 1999). Die genannten alternativen Bezeichnungen bzw. Bedeutungen des Begriffs der kumulativen Ressource werden hier nicht weiter verwendet.

Als Generalisierung des (RCPSP) handelt es sich beim (RCPSP/c) mit dem Ziel der Projektdauerminimierung ebenfalls um ein NP-schweres Problem. Das Entscheidungsproblem zur Existenz einer zulässigen Lösung des (RCPSP/c) ist NP-vollständig (Neumann, Schwindt und Zimmermann 2003, S. 130). Lösungsverfahren bei Nutzungen zum Vorgangsstart und -ende sind in Schwindt und Trautmann (2000), Neumann und Schwindt (2002), Neumann, Schwindt und Zimmermann (2003, S. 129ff), Laborie (2003), Schwindt, Fink u. a. (2007) und Carlier, Moukrim und Xu (2009) zu finden. Lösungsverfahren bei Nutzungen während der Vorgangsausführung werden in Neumann, Schwindt und Zimmermann (2003, S. 138ff), Neumann, Schwindt und Trautmann (2005) und Sourd und Rogerie (2005) behandelt. Wie ein Vergleich von Neumann, Schwindt und Trautmann (2005, S. 507) zeigt, hängt der Rechenaufwand zur Lösung des (RCPSP/c) von den verwendeten Nutzungsarten ab. Das Problem kann demnach mit deutlich geringerem Aufwand gelöst werden, wenn nur Nutzungen zum Vorgangsstart und -ende erlaubt sind.

Da kumulative Ressourcen zur Modellierung von Stoffströmen geeignet sind, werden sie als zentraler Modellbestandteil für das im Rahmen dieser Arbeit entwickelte Optimierungswerkzeug herangezogen. Deshalb wird für eine detaillierte Literaturrecherche zu Lösungsverfahren für das (RCPSP/c) auf Kapitel 5 verwiesen.

### 2.2.7 Weitere Modelle und Restriktionen

In der Schedulingliteratur sind zahlreiche Erweiterungen der in den vorangegangenen Abschnitten vorgestellten Modelle zu finden. Im Folgenden werden exemplarisch einige Erweiterungen qualitativ erläutert und mit Verweisen auf weiterführende Literatur versehen.

Bisher wurde stets davon ausgegangen, dass Vorgänge innerhalb eines zusammenhängenden Zeitraums ausgeführt werden müssen. Das Konzept der *unterbrechbaren Vorgänge* erlaubt es hingegen, die Ausführung von Vorgängen anzuhalten und zu einem späteren Zeitpunkt wieder fortzuführen. Im bisher vorgestellten Modell können unterbrechbare Vorgänge abgebildet werden, indem sie von Projektverantwortlichen im Voraus in Teilvorgänge aufgespalten werden. Die Teilvorgänge müssen über Vorrangbeziehungen zu einer Kette verknüpft werden. Es ist somit eine modellexogene Entscheidung darüber zu treffen, an welchen Stellen Unterbrechungen der Vorgänge möglich sind. Für modellendogene Entscheidungen über Vorgangsunterbrechungen werden zwei Erweiterungen der vorgestellten Modelle unterschieden (Quintanilla u. a. 2015, S. 232):

- Bei diskreten Unterbrechungen können Vorgänge zu jedem diskreten Zeitpunkt unterbrochen werden. Die möglichen Unterbrechungsstellen eines Vorgangs sind somit von der Feinheit der vorgenommenen Diskretisierung der Zeit abhängig. Das Schedulingproblem mit diskreten Unterbrechungen kann mit den vorgestellten Modellen behandelt werden, indem alle Vorgänge in Teilvorgänge mit einer Dauer von jeweils einer Periode aufgeteilt und über eine Kette von Vorrangbeziehungen miteinander verknüpft werden (Quintanilla u. a. 2015, S. 233; Demeulemeester und Herroelen 1996, S. 3).
- Bei kontinuierlichen Unterbrechungen können Vorgänge zu jedem beliebigen Zeitpunkt unterbrochen werden. In einem Schedulingproblem zu diesem Modell sind damit nicht mehr Entscheidungen zu den Startzeitpunkten der Vorgänge zu treffen, sondern zu den Ausführungszeiträumen der Vorgänge. Schwindt und Paetz (2015, S. 259) verwenden als Entscheidungsvariable eine Funktion, die jedem Zeitpunkt des kontinuierlichen Zeithorizonts den relativen Fortschritt jedes Vorgangs zuordnet.

Es gibt zahlreiche Varianten von Schedulingproblemen mit unterbrechbaren Vorgängen. Beispielsweise berücksichtigen Quintanilla u. a. (2015, S. 233) Restriktionen zur maximalen Anzahl der Unterbrechungen und zur minimalen erforderlichen Ausführungsdauer eines Vorgangs zwischen zwei Unterbrechungen.

Um eine Lösung eines Schedulingproblems in die Praxis zu überführen, muss der modellier- te Zeithorizont auf den realen Zeithorizont übertragen werden. Das bedeutet beispielsweise, dass ein Tag  $t$  im Modell einem Tag  $t + 2$  in der Realität entsprechen kann, da zwischen Projektstart und Tag  $t$  zwei arbeitsfreie Tage liegen. Es wurde bisher davon ausgegangen, dass arbeitsfreie Tage keine Auswirkungen auf das Scheduling haben: Wenn während der Ausführung eines Vorgangs ein arbeitsfreier Tag liegt, wird angenommen, dass der Vorgang am folgenden Arbeitstag ohne Einschränkung fortgesetzt werden kann. Des-

halb sind arbeitsfreie Tage nicht Bestandteil des Modells, sondern werden erst bei der Überführung einer Lösung in die Praxis eingefügt. Wenn es allerdings Vorgänge gibt, die explizit nicht durch arbeitsfreie Tage unterbrochen werden dürfen, macht dies eine Modellerweiterung erforderlich. In dieser Modellerweiterung wird der reale Zeithorizont inklusive arbeitsfreier Tagen abgebildet. In der englischsprachigen Schedulingliteratur wird in diesem Zusammenhang von *Calendarization* gesprochen. Für Details wird auf Neumann, Schwindt und Zimmermann (2003, S. 120ff) verwiesen, worauf die Ausführungen dieses Absatzes beruhen.

In bestimmten praktischen Problemstellungen kann die Nutzung erneuerbarer Ressourcen durch verschiedene Vorgänge *Umrüstzeiten* erforderlich machen. Wenn beispielsweise eine Maschine nach der Nutzung durch einen bestimmten Vorgang gereinigt werden muss, kann dies als Umrüstzeit modelliert werden. Im einfacheren Fall sind die Umrüstzeiten *reihenfolgeunabhängig*, d. h., sie basieren nur auf demjenigen Vorgang, der eine erneuerbare Ressource als Nächstes beansprucht (Neumann, Schwindt und Zimmermann 2003, S. 113). Aufwendiger sind im Gegensatz dazu *reihenfolgeabhängige* Umrüstzeiten, die auf allen vorangegangenen Vorgängen basieren, die eine erneuerbare Ressource bereits beansprucht haben (Neumann, Schwindt und Zimmermann 2003, S. 150ff).

Die *Kostenplanung* ist neben der Terminplanung eine weitere wesentliche Themengruppe des Projektmanagements (DIN Deutsches Institut für Normung 2016, S. 20f). In der Literatur sind zahlreiche Modelle zu finden, die die Kosten in das Scheduling einbinden. Damit wird eine integrierte Termin- und Kostenplanung mit Methoden des Operations Research ermöglicht. Kosten werden dem Schedulingproblem dazu zunächst in Form zusätzlicher Problemparameter zugänglich gemacht, beispielsweise als *vorgangsbezogene* oder *ressourcenbezogene Kosten*. In das Schedulingproblem können dann zusätzliche Restriktionen aufgenommen werden, die beispielsweise die Einhaltung von Budgetrestriktionen sicherstellen. Darüber hinaus spielen Kosten für die Zielsetzungen in der Regel eine bedeutende Rolle (vgl. nächster Abschnitt 2.2.8). Schedulingprobleme unter Berücksichtigung von Kosten behandeln beispielsweise Möhring (1984), Demeulemeester, Herroelen und Elmaghraby (1996) oder He u. a. (2015) mit jeweils unterschiedlichen Schwerpunkten.

### 2.2.8 Zielsetzungen

Eine Zielsetzung im Schedulingproblem wird mit einer Zielfunktion und einer Angabe zur Präferenz hoher oder niedriger Werte abgebildet. Ohne Beschränkung der Allgemeinheit kann eine Minimierungspräferenz angenommen werden (vgl. Abschnitt 2.2.1). Für alle bisher vorgestellten Schedulingprobleme wurde die generische Zielsetzung  $\min_S z(S)$  (bzw.  $\min_{S,M} z(S, M)$  im Mehr-Modus-Fall) verwendet. Bei der Entwicklung und Anwendung von Lösungsverfahren ist jedoch die konkrete Zielsetzung von besonderer Bedeutung. Deshalb muss eine vollständige Formulierung eines Schedulingproblems eine konkrete Zielsetzung enthalten. In diesem Abschnitt werden einige relevante Zielsetzungen vorgestellt.

Ein hinsichtlich der Lösung von Schedulingproblemen wesentliches Unterscheidungskriterium der Zielsetzungen ist die *Regularität*. Eine Zielsetzung wird als *regulär* bezeichnet, falls

der Zielfunktionswert monoton steigend für ansteigende Startzeitpunkte  $S_i$  ist. Praktisch bedeutet eine reguläre Zielsetzung, dass die zeitliche Rechtsverschiebung eines Vorgangs nie zu einem besseren Zielfunktionswert führen kann. Formal ausgedrückt muss für eine reguläre Zielsetzung bei  $S \leq S'$  (wobei „ $\leq$ “ komponentenweise zu interpretieren ist) gelten, dass bei Minimierungspräferenz  $z(S) \leq z(S')$  (Neumann, Schwindt und Zimmermann 2003, S. 117). In der Regel können Lösungsverfahren, die für eine reguläre Zielsetzung entwickelt wurden, auch zur Lösung von Problemen mit anderen regulären Zielsetzungen verwendet werden. Das folgt daraus, dass ein optimaler Schedule für ein Problem mit einer regulären Zielsetzung gleichzeitig auch der optimale Schedule für alle anderen Probleme mit regulärer Zielsetzung ist (Neumann, Schwindt und Zimmermann 2003, S. 118).

Eine weitere Unterscheidung erfolgt nach Węglarz u. a. (2011) mit der Klassifizierung in *zeitabhängige*, *ressourcenabhängige* und *finanzielle Zielsetzungen*. Jede dieser drei Klassen kann sowohl reguläre als auch nichtreguläre Zielsetzungen enthalten. Im Folgenden werden die Klassen ausführlicher betrachtet.

Die klassische zeitabhängige Zielsetzung ist die *Minimierung der Projektdauer*. Falls ein fiktiver Vorgang  $I + 1$  zur Abbildung des Projektendes erzeugt wurde, kann die Minimierung der Projektdauer unter Verwendung der Zielfunktion  $z(S) := S_{I+1}$  ausgedrückt werden. Bei der Minimierung der Projektdauer handelt es sich um eine der am häufigsten in der Literatur betrachteten Zielsetzungen (Węglarz u. a. 2011, S. 182). Eine weitere zeitabhängige Zielsetzung ist die *Minimierung der gewichteten Fertigstellungszeitpunkte* (engl. weighted flow time) (Neumann, Schwindt und Zimmermann 2003, S. 118). Sie kann unter Verwendung der Zielfunktion  $z(S) := \sum_{i \in V} w_i (S_i + d_i)$  ausgedrückt werden, wobei  $w_i$  das Gewicht eines Vorgangs ist. Je größer das Gewicht eines Vorgangs, desto stärker wird seine Fertigstellung priorisiert. Sind für die Vorgänge  $i \in V$  *Fälligkeitstermine*  $\bar{d}_i$  vorgegeben, wird dadurch die Zielsetzung der *Minimierung der gewichteten Verspätung* (engl. weighted tardiness) motiviert (Neumann, Schwindt und Zimmermann 2003, S. 118). Die *Verspätung* eines Vorgangs  $i \in V$  entspricht  $\max\{S_i + d_i - \bar{d}_i, 0\}$ . Die Zielfunktion kann somit als  $z(S) := \sum_{i \in V} w_i \max\{S_i + d_i - \bar{d}_i, 0\}$  formuliert werden. Je größer das Gewicht eines Vorgangs, desto stärker wird seine verspätete Fertigstellung bestraft. Eng verwandt damit ist auch die Zielsetzung zur *Minimierung der maximalen Verspätung*, d. h.  $z(S) := \max_{i \in V} (w_i \max\{S_i + d_i - \bar{d}_i, 0\})$  (Węglarz u. a. 2011, S. 182). Bei allen in diesem Absatz genannten zeitabhängigen Zielsetzungen handelt es sich um reguläre Zielsetzungen (Węglarz u. a. 2011, S. 182).

Ressourcenabhängige Zielsetzungen sind Funktionen der Ressourcennutzungen (Węglarz u. a. 2011, S. 182). Es sei darauf hingewiesen, dass die Ressourcennutzungen ihrerseits wiederum Funktionen der Zeit sind, sodass im weiteren Sinne alle Zielsetzungen zeitabhängig sind. Eine häufig in der Literatur betrachtete ressourcenabhängige Zielsetzung ist die *Minimierung der Ressourcenbeschaffungskosten*. Ein Schedulingproblem mit dieser Zielsetzung wird als *Ressourceninvestitionsproblem* (engl. resource investment problem; oder: resource availability cost problem) bezeichnet. Die maximalen Ressourcennutzungen werden dabei mit Kosten bewertet, welche in Summe minimiert werden sollen. Da die Projektdauer nicht mehr Teil der Zielsetzung ist, wird in der Regel eine Restriktion zur Festschreibung einer maximalen Projektdauer  $\bar{d}$  eingeführt. Die Zielfunktion kann dann

als  $z(S) := \sum_{k \in \mathcal{R}} c_k \max_{t \in [0, \bar{d}]} r_k(S, t)$  formuliert werden, wobei  $\mathcal{R}$  die Menge eines beliebigen Ressourcentyps,  $c_k$  der Kostenfaktor einer Ressource  $k \in \mathcal{R}$  und  $r_k(S, t)$  die Nutzung von  $k$  zu einem Zeitpunkt  $t$  bei gegebenem Schedule  $S$  (Zimmermann u. a. 2006, S. 122) ist. Das Ressourceninvestitionsproblem wurde erstmalig von Möhring (1984) diskutiert. Falls eine Ressourcennutzung erst ab einem bestimmten Niveau mit Kosten verbunden ist, führt dies zur Zielsetzung der *Minimierung der Überschreitungskosten*. Die Zielfunktion kann als  $z(S) := \sum_{k \in \mathcal{R}} c_k \int_{t \in [0, \bar{d}]} [r_k(S, t) - Y_k]^+ dt$  formuliert werden, wobei  $Y_k$  dem Niveau entspricht, ab welchem die Nutzung der Ressource  $k \in \mathcal{R}$  mit Kosten  $c_k$  verbunden ist (Zimmermann u. a. 2006, S. 122). Die Notation  $[x]^+$  bedeutet, dass nur über positive Werte von  $x$  integriert wird und negative Werte als null einfließen. Ein Schedulingproblem mit der Zielsetzung zur Minimierung der Überschreitungskosten wird als *Ressourcenabweichungsproblem* bezeichnet. Eine weitere ressourcenabhängige Zielsetzung ist die *Ressourcennivellierung*, welche die Absicht einer möglichst gleichmäßigen Auslastung der Ressourcen verfolgt. Die Zielfunktion kann je nach gewünschter Bestrafung der Abweichungen von einer gleichmäßigen Auslastung definiert werden. Beispielsweise eignet sich die integrierte, quadrierte Ressourcennutzung:  $z(S) := \sum_{k \in \mathcal{R}} \int_{t \in [0, \bar{d}]} r_k^2(S, t) dt$  (Zimmermann u. a. 2006, S. 123). Ein Schedulingproblem mit dieser Zielsetzung wird als *Ressourcennivellierungsproblem* bezeichnet. Bei allen in diesem Absatz genannten ressourcenabhängigen Zielsetzungen handelt es sich um nichtreguläre Zielsetzungen (Zimmermann u. a. 2006, S. 163).

Finanzielle Zielsetzungen berücksichtigen eingehende und ausgehende Zahlungsströme (Węglarz u. a. 2011, S. 183). Da einige ressourcenabhängige Zielsetzungen, wie beispielsweise die Minimierung der Ressourcenbeschaffungskosten, bereits ausgehende Zahlungsströme enthalten, können sie auch als Spezialfall einer finanziellen Zielsetzung betrachtet werden. Üblicherweise zeichnen sich finanzielle Zielsetzungen jedoch durch die Betrachtung von vorgangsbezogenen Kosten aus. Eine am unternehmerischen Ziel der Gewinnmaximierung ausgerichtete Zielsetzung ist die *Maximierung des Kapitalwertes* (Zimmermann u. a. 2006, S. 121). Die Zielfunktion kann (unter Verwendung einer Minimierungspräferenz) als  $z(S) := - \sum_{i \in V} c_i v^{S_i}$  ausgedrückt werden, wobei  $c_i$  der Höhe des mit dem Start des Vorgangs  $i$  ausgelösten Zahlungsstroms entspricht und  $v$  ein Diskontfaktor ist (Zimmermann u. a. 2006, S. 121).

Wenn in einem Schedulingproblem mehrere *Einzelziele* gleichzeitig betrachtet werden sollen, handelt es sich um ein *multikriterielles Schedulingproblem*. Oft stehen diese Einzelziele in einem Konflikt zueinander, der nur durch eine modellexogene Entscheidung der Projektverantwortlichen aufgelöst werden kann. Im Operations Research gibt es verschiedene Konzepte, diese Entscheidung in ein Optimierungsproblem bzw. ein Lösungsverfahren einzubinden. Sie lassen sich in folgende drei Kategorien einteilen (Hwang und Masud 1979; zitiert nach Ballestín und Blanco 2015, S. 412):

- Die Projektverantwortlichen definieren ein Gesamtziel als Funktion der Einzelziele. Über die Gestaltung der Funktion, in welcher die Einzelziele in der Regel auf eine bestimmte Weise gewichtet werden, muss modellexogen entschieden werden. Dem Lösungsverfahren ist nur das Gesamtziel bekannt.

- Es werden mehrere Lösungen zu einem Problem berechnet und in einem *Lösungspool* gesammelt an die Projektverantwortlichen ausgegeben. Dabei sind die Einzelziele dem Lösungsverfahren im Idealfall bekannt, sodass diejenigen Lösungen berechnet werden können, in denen keine Verbesserung hinsichtlich eines bestimmten Einzelziels ohne eine Schlechterstellung hinsichtlich eines anderen Einzelziels möglich ist (die sogenannte *Paretofront*). Die Projektverantwortlichen wählen dann eine Lösung aus dem Lösungspool, welche in der Praxis ausgeführt wird. Hierbei handelt es sich somit ebenfalls um eine modellexogene Entscheidung.
- Die Projektverantwortlichen erhalten während der Durchführung des Lösungsverfahrens die Möglichkeit, abhängig vom jeweiligen Stand der Optimierung, modellexogene Entscheidungen hinsichtlich der Einzelziele zu treffen. Sie können damit das Lösungsverfahren in die gewünschte Richtung leiten.

### 2.3 Lösungsverfahren

In diesem Abschnitt werden Grundlagen zu Lösungsverfahren für Schedulingprobleme dargelegt. Die Darlegung erfolgt überwiegend anhand des (RCPSp) und des (RCPSp/max), da es sich hierbei um die klassischen Probleme des Scheduling handelt. Lösungsverfahren für das (RCPSp/c) sind für diese Arbeit von zentraler Bedeutung und werden daher in den weiterführenden Kapiteln ausführlich behandelt.

Kolisch (1996b, S. 321) unterscheidet bei den exakten Lösungsverfahren für Schedulingprobleme zwischen dynamischer Programmierung, ganzzahliger Programmierung und Enumerationsansätzen, die die Basis von Branch-and-Bound-Verfahren bilden. Węglarz u. a. (2011, S. 187) stellen in ihrer Übersichtsarbeit fest, dass nahezu alle exakten Lösungsansätze auf Branch-and-Bound-Verfahren basieren. Laut Herroelen (2005, S. 420) sind exakte Lösungsverfahren im Projektscheduling nicht weit verbreitet. Begründet wird dies damit, dass bereits viele Instanzen mit mehr als 60 Vorgängen nicht mehr exakt gelöst werden können. Studien haben allerdings gezeigt, dass praktische Problemstellungen mehr Vorgänge enthalten (z. B. Liberatore u. a. 2001; zitiert nach Herroelen 2005, S. 420). Selbst wenn die Rechenleistungen im Gegensatz zum Jahr 2005, in welchem Herroelen (2005) ihre Aussage getroffen hatten, stark zugenommen haben, bleibt die exakte Lösung des (RCPSp) aufgrund seiner NP-Schwere für große Probleminstanzen eine Herausforderung (Agarwal u. a. 2015, S. 58). Dies wird in auch in Coelho und Vanhoucke (2020, S. 9) belegt, wo Instanzen mit nur 30 Vorgängen betrachtet werden, welche auf Hochleistungsrechnern innerhalb 20-stündiger Rechenzeit nicht optimal gelöst werden konnten. Da in dieser Arbeit ein Lösungsverfahren entwickelt werden soll, welches auch mit sehr großen Probleminstanzen umgehen kann, wird im vorliegenden Grundlagenkapitel nicht weiter auf exakte Lösungsverfahren eingegangen.

Bei Heuristiken unterscheidet Kolisch (1996b, S. 321) zwischen prioritätsregelbasierten Verfahren, gekürzten Branch-and-Bound-Verfahren, auf ganzzahliger Programmierung basierenden Heuristiken, auf disjunktiven Pfeilen basierenden Konzepten (engl. disjunc-

tive arc concepts) und lokaler Suche. Węglarz u. a. (2011, S. 189) ergänzen diese Kategorisierung um Metaheuristiken. In Abschnitt 2.3.1 werden zunächst zwei sogenannte Generierungsschemas vorgestellt, mit denen sich zulässige Lösungen für das (RCPSP) und das (RCPSP/max) berechnen lassen. Sie bilden zusammen mit einer Prioritätsregel die Komponenten eines prioritätsregelbasierten Verfahrens (Kolisch 1996b, S. 322), welches in Abschnitt 2.3.2 erläutert wird. Auf Metaheuristiken wird in Abschnitt 2.3.3 eingegangen. Andere der oben genannten Heuristiken sind nicht Teil dieses Grundlagenkapitels, da sie nicht in Verbindung mit den weiteren Inhalten dieser Arbeit stehen.

### 2.3.1 Generierungsschemas

*Generierungsschemas* sind Algorithmen zur Berechnung zulässiger Lösungen für Schedulingprobleme. Darin werden iterativ Startzeitpunkte für alle Vorgänge bestimmt. Da das (TPSP) und das (TPSP/max) bereits in Polynomialzeit optimal gelöst werden können, sind für diese Probleme keine Generierungsschemas erforderlich. Zur Lösung komplexer Schedulingprobleme unter Ressourcenrestriktionen sind Generierungsschemas jedoch von zentraler Bedeutung, wie auch in Abschnitt 2.3.3 bei der Vorstellung von Metaheuristiken gezeigt werden wird.

In der Literatur werden zwei Generierungsschemas unterschieden (Kolisch 2015, S. 2): Das *serielle* und das *parallele* Generierungsschema. Grundsätzlich unterscheiden sich beide Generierungsschemas dadurch, dass die serielle Variante über Vorgänge und die parallele Variante über Zeitpunkte iteriert (Kolisch 2015, S. 9 und S. 11). Die nachfolgenden Darlegungen basieren auf Kolisch (1996b), Zimmermann u. a. (2006, S. 248ff) und Kolisch (2015).

Zunächst wird ausführlich auf das **serielle Generierungsschema** (engl. serial schedule generation scheme) eingegangen. Algorithmus 1 beschreibt das serielle Generierungsschema formal. Initial sind die zu bestimmenden Startzeitpunkte  $S_i$  für alle Vorgänge  $i = 0, \dots, I + 1$  unbekannt, was durch eine Dummybelegung von  $S_i = -\infty$  ausgedrückt wird (vgl. Zeile 1). Damit gelten die Vorgänge als *uneingeplant*. Ein Vorgang mit  $S_i \geq 0$  wird hingegen als *eingepplant* bezeichnet. Solange mindestens ein uneingeplanter Vorgang existiert, wird die vorhandene Lösung als *partieller Schedule* bezeichnet. In Zeile 2 wird solange eine Schleife durchlaufen, bis alle Vorgänge eingepplant sind. Da der Zeithorizont im (RCPSP) unbegrenzt ist, wird, sofern  $\bar{I}_{\text{RCPSP}}$  nicht unzulässig ist, in jeder Iteration der Schleife ein zulässiger Startzeitpunkt gefunden werden. Deshalb entspricht die Anzahl der Iterationen im seriellen Generierungsschema der Anzahl der Vorgänge  $I + 1$ . In Zeile 3 wird die Menge der einplanbaren Vorgänge  $\mathcal{E}$  als diejenige Menge der noch uneingeplanten Vorgänge bestimmt, deren direkte Vorgänger gemäß Gleichung (2.1) eingepplant sind. Dadurch wird die Einhaltung der Vorrangrestriktionen sichergestellt. Anschließend wird in Zeile 4 ein Vorgang  $i^*$  aus  $\mathcal{E}$  ausgewählt. Entscheidungskriterien zu dieser Auswahl werden im nächsten Abschnitt behandelt. In Zeile 5 wird für  $i^*$  der frühestmögliche vorrang- und ressourcenzulässige Zeitpunkt als Startzeitpunkt  $S_{i^*}$  bestimmt. Die Belegung der Entscheidungsvariable  $S_{i^*}$  mit einem Wert  $\geq 0$  wird als *Einplanung* von Vorgang  $i^*$  bezeichnet.

Das erneuerbare Ressourcenprofil wird in effizienten Implementierungen nicht in jeder Iteration neu berechnet, sondern bei der Einplanung eines Vorgangs jeweils über seinen Ausführungszeitraum aktualisiert. Für Erläuterungen zur effizienten Verwaltung des Ressourcenprofils wird auf Neumann, Schwindt und Zimmermann (2003, S. 87) verwiesen. Laut Kolisch (2015, S. 10) kann Algorithmus 1 mit einer Zeitkomplexität von  $O(|V|^2|\mathcal{R}^\alpha|)$  implementiert werden.

---

**Algorithmus 1:** SerialScheduleGenerationScheme

---

**Eingabe:** Instanz  $\mathcal{I}_{\text{RCPSP}}$

- 1 initialisiere partiellen Schedule  $S_i := -\infty$  für  $i = 0, \dots, I + 1$ ;
- 2 **solange** Vorgänge  $i$  mit  $S_i = -\infty$  verbleiben **tue**
- 3     bestimme  $\mathcal{E} := \{i \in V \mid S_i = -\infty \wedge S_{i'} \geq 0 \text{ für alle } i' \in \text{Pred}(i)\}$ ;
- 4     wähle einen Vorgang  $i^* \in \mathcal{E}$ ;
- 5     bestimme  $S_{i^*} := \min\{t \geq 0 \mid t \geq S_{i'} + d_{i'} \text{ für alle } i' \in \text{Pred}(i^*)$   
 $\quad \wedge r_k^\alpha(S, \tau) + r_{i^*k}^\alpha \leq R_k^\alpha \text{ für alle } t \leq \tau < t + d_{i^*} \text{ und alle } k \in \mathcal{R}^\alpha\}$ ;
- 6 **Ende**

**Ausgabe:** zulässiger Schedule für  $\mathcal{I}_{\text{RCPSP}}$

---

Algorithmus 1 kann ohne Weiteres auch auf solche Instanzen des (RCPSP/max) angewendet werden, die ausschließlich zeitliche Mindestabstände enthalten. In diesem Fall enthält der Netzplan  $N^{\text{temp}}$  keine Zyklen und die Menge der direkten Vorgänger kann analog zu Gleichung (2.1) bestimmt werden. Es wird auch davon gesprochen, dass sich hier die Knoten im Netzplan *topologisch sortieren* lassen. Übertragen auf die Vorgänge bedeutet dies, jedem Vorgang  $i$ , der direkter Vorgänger eines anderen Vorgangs  $i'$  ist, lässt sich in einer topologisch sortierten Folge eine Position vor  $i'$  zuweisen. Eine solche Folge wird auch als *vorrangzulässig* bezeichnet. Falls eine Instanz des (RCPSP/max) jedoch zeitliche Höchstabstände enthält, sind wie in Abschnitt 2.2.3 erläutert, Zyklen in  $N^{\text{temp}}$  möglich. Der Netzplan lässt sich unter Zuhilfenahme der Gleichung (2.1) nicht topologisch sortieren, da gleichzeitig Zeitbeziehungen  $(i, i')$  und  $(i', i)$  zwischen zwei Vorgängen  $i \neq i'$  existieren können. Um Algorithmus 1 anzuwenden, ist deshalb die in Gleichung (2.6) definierte Menge der direkten Vorgänger  $\text{Pred}^{\text{temp}}$  zu verwenden. Mit der damit vorgegebenen strengen Ordnung der Vorgänge ist allerdings nur sichergestellt, dass zeitliche Mindestabstände eingehalten werden. Die Einhaltung der zeitlichen Höchstabstände muss in einer zusätzlichen Verzweigung des Algorithmus überprüft werden: Falls in einer Iteration zeitliche Höchstabstände verletzt sind, ist ein sogenannter *Ausplanungsschritt* erforderlich. Dabei werden alle Vorgänge ausgeplant, d. h.  $S_i := -\infty$  gesetzt, zu denen der zeitliche Höchstabstand nicht eingehalten werden kann. Diese Vorgänge werden nach rechts verschoben, d. h., ihre frühestmöglichen Startzeitpunkte werden hinreichend erhöht, sodass die zeitlichen Höchstabstände potenziell eingehalten werden können. In der darauffolgenden Iteration können dieselben zeitlichen Höchstabstände aufgrund der Ressourcenrestriktionen allerdings erneut verletzt werden. Deshalb ist beim seriellen Generierungsschema für das (RCPSP/max) nicht garantiert, dass eine zulässige Lösung gefunden wird. Das Generierungsschema kann auch *fehlschlagen*. Dies steht in Einklang mit der in Abschnitt 2.2.4 getroffenen Aussage, dass das Entscheidungsproblem zur Existenz einer zulässigen Lösung



für das (RCPSP/max) NP-vollständig ist. In Neumann, Schwindt und Zimmermann (2003, S. 85ff) oder Zimmermann u. a. (2006, S. 249ff) wird das serielle Generierungsschema für das (RCPSP/max) ausführlich beschrieben. Darin ist auch belegt, dass die Ausplanungsschritte die Zeitkomplexität des Algorithmus erhöhen. Beispiel 2.2 veranschaulicht eine Situation, in der das serielle Generierungsschema für ein (RCPSP/max) fehlschlagen würde.

**Beispiel 2.2:** Eine Instanz  $\mathcal{I}_{\text{RCPSP/max}}$  besteht aus zwei realen Vorgängen 1 und 2 mit Dauern von  $d_1 = d_2 = 2$  und Nutzungen einer erneuerbaren Ressource R1 in Höhe von  $r_{1,\text{R1}}^\alpha = r_{2,\text{R1}}^\alpha := 1$ . Für die Ressourcenkapazität gilt  $R_{\text{R1}}^\alpha := 1$ . Außerdem existieren Zeitabstände von  $\delta_{1,2} = 0$  und  $\delta_{2,1} = -1$ , d. h., Vorgang 2 darf nicht früher als Vorgang 1 starten und gleichzeitig höchstens eine Periode nach Vorgang 1 starten. Aufgrund der begrenzten Ressourcenkapazität können sich beide Vorgänge nie zeitgleich in Ausführung befinden. Dennoch müssen sie sich aufgrund der Zeitbeziehungen um mindestens eine Periode überschneiden. Das serielle Generierungsschema würde wiederholt Ausplanungsschritte durchführen, ohne Vorgang 2 jemals zulässig einplanen zu können. Während in diesem einfachen Beispiel ersichtlich wäre, dass  $\mathcal{I}_{\text{RCPSP/max}}$  unlösbar ist, ist dies aufgrund der Komplexität des (RCPSP/max) generell nicht in Polynomialzeit erkennbar.  $\square$

Das **parallele Generierungsschema** für das (RCPSP) iteriert über Zeitpunkte. Zu jedem betrachteten Zeitpunkt werden so viele Vorgänge eingeplant, wie unter Einhaltung der Vorrang- und Ressourcenrestriktionen möglich. Die Auswahl der einzuplanenden Vorgänge erfolgt analog zur seriellen Variante anhand eines Entscheidungskriteriums, worauf im nächsten Abschnitt eingegangen wird. Wenn keine Vorgänge mehr eingeplant werden können, wird der betrachtete Zeitpunkt auf den nächstmöglichen Endzeitpunkt eines eingeplanten Vorgangs gesetzt, d. h. auf denjenigen Zeitpunkt, zu dem potenziell wieder Ressourcen freigegeben werden. Für jede Instanz wird eine zulässige Lösung gefunden, falls die Instanz zulässig ist. Ein Algorithmus für das parallele Generierungsschema ist beispielsweise in Kolisch (2015, S. 12) zu finden. An dieser Stelle wird aus folgenden Gründen nicht weiter auf das parallele Generierungsschema eingegangen: In einer Studie von Kolisch (1996b, S. 330) erzielte das serielle Generierungsschema bessere Lösungen, wenn es vielfach (mehr als 40 Mal) mit unterschiedlichen Entscheidungskriterien für die Auswahl der einzuplanenden Vorgängen auf eine Instanz angewendet wurde. Deshalb wird das serielle Generierungsschema laut Kolisch (2015, S. 13) auch überwiegend in Metaheuristiken eingesetzt. Darüber hinaus ist bei Verwendung des parallelen Generierungsschemas in Verbindung mit regulären Zielfunktionen nicht sichergestellt, dass der durchsuchte Lösungsraum die optimale Lösung enthält (Sprecher, Kolisch u. a. 1995, S. 12).

Bei der Anwendung des parallelen Generierungsschemas auf das (RCPSP/max) können aufgrund der zeitlichen Höchstabstände Ausplanungsschritte erforderlich werden. Für einen Algorithmus wird auf Neumann, Schwindt und Zimmermann (2003, S. 91) verwiesen. Die geringere Verbreitung des parallelen Generierungsschemas im Vergleich zum seriellen ist auch daran ersichtlich, dass im Lehrbuch von Zimmermann u. a. (2006) keine Ausführungen zum parallelen Generierungsschema enthalten sind.

### 2.3.2 Prioritätsregelbasierte Verfahren

*Prioritätsregelbasierte Verfahren* sind Heuristiken zur Lösung von Schedulingproblemen, die aus einem Generierungsschema und einer *Prioritätsregel* bestehen Kolisch (1996b, S. 322). Die nachfolgenden Darlegungen basieren auf Kolisch (1996b), Zimmermann u. a. (2006, S. 248ff) und Kolisch (2015). Die Prioritätsregel dient als Entscheidungskriterium zur Auswahl eines einzuplanenden Vorgangs in einer Iteration eines Generierungsschemas (vgl. Zeile 4 in Algorithmus 1). Bekannte Prioritätsregeln sind beispielsweise „Most Total Successors“, „Smallest Latest Start Time“ oder „Minimum Slack“. Diese Prioritätsregeln geben vor, dass Vorgänge mit mehr transitiven Nachfolgern, kleineren spätestmöglichen Endzeitpunkten bzw. kleineren Gesamtpuffern mit höherer Priorität eingeplant werden sollen. Die Prioritätsregel wird allerdings nur als Entscheidungskriterium unter den einplanbaren Vorgängen  $\mathcal{E}$  herangezogen. Das heißt, die Zulässigkeit der Vorrangrestriktionen ist stets sicherzustellen bevor die Priorität betrachtet wird.

Es wird zwischen *statischen* und *dynamischen* Prioritätsregeln unterschieden (Neumann, Schwindt und Zimmermann 2003, S. 85). Während bei statischen Prioritätsregeln eine einmalige Berechnung der Priorität jedes Vorgangs zu Beginn des Lösungsverfahrens ausreicht, müssen die Prioritäten bei dynamischen Prioritätsregeln in jeder Iteration des Generierungsschemas neu berechnet werden. Die Prioritätsregeln Smallest Latest Start Time und Minimum Slack können beispielsweise als statische oder dynamische Prioritätsregeln umgesetzt werden, wobei ohne explizite Nennung üblicherweise die statische Variante gemeint ist.

Nach Neumann, Schwindt und Zimmermann (2003, S. 84) wird die Priorität jedes Vorgangs  $i \in V$  als  $\pi(i)$  bezeichnet. Bei gleicher Priorität wird per Konvention derjenige Vorgang mit der kleineren Nummer gewählt. Für ein prioritätsregelbasiertes Verfahren wird Zeile 4 in Algorithmus 1 durch

$$i^* := \min\{i \in \mathcal{E} \mid \pi(i) = \underset{i' \in \mathcal{E}}{\text{ext}} \pi(i')\}$$

ersetzt, wobei  $\text{ext} \in \{\min, \max\}$  gemäß gewählter Prioritätsregel gesetzt wird. Als zusätzliche Eingabe für Algorithmus 1 wird die Prioritätsregel benötigt.

Das serielle Generierungsschema kann umgestaltet werden, sodass es über eine vorrangzulässige *Vorgangsfolge*  $\ell = (0, i_1, i_2, \dots, i_I, I+1)$  (in der Literatur auch Vorgangsliste genannt) iteriert (Kolisch 2015, S. 11). Dazu müssen lediglich die Schritte zur Bestimmung der einplanbaren Vorgänge  $\mathcal{E}$  (Zeile 3) sowie die Auswahl des einzuplanenden Vorgangs  $i^*$  (Zeile 4) in einen dem Generierungsschema vorgeschalteten Schritt ausgelagert werden. Die Prioritätsregel ist damit in die Vorgangsfolge eingebettet und wird nicht mehr explizit an das serielle Generierungsschema übergeben. Die Vorgangsfolge kann als eine *Codierung* einer Lösung aufgefasst werden und die Anwendung des seriellen Generierungsschemas darauf entspricht der Decodierung (Kolisch 2015, S. 11). Während bei der Decodierung aus jeder Vorgangsfolge ein eindeutiger Schedule hervorgeht, lässt sich umgekehrt ein bestehender Schedule nicht immer auf eine eindeutige Vorgangsfolge zurückführen (Hartmann 1998, S. 736). Neben der Vorgangsfolge unterscheiden Kolisch und Hartmann (1999)

(zitiert nach van Peteghem und Vanhoucke 2014, S. 63) noch weitere Codierungsarten. Auf Details wird hier nicht eingegangen, da die Vorgangsfolge laut Kolisch und Hartmann (2006, S. 35) die verbreitetste Codierart darstellt. Hartmann und Kolisch (2000, S. 10) sehen einen wesentlichen Vorteil der Vorgangsfolge in der höheren Leistungsfähigkeit des seriellen Generierungsschemas, die durch das Entfernen der Zeilen 3 und 4 in Algorithmus 1 erreicht wird. Üblicherweise arbeiten Heuristiken zur Lösung des (RCPSP) auf Ebene einer Codierung und nicht direkt auf Ebene des Schedules (z. B. Geiger 2017; Hartmann 1998), da dies laut Debels u. a. (2006, S. 5) aus Gründen der Effektivität und Handhabbarkeit sinnvoll ist. So ist über den Decodierungsprozess mit Hilfe des seriellen Generierungsschemas stets sichergestellt, dass Lösungen zulässig sind. Dennoch gibt es in der Literatur auch vereinzelt Heuristiken, die direkt auf Ebene des Schedules arbeiten, wie z. B. Thomas und Salhi (1998).

Ein Durchlauf eines Generierungsschemas mit einer bestimmten Prioritätsregel wird als *single-pass* prioritätsregelbasiertes Verfahren bezeichnet. Im Gegensatz dazu wird bei mehrfachen Aufrufen eines Generierungsschemas mit unterschiedlichen Prioritätsregeln von einem *multi-pass* prioritätsregelbasierten Verfahren gesprochen. Bei Letzterem kann entweder auf deterministische Weise eine vordefinierte Menge an Prioritätsregeln durchlaufen werden. Alternativ können Prioritätsregeln unter Zufallseinfluss systematisch abgeändert werden, um eine gewünschte Variabilität bei der Durchsuchung des Lösungsraums zu erzielen (Kolisch 1996b, S. 324). Ein solches probabilistisches multi-pass prioritätsregelbasiertes Verfahren wird als *Sampling* bezeichnet. Formal wird jedem Vorgang  $i \in \mathcal{E}$  eine Wahrscheinlichkeit zugeordnet, mit der er zur Einplanung ausgewählt wird. Zeile 4 in Algorithmus 1 wird dann durch ein Zufallsexperiment ersetzt, bei welchem unter Berücksichtigung dieser Wahrscheinlichkeiten ein Vorgang aus  $\mathcal{E}$  gezogen wird. Zur Bestimmung der Wahrscheinlichkeiten gibt es unterschiedliche Ansätze, die bis zu einer reinen Gleichverteilung über alle Vorgänge reichen. Für Details wird auf Kolisch (1996b, S. 324) verwiesen.

### 2.3.3 Metaheuristiken

*Metaheuristiken* geben laut Agarwal u. a. (2015, S. 60) eine Strategie vor, nach der der Lösungsraum eines Problems durchsucht wird. Sie laufen in der Regel so lange in einer Schleife weiter, bis ein Abbruchkriterium zutrifft (Agarwal u. a. 2015, S. 60). Übliche Abbruchkriterien sind das Erreichen einer vorgegebenen Rechenzeit oder einer vorgegebenen Anzahl an betrachteten Lösungen. Wie die Vorsilbe „meta“ zu verstehen gibt, sind sie nicht auf ein bestimmtes Problem zugeschnitten, sondern generischer Art.

Aufgrund der NP-Schwere des (RCPSP) werden laut Agarwal u. a. (2015, S. 57 und S. 60) häufig Metaheuristiken zu dessen Lösung eingesetzt. Eine Metaheuristik arbeitet aus den im vorangegangenen Abschnitt erläuterten Gründen üblicherweise auf Ebene der Codierung eines Schedules, d. h. typischerweise auf Ebene der Vorgangsfolge. Die Metaheuristik schreibt beispielsweise vor, unter welchen Voraussetzungen wie viele Änderungen an einer bestehenden Vorgangsfolge vorgenommen werden oder ob eine komplett neue

Vorgangsfolge berechnet wird. Sie schreibt auch vor, welche Vorgangsfolge bei der weiteren Durchsuchung des Lösungsraums als Ausgangspunkt dient. So ist es möglich, jeweils nur von derjenigen Vorgangsfolge auszugehen, die zum besten gefundenen Schedule geführt hat, oder auch von einer schlechteren Vorgangsfolge auszugehen. Letzteres ist mit der Absicht verbunden, lokale Minima zu verlassen.

Auf der Ebene der Codierung liegen keine Informationen zur Lösungsqualität vor. Ob eine betrachtete Vorgangsfolge einen qualitativ hochwertigen Schedule beschreibt oder nicht, kann lediglich mittels Dekodierung festgestellt werden. Durch die Kombination einer generischen Metaheuristik mit einem problemspezifischen Dekodierungsmechanismus wird somit ein Lösungsverfahren konstruiert. Im Fall des (RCPS) werden zur Dekodierung das serielle oder das parallele Generierungsschema eingesetzt. Kolisch (2015, S. 2) bezeichnet die Generierungsschemata daher auch als „Backbone“ der Heuristiken zur Lösung von Schedulingproblemen.

In der Literatur sind zahlreiche Metaheuristiken zu finden. Beispielhaft zu nennen sind genetische Algorithmen, die Tabu-Suche oder die Simulated-Annealing-Metaheuristik. Auch das im vorangegangenen Abschnitt beschriebene Sampling kann aufgrund des Zufallseinflusses als eine (vergleichsweise einfache) Metaheuristik aufgefasst werden (Agarwal u. a. 2015, S. 61 und S. 66). In Kapitel 6 werden eine Multi-Start- und eine Simulated-Annealing-Metaheuristik detailliert vorgestellt, da sie für die Lösungsverfahren dieser Arbeit eingesetzt werden. Für weitere Informationen zu anderen Metaheuristiken – sowohl allgemein als auch im Kontext des (RCPS) – wird auf Agarwal u. a. (2015) verwiesen.

## 2.4 Stoffstrombasiertes Produktionsmanagement

Dieser Abschnitt befasst sich mit ausgewählten Grundlagen zum stoffstrombasierten Produktionsmanagement. Obwohl sich die vorliegende Arbeit mit Planungsproblemen auf dem Gebiet des Projektmanagements beschäftigt, ist ein Verständnis der Stoffströme im Produktionsumfeld aufgrund der Überschneidungen zwischen beiden Disziplinen erforderlich. Insbesondere die im Produktionsmanagement angewandten graphentheoretischen Modelle bilden eine wichtige Basis für die Ausarbeitung eines Stoffstrommodells und dessen Abgrenzung von der Literatur im nachfolgenden Kapitel.

### 2.4.1 Begriffliche Abgrenzung

Die *Produktion* kann als „der gelenkte Einsatz von Gütern und Dienstleistungen [...] zum Abbau von Rohstoffen oder zur Herstellung bzw. Fertigung von Gütern und zur Erzeugung von Dienstleistungen“ definiert werden (Bloech u. a. 2014, S. 3). Das *Produktionsmanagement* umfasst die „Aufgabe der Konkretisierung und Durchsetzung der von der Unternehmensleitung formulierten strategischen Unternehmensziele“ (Schultmann 2003, S. 28). Die *Produktionsplanung* ist schließlich für die „systematische Identifizierung

der gegenwärtigen Handlungsmöglichkeiten und deren Festlegung im Hinblick auf eine möglichst gute Erfüllung der Produktionsziele“ verantwortlich (Schultmann 2003, S. 28).

Das Produktionsmanagement wird mit dem Attribut *stoffstrombasiert* versehen, falls die strategischen Unternehmensziele explizit den Ressourcenverbrauch, die stofflichen und energetischen Emissionen sowie die Effizienz des Stoff- und Energieeinsatzes (bzw. -umsatzes) einbeziehen (Schultmann 2003, S. 48). Diese Definition entstammt jedoch dem Betrachtungshorizont eines Kreislaufwirtschaftssystems. Innerhalb der Unternehmensgrenzen werden in der Regel primär ökonomische Zielsetzungen verfolgt, mit denen die stoffstrombezogenen Zielsetzungen in Konflikt geraten können (vgl. Fröhling 2006, S. 23). Aus diesem Grund wird die Produktionsplanung als Teil des Produktionsmanagements im Kontext dieser Arbeit bereits als *stoffstrombasiert* betrachtet, wenn bei der Identifizierung der gegenwärtigen Handlungsmöglichkeiten die durch Stoffströme hervorgerufenen Rahmenbedingungen und Restriktionen berücksichtigt werden, ohne dass die Stoffströme notwendigerweise in das unternehmerische Zielsystem einfließen.

Die Produktion mit ihrer zugrundeliegenden Absicht zur Herstellung von Gütern unterscheidet in der Regel von einem Projekt, weshalb sich die Projektplanung grundsätzlich auch von der Produktionsplanung unterscheidet (Bloech u. a. 2014, S. 235f). Dennoch kann unter bestimmten Voraussetzungen ein Zusammenhang zwischen beiden Disziplinen hergestellt werden, wie nachfolgend gezeigt wird.

### 2.4.2 Zusammenhang zwischen Projektplanung und Produktionsplanung

Die Ablaufplanung im produktionswirtschaftlichen Kontext befasst sich als Teilgebiet der Produktionsplanung mit der „zeitlichen, mengenmäßigen und räumlichen Planung des Produktionsvollzugs“ (Schultmann 1998, S. 45). Produktionssysteme können anhand von zahlreichen Kriterien charakterisiert werden, die die Grundlage für die Auswahl geeigneter Modelle und Methoden für die Ablaufplanung bilden. Um eine Verbindung zur Projektplanung herzustellen, sind dabei insbesondere die Kriterien des Repetitionstyps, des Anordnungstyps und des Auftragsstyps relevant (Schultmann 1998, S. 46). Der *Repetitionstyp* beschreibt die Anzahl der von einem Produkt herzustellenden Gesamtmenge und wird in Einzel-, Serien- und Massenfertigung gegliedert. Der *Anordnungstyp* gibt Auskunft über die räumliche Anordnung der Produktionsschritte und wird in Fließ-, Werkstatt- und Baustellenfertigung unterteilt. Während bei der Fließfertigung eine fest vorgegebene Reihenfolge eingehalten werden muss, ist diese bei der Werkstattfertigung uneinheitlich. Bei der Baustellenfertigung ist das Produkt an einen Ort gebunden. Der *Auftragstyp* charakterisiert den Marktbezug der Produktion und unterscheidet zwischen Auftrags- und Vorratsfertigung.

Produktionssysteme, welche durch eine Einzelfertigung, eine Baustellenfertigung und eine Auftragsfertigung gekennzeichnet sind, entsprechen zu großen Teilen der in Abschnitt 2.1.1 vorgenommenen Definition eines Projekts. Die Produktionsplanung kann in diesem Fall mit Methoden und Modellen aus dem Bereich der Projektplanung durchgeführt werden. Dabei kann es auch ausreichend sein, wenn die drei Kriterien nur teilweise wie

beschrieben ausgeprägt sind. Inwiefern eine Verbindung zur Projektplanung hergestellt werden kann, hängt letztendlich von der Vereinbarkeit des verwendeten Modells mit den planerischen Rahmenbedingungen ab. Solange beispielsweise Umrüstzeiten für die Produktionsablaufplanung keine Rolle spielen und die Produktionsmengen vorbestimmt sind, ist das RCPSp als klassisches Problem der Projektablaufplanung meist gut dazu geeignet, auch den Produktionsablauf zu planen (siehe z. B. Schwindt und Trautmann 2000; Neumann, Schwindt und Trautmann 2005).

Die Übertragbarkeit von Methoden aus der Projektplanung auf die Produktionsplanung gilt unter den beschriebenen Voraussetzungen auch im umgekehrten Richtung, d. h. aus der Produktionsplanung in die Projektplanung. Da die Literatur nur in geringem Umfang Beiträge zum stoffstrombasierten Projektmanagement bereitstellt (siehe Abschnitt 3.5.2), werden bei der Entwicklung des Stoffstrommodells im nachfolgenden Kapitel Ansätze aus der stoffstrombasierten Produktionsplanung herangezogen und kombiniert, um Stoffströme in Projekten abzubilden. Diese Herangehensweise unterstützt beim Verständnis und der Einordnung der Modellierungskonzepte, die dem entwickelten Optimierungswerkzeug zugrunde liegen.

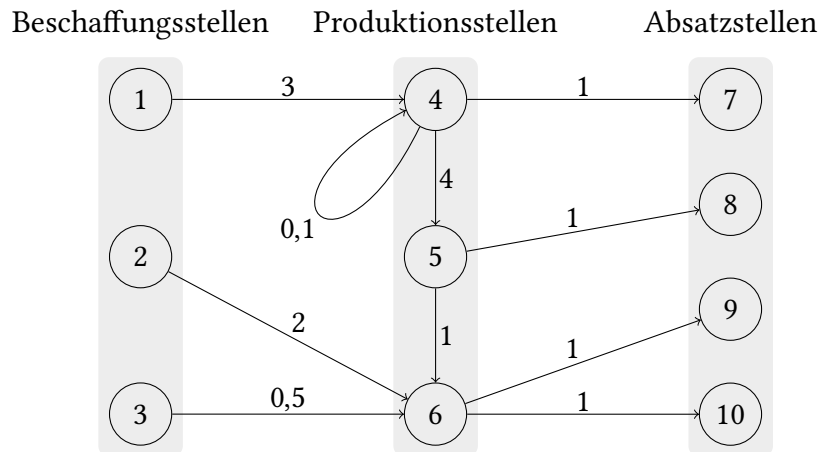
### 2.4.3 Graphentheoretische Modelle aus dem Produktionsmanagement

Nachfolgend werden gängige Modelle aus dem Produktionsmanagement auf ihre Eignung zur Abbildung von Stoffströmen untersucht. Es wird primär auf die qualitative Darstellung der Modelle eingegangen, da dies für ein Grundverständnis ausreichend ist. Für quantitative Details zu den Modellen wird auf weiterführende Literatur verwiesen.

#### 2.4.3.1 Input-Output-Graphen

Die *Input-Output-Analyse* ist eine von Kloock (1969) entwickelte Methodik, um Produktionsprozesse zu formulieren. Zur Darstellung der in den jeweiligen Produktionsstellen eingebrachten Gütereinsätze (Inputs) und der damit erzeugten Ausbringungsmenge (Outputs) werden *Input-Output-Graphen* verwendet. Dabei handelt es sich um bewertete Digraphen, wobei Knoten die Produktionsstellen darstellen und die Kanten deren Verflechtungen wiedergeben. Die Bewertung einer Kante von Knoten  $j$  zu Knoten  $j'$  gibt an, wie viele Mengeneinheiten eines Produkts aus Produktionsstelle  $j$  zur Produktion einer Mengeneinheit in Produktionsstelle  $j'$  benötigt werden. Abhängig von den Systemgrenzen können auch Beschaffungs- und Absatzstellen mit ihren jeweiligen Verflechtungen hinzugefügt werden. Die Verflechtungen in Input-Output-Graphen können beliebiger Art sein. Falls eine gesamte Kreislaufwirtschaft modelliert werden soll, treten insbesondere zyklische Strukturen auf (Schultmann 2003, S. 154). Ein Graph ohne isolierte Knoten (d. h. ohne Beschaffungs-, Produktions- oder Absatzstellen, die nicht mit dem restlichen Graphen verflochten sind) wird auch als *Netzwerk* bezeichnet. Abbildung 2.3 zeigt einen Input-Output-Graphen aus Fröhling (2006, S. 61), der um exemplarische Gewichtungen ergänzt wurde.

Falls eine Produktion mit Stoffströmen einhergeht (i. d. R. bei physischen Produkten), sind Input-Output-Graphen zur Abbildung der Struktur der Stoffströme geeignet. Bei Produktionsstellen handelt es sich dann um Prozesse, in denen Stoffe umgeformt oder umgewandelt werden. Beschaffungsstellen sind Quellen und Absatzstellen sind Senken der Stoffströme. Je nach Modellkontext müssen die Mengeneinheiten in Einheiten zur Beschreibung von Stoffströmen umgerechnet werden (z. B. Volumeneinheiten, Flächeneinheiten, Anzahl an Containern).

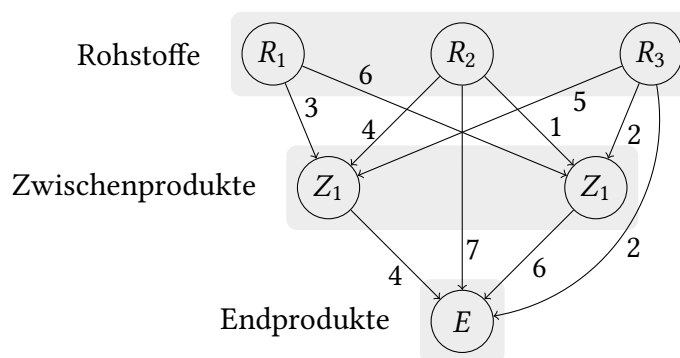


**Abbildung 2.3:** Beispiel für einen Input-Output-Graphen (nach Fröhling 2006, S. 61).

### 2.4.3.2 Gozintographen

*Gozintographen* dienen der Abbildung der Erzeugnisstruktur eines Produkts. Sie können auch als Veranschaulichung einer Stückliste definiert werden. Die Knoten eines Gozintographen stellen dabei die einzelnen Stufen eines Produkts dar und werden in Rohstoffe, Zwischen- und Endprodukte untergliedert (vgl. Bloech u. a. 2014, S. 155ff). Im Gegensatz zum Input-Output-Graphen steht somit nicht das Produktionssystem im Vordergrund, sondern lediglich die Zusammensetzung eines Produkts. Aus diesem Grund enthalten Gozintographen keine Schleifen und Zyklen. Abbildung 2.4 zeigt einen Gozintographen mit einem Endprodukt.

Wenn in einem Input-Output-Graphen ausschließlich die für die Erzeugnisstruktur eines Produkts relevanten Verflechtungen enthalten sind und davon ausgegangen wird, dass jede Produktionsstelle ein separates Zwischenprodukt erzeugt und jede Absatzstelle ein separates Endprodukt abnimmt, ist die Darstellung äquivalent zu einem Gozintographen. Wenn jedoch beispielsweise ein Zwischenprodukt bei seiner Herstellung verschiedene Produktionsstellen durchläuft, unterscheiden sich Input-Output-Graph und Gozintograph. Da Stoffströme nicht mit der Erzeugnisstruktur eines Produkts übereinstimmen müssen (z. B. wenn Reststoffe oder Abfälle anfallen), sind Gozintographen nicht zur Abbildung von Stoffströmen geeignet.



**Abbildung 2.4:** Beispiel für einen Gozintographen (nach Bloech u. a. 2014, S. 157).

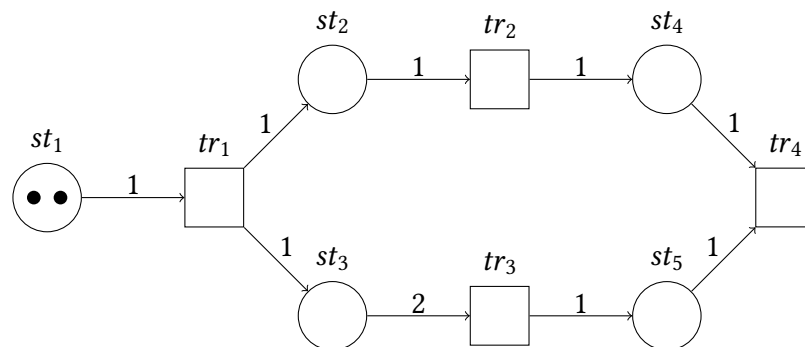
### 2.4.3.3 Petri-Netze

*Petri-Netze* sind eine auf Carl Adam Petri zurückgehende „Simulationsmethodik zur Abbildung gleichzeitig ablaufender Teilprozesse [...] [und] bilden damit eine zentrale modelltechnische Grundlage sowohl für die Darstellung von Stoffstromnetzen als auch für Fluss- und Bestandsrechnungen komplexer Stoffsysteme“ (Schultmann 1998, S. 214). Bei der Petri-Netz-Theorie handelt es sich um ein umfangreiches Forschungsfeld, welches in diesem Abschnitt nur in seinen Grundzügen angeschnitten wird. Der Fokus liegt hierbei auf den sogenannten *Stellen-Transitions-Netzen*, die zur Modellierung von Produktionssystemen geeignet sind. Stellen-Transitions-Netze bestehen aus zwei Typen von Knoten, den *Stellen* und *Transitions*. Die Stellen modellieren Zustände und bei Transitions handelt es sich um Ereignisse, die eine Zustandsveränderung herbeiführen. Stellen und Transitions sind mit gerichteten Kanten verbunden, die ihre Abhängigkeiten abbilden. Graphentheoretisch handelt es sich bei Petri-Netzen um sogenannte *bipartite Graphen*, da Stellen und Transitions immer nur mit dem jeweils gegenteiligen Knotentyp über eine Kante verbunden sein dürfen, d. h., sie müssen sich immer abwechseln. Es ist zulässig, durch Rückführungen Zyklen zu erzeugen. In der grafischen Darstellung werden Stellen als Kreise und Transitions als Rechtecke kenntlich gemacht. Stellen-Transitions-Netze dienen neben der Darstellung statischer Systeme auch dazu, das dynamische Verhalten eines Systems zu beschreiben. Dazu werden Stellen mit der Eigenschaft versehen, eine begrenzte oder unbegrenzte Anzahl von *Marken* aufzunehmen, die grafisch mittels Punkten abgebildet werden. Eine Transition kann *schalten*, indem sie eine gemäß ihrer Schaltregel vorgegebene Anzahl an Marken aus den ihr vorgeschalteten Stellen entnimmt und in den ihr nachgeschalteten Stellen ablegt. Im Graphen werden die Schaltregeln über die Kantenbewertungen spezifiziert. Für weiterführende Grundlagen zu Petri-Netzen wird auf Murata (1989) verwiesen.

Abbildung 2.5 zeigt ein Stellen-Transitions-Netz mit vier Transitions und fünf Stellen. Die Anfangsmarkierung ist ebenfalls eingezeichnet. Im abgebildeten Anfangszustand kann Transition  $tr_1$  schalten und legt jeweils eine Marke in Stelle  $st_2$  sowie in Stelle  $st_3$ . Transition  $tr_2$  kann danach direkt schalten. Transition  $tr_3$  kann erst schalten, nachdem Transition  $tr_1$  ein zweites Mal geschaltet hat, da gemäß der Bewertung von Kante  $(st_3, tr_3)$  zwei Marken für eine Schaltung benötigt werden. Nachdem sowohl Transition  $tr_2$  als



auch Transition  $tr_3$  mindestens einmal geschaltet haben, kann Transition  $tr_4$  schalten. Sie verbraucht beide dazu benötigten Marken aus den Stellen  $st_4$  und  $st_5$ .



**Abbildung 2.5:** Beispiel für ein Petri-Netz.

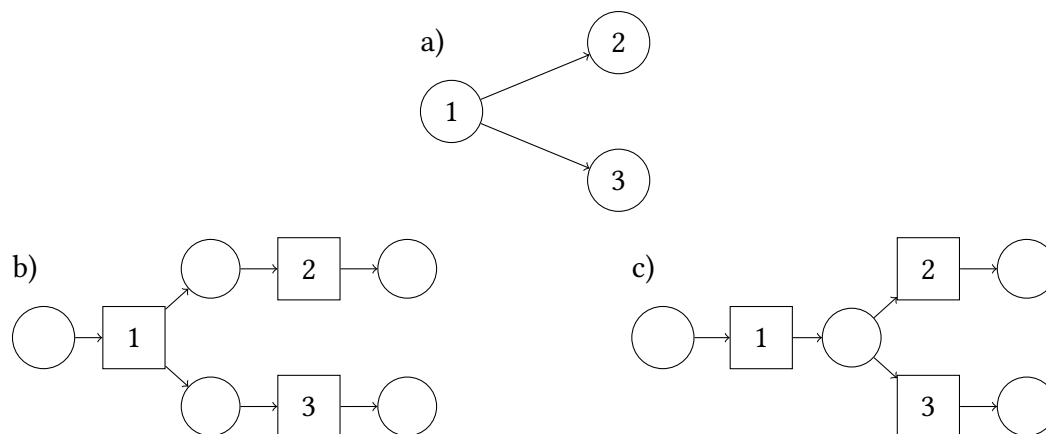
Transitionen symbolisieren im Kontext von Produktionssystemen die Stoffumwandlungs- bzw. -umformungsprozesse. Die in den Input-Output-Graphen als Kreise dargestellten Produktionsstellen sind deshalb nicht mit Stellen im Sinne des Petri-Netz-Formalismus zu verwechseln, sondern entsprechen vielmehr den Transitionen. Die Stellen im Petri-Netz ermöglichen hingegen die Modellierung von Beständen an Roh-, Hilfs- und Betriebsstoffen, Zwischen- und Endprodukten und Abfällen (vgl. Schultmann 2003, S. 215). Mit diesem Konzept erweitert das Modell der Petri-Netze die bisher vorgestellten graphentheoretischen Modelle wesentlich, sodass die Beschreibung des dynamischen Verhaltens von Stoffströmen ermöglicht wird. So erlauben es Petri-Netze beispielsweise im Gegensatz zu Input-Output-Graphen, eine Entscheidungssituation zwischen zwei oder mehr Transitionen abzubilden, indem jeder dieser Transitionen eine und dieselbe Stelle vorgeschaltet wird. Schultmann (2003, S. 214ff) demonstriert die Eignung von Petri-Netzen zur Abbildung von Stoffströmen am Beispiel der keramischen Industrie.

#### 2.4.3.4 State-Task-Netze

Bei *State-Task-Netzen* handelt es sich um ein Modell, welches von Kondili u. a. (1993) zur Abbildung von Stoffströmen in der Prozessindustrie vorgestellt wurde. Ihre Entwicklung motivieren sie mit Defiziten in anderen Darstellungsformen, die die Gefahr doppeldeutiger Interpretationsmöglichkeiten bergen. So kritisieren die Autoren, dass aus einer Verzweigung oft nicht ersichtlich wird, ob die nachfolgenden Prozesse mit verschiedenen- oder gleichartigen Stoffströmen versorgt werden (Kondili u. a. 1993, S. 213f). State-Task-Netzwerke hingegen drücken eindeutig aus, um welchen Fall es sich hierbei handelt. Sie bestehen aus *Zuständen* (engl. states) und *Aufgaben* (engl. tasks), die über gerichtete Kanten abwechselnd miteinander verbunden sind. Die Zustände werden als Kreise und die Aufgaben als Rechtecke abgebildet, d. h., es handelt sich um einen bipartiten Graphen. Hier wird ersichtlich, dass State-Task-Netze grafisch identisch mit Stellen-Transitions-Netzen sind und sich die Definitionen von Zuständen/Stellen und Aufgaben/Transitionen stark ähneln. Es ist bemerkenswert, dass Kondili u. a. (1993) sich in ihrer Arbeit nicht explizit

vom Petri-Netz-Konzept abgrenzen. Aus ihrer Modellspezifikation lässt sich indes ableiten, dass es sich um ein gegenüber den Petri-Netzen deutlich reduzierten Modellumfang handelt. State-Task-Netze dienen lediglich der Abbildung eines statischen Produktionssystems und können als eine Erweiterung von Input-Output-Graphen verstanden werden, die eine Präzisierung der Stoffarten ermöglicht, die zwischen verschiedenen Produktionsstellen fließen. Jeder Zustand entspricht per Definition einer bestimmten Stoffart. Somit müssen auch sämtliche Stoffe, die in einen Zustand einfließen, derselben Art entsprechen.

Bei Abbildung 2.6a handelt es sich um einen Input-Output-Graphen mit drei Produktionsstellen und einer Verzweigung nach Produktionsstelle 1. Es ist nicht ersichtlich, ob Produktionsstellen 2 und 3 mit verschiedenartigen oder gleichartigen Stoffströmen versorgt werden. Abbildung 2.6b zeigt das entsprechende State-Task-Netz, falls Produktionsstellen 2 und 3 mit verschiedenartigen Stoffströmen versorgt werden. Abbildung 2.6c hingegen zeigt das State-Task-Netzwerk für den Fall eines gleichartigen Stoffstroms. Es ist zu beachten, dass Produktionsstellen in State-Task-Netzwerken den Aufgaben entsprechen und somit als Rechtecke abgebildet werden.



**Abbildung 2.6:** Darstellung a) eines Input-Output-Graphen zur Versorgung der Produktionsstellen 2 und 3 als b) State-Task-Netz mit verschiedenartigen Stoffströmen und als c) State-Task-Netz mit gleichartigen Stoffströmen (nach Kondili u. a. 1993, S. 213f).

#### 2.4.4 Vergenzen von Produktionsstrukturen und Stoffströmen

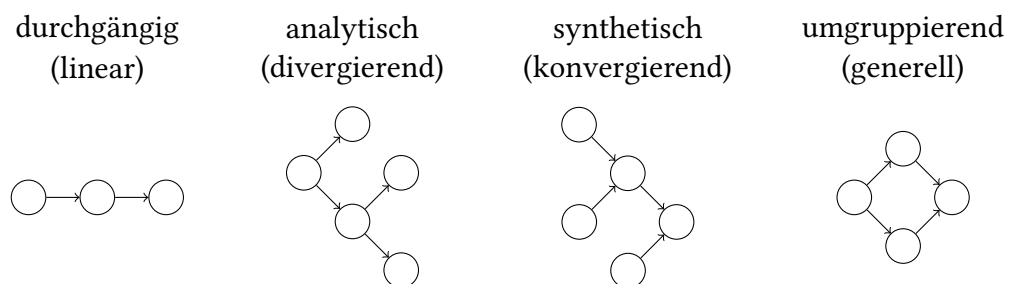
Produktionsstrukturen können anhand sogenannter *Vergenzen* unterschieden werden, die beschreiben, inwiefern Güter beim Durchlaufen von Produktionsprozessen miteinander vereint oder aufgespalten werden. Unter welchen Voraussetzungen Güter als vereint bzw. aufgespalten gelten, ist je nach Modellkontext und technologischen Rahmenbedingungen zu spezifizieren.

Auch die mit einer Produktionsstruktur einhergehenden Stoffströme lassen sich anhand ihrer Vergenz unterscheiden, die direkt vom Vergenztyp der Produktionsstruktur abgeleitet

wird. Da Input-Output-Graphen zur Abbildung der Struktur von Stoffströmen geeignet sind, werden sie im Folgenden herangezogen, um Vergenztypen graphisch zu illustrieren (vgl. Abbildung 2.7). Der unten beschriebene Zusammenhang gilt jedoch auch für die anderen vorgestellten graphentheoretischen Modelle, d. h. an den Graphen ist jeweils zu erkennen, um welchen Vergenztyp es sich handelt.

Nach Fröhling (2006, S. 10f) können folgende Vergenztypen für Produktionsstrukturen und die damit einhergehenden Stoffströme unterschieden werden:

- Eine *durchgängige* (oder glatte) Produktionsstruktur ist dadurch gekennzeichnet, dass jeder Produktionsprozess aus einem Input einen Output erzeugt. Die dadurch hervorgerufenen Stoffströme werden als *lineare Stoffströme* bezeichnet und sind graphisch an linearen Input-Output-Graphen zu erkennen.
- Eine *analytische* Produktionsstruktur ist dadurch gekennzeichnet, dass aus einem Input mehrere Outputs erzeugt werden. Die dadurch hervorgerufenen Stoffströme werden als *divergierende Stoffströme* bezeichnet und sind graphisch an Input-Output-Graphen in Form eines Out-Trees zu erkennen. Divergierende Stoffströme umfassen auch lineare Stoffströme als Spezialfall.
- Eine *synthetische* Produktionsstruktur ist dadurch gekennzeichnet, dass aus mehreren Inputs ein Output erzeugt wird. Die dadurch hervorgerufenen Stoffströme werden als *konvergierende Stoffströme* bezeichnet und sind graphisch an Input-Output-Graphen in Form eines In-Trees zu erkennen. Konvergierende Stoffströme umfassen auch lineare Stoffströme als Spezialfall.
- Eine *umgruppierende* (oder austauschende) Produktionsstruktur ist dadurch gekennzeichnet, dass in jeder Produktionsstelle aus beliebig vielen Inputs beliebig viele Outputs erzeugt werden können. Die dadurch hervorgerufenen Stoffströme werden als *generelle Stoffströme* bezeichnet (in Anlehnung an Neumann und Schwindt 1997). Generelle Stoffströme umfassen auch lineare, divergierende und konvergierende Stoffströme als Spezialfälle.



**Abbildung 2.7:** Verschiedene Input-Output-Graphen mit Angabe des Vergenztyps der abgebildeten Produktionsstruktur (und der damit einhergehenden Stoffströme).



## 3 Stoffstrombasierte Planung von Rückbauprojekten

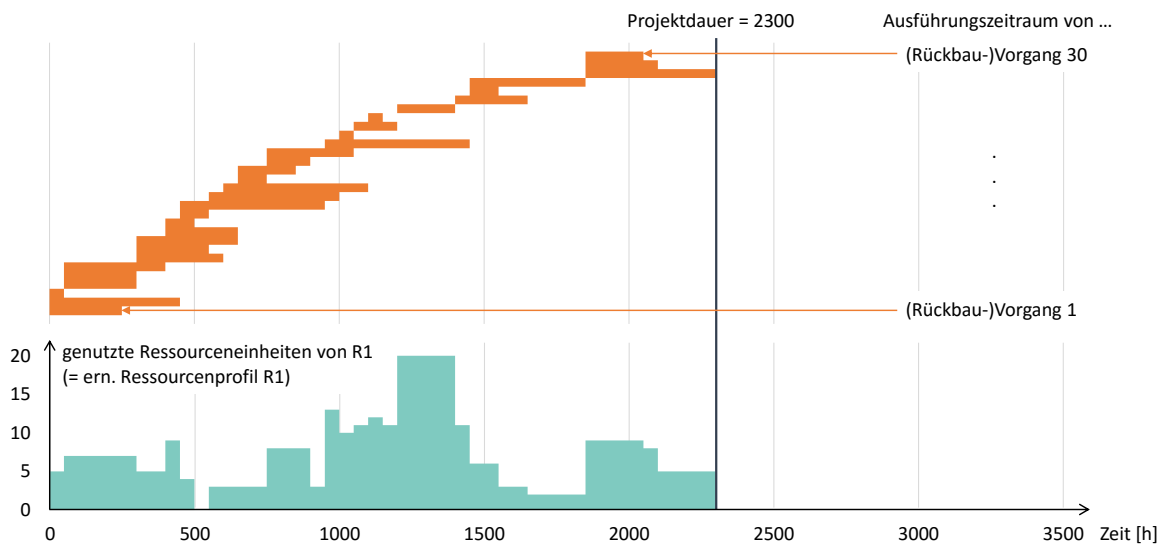
Dieses Kapitel fokussiert sich auf die konzeptionelle Formulierung des mathematischen Problems, welches den Kern des in dieser Arbeit entwickelten stoffstrombasierten Optimierungswerkzeugs bildet. Dazu wird in Abschnitt 3.1 die betrachtete Problemstellung skizziert und unter Verwendung verschiedener Gantt-Diagramme illustriert. Um die stoffstrombasierte Rückbauplanung zu motivieren, wird in Abschnitt 3.2 der Rückbau kerntechnischer Anlagen als Anwendungsbeispiel angeführt. Kernkraftwerke eignen sich in besonderer Weise als Anwendungsumgebung für das in dieser Arbeit entwickelte Optimierungswerkzeug, da sie hochkomplex und stoffstromintensiv sind. Es wird ausführlich auf technische und gesetzliche Rahmenbedingungen eingegangen, die bei der Verarbeitung der in kerntechnischen Rückbauprojekten anfallenden Stoffströme zu beachten sind.

Obwohl an verschiedenen weiteren Stellen auf den kerntechnischen Rückbau als Anwendungsbeispiel zurückgegriffen wird, erfolgt die weitere Darlegung aus anwendungsübergreifender Perspektive. In Abschnitt 3.3 werden die Modellanforderungen, -einschränkungen und -abgrenzungen definiert und begründet. Damit sind die Voraussetzungen geschaffen, um in Abschnitt 3.4 das mathematische Modell und Problem zu spezifizieren. Anschließend zieht Abschnitt 3.5 einen Vergleich zu existierenden Ansätzen in der Literatur.

### 3.1 Problemstellung

Begrenzte Ressourcenkapazitäten können zu Verzögerungen im Projektablauf führen. Im vorangegangenen Kapitel wurden geeignete Schedulingmodelle vorgestellt, um derartige Verzögerungen in die Planung zu integrieren. So können die erneuerbaren Ressourcenrestriktionen im (RCPS) erzwingen, dass Vorgänge nicht zeitgleich ausgeführt werden dürfen, obwohl es ihre Zeitstruktur zulassen würde. Abbildungen 3.1 und 3.2 verdeutlichen exemplarisch, wie sich derartige Ressourcenknappheiten auf den Projektablauf auswirken. Abbildung 3.1 zeigt das Gantt-Diagramm eines Schedules, zu dessen Ausführung unbegrenzte erneuerbare Ressourcenkapazitäten zur Verfügung stehen (d. h.  $R_k^\alpha = \infty$  für alle  $k \in \mathcal{R}^\alpha$ ). Damit reduziert sich das (RCPS) auf ein (TPSP). Wenn angenommen wird, dass die Projektdauer minimiert werden soll, zeigt Abbildung 3.1 die optimale Lösung mit einer Projektdauer von 2300 Stunden. Unterhalb des Gantt-Diagramms ist das Ressourcenprofil der exemplarisch ausgewählten erneuerbaren Ressource R1 dargestellt. Abbildung 3.2 zeigt einen Schedule für dasselbe Projekt, wobei hier nur begrenzte erneuer-

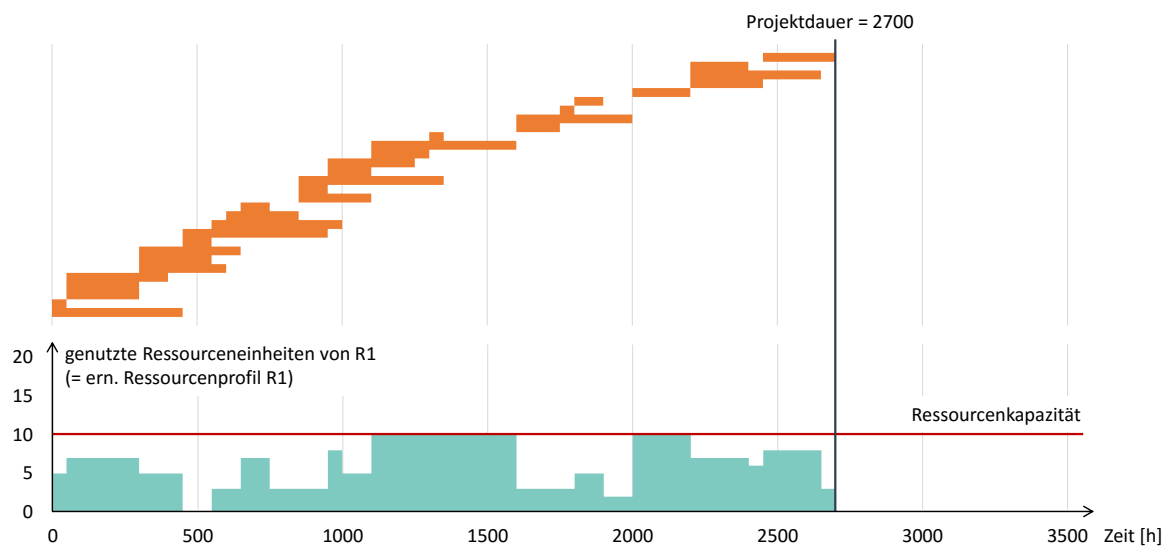
bare Ressourcenkapazitäten zur Verfügung stehen. Dies ist durch die horizontale Linie im Ressourcenprofil von R1 kenntlich gemacht. Im Vergleich mit Abbildung 3.1 ist erkennbar, dass die Ressourcenrestriktionen das betrachtete Projekt um 400 Stunden verzögern. Bei dem in Abbildung 3.2 gezeigten Schedule handelt es sich um eine Lösung des (RCPSP).



**Abbildung 3.1:** Gantt-Diagramm und erneuerbares Ressourcenprofil zu einem Projekt mit unbegrenzten erneuerbaren Ressourcenkapazitäten; Lösung des (TPSP).

Schedulingmodelle können der kurzfristigen (operativen) Planungsebene zugeordnet werden (Hübner 2019, S. 139 und S. 227ff; Schultmann 1998, S. 45), weil sie für jeden Vorgang einen individuellen Startzeitpunkt berechnen. Je nach betrachteter Zielsetzung, gewähltem Zeithorizont und vorliegendem Detaillierungsgrad des Projektstrukturplans kann allerdings auch die langfristige (strategische) oder die mittelfristige (taktische) Planungsebene in den Vordergrund rücken (Zimmermann u. a. 2006, S. 37f). Schultmann (1998, S. 45) stellt diesbezüglich fest, dass sich die Planungsebenen nicht klar voneinander abgrenzen lassen. So wurde beim Vergleich der Schedules in den Abbildungen 3.1 und 3.2 auf einer überwiegend strategischen Ebene eingeschätzt, inwiefern sich die begrenzten Ressourcenkapazitäten auf die Projektdauer auswirken. Schedulingmodelle sind somit auch dazu geeignet, eine Gesamteinschätzung über wichtige Kennzahlen, wie die Projektdauer, abzugeben. Gleichzeitig liefern die berechneten Lösungen detaillierte Informationen zur operativen Projektdurchführung. Durch Anpassung des Detaillierungsgrads der Eingabeparameter haben Projektverantwortliche die Möglichkeit, das Schedulingproblem stärker in Richtung der operativen oder der strategischen Planungsebene auszurichten. Die in dieser Arbeit betrachteten Rückbauprojekte werden, wie nachfolgend hergeleitet, überwiegend aus einer strategischen Perspektive betrachtet.

Bei Rückbauprojekten können neben den Verzögerungen aufgrund begrenzter Ressourcenkapazitäten auch Verzögerungen durch die freigesetzten Stoffströme auftreten, da die Verarbeitungskapazitäten für die Stoffe begrenzt sind. Diese Verzögerungen machen sich nicht wie bei den erneuerbaren Ressourcen unmittelbar im Rückbau bemerkbar. Erst wenn die Lagerkapazitäten erschöpft sind, tritt ein Rückkopplungseffekt auf, der



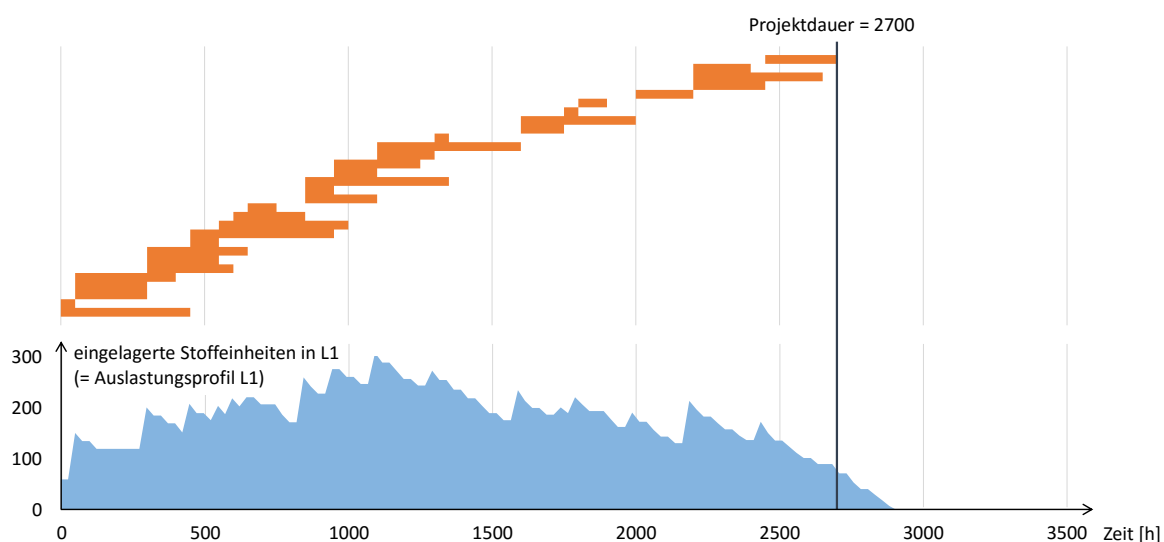
**Abbildung 3.2:** Gantt-Diagramm und erneuerbares Ressourcenprofil zu einem Projekt mit begrenzten erneuerbaren Ressourcenkapazitäten; Lösung des (RCPSP).

die Unterbrechung des weiteren Rückbaus erzwingt. Ohne Unterbrechung stünden keine weiteren Möglichkeiten zur Pufferung der freigesetzten Stoffe zur Verfügung. Zur Verdeutlichung dieses Rückkopplungseffekts wird von zwei Planungsbereichen gesprochen: Im *Rückbaubereich* ist ein klassisches, ressourcenbeschränktes Projekt zu planen. Die Vorgänge im Rückbaubereich werden als *Rückbauvorgänge* bezeichnet. Bei ihrer Ausführung werden Stoffe freigesetzt, die im *Stoffstrombereich* unter Berücksichtigung begrenzter Verarbeitungs- und Lagerkapazitäten verarbeitet werden müssen. Im Zusammenspiel bilden die begrenzten Verarbeitungs- und Lagerkapazitäten Engpässe bei der Stoffverarbeitung. Es ist Rückbauprojekten inhärent, dass der Stoffstrombereich dem Rückbaubereich nachgeschaltet ist.

Die in dieser Arbeit betrachtete Problemstellung ist *primär* über die Frage beschrieben, welche Auswirkungen die Engpässe im Stoffstrombereich auf die Zielsetzung und den Schedule im Rückbaubereich haben. Die Antwort auf diese Frage umfasst Informationen zum Zielfunktionswert und zu den Startzeiten der Rückbauvorgänge. Es liegt hier überwiegend ein Fokus auf die strategische Planungsebene vor. Dennoch soll das Modell auch eine operative Planung mit hohem Detaillierungsgrad ermöglichen. *Sekundär* ist die Problemstellung demnach auch über die Frage beschrieben, wie der zeitliche Ablauf der Stoffverarbeitung aussieht, sodass der Stoffstrombereich nicht überlastet und die Restriktionen im Rückbaubereich eingehalten werden. Die Antwort auf diese sekundäre Frage umfasst Informationen zur zeitabhängigen Ressourcen- und Lagerauslastung, sowie zur Verarbeitungsreihenfolge im Stoffstrombereich. Indem in Kapitel 4 auf existierenden vorgangsbasierten Schedulingmodellen aufgebaut wird, zeigt sich, dass die Beantwortung der primären Frage mit der Beantwortung der sekundären Frage einhergeht. Die vorgenommene Abstufung in Form einer primären und sekundären Frage ist jedoch ausschlaggebend dafür, wie das mathematische Problem in diesem Kapitel entwickelt wird. Vereinfacht ausgedrückt, soll das Optimierungswerkzeug sich nicht mit der Reihenfolge- und Maschi-

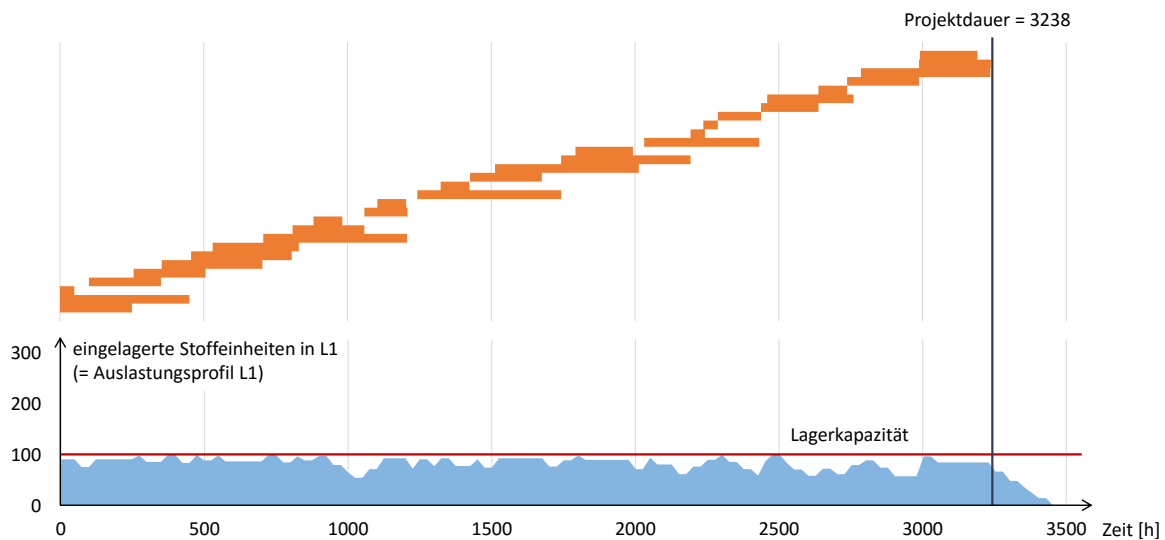
nenbelegungsplanung bei der Stoffverarbeitung, sondern mit der Ablaufplanung eines Rückbauprojekts befassen. Da zur Ablaufplanung des Rückbauprojekts der Stoffstrombereich jedoch auf geeignete Weise modelliert werden muss, wird die Reihenfolge- und Maschinenbelegungsplanung des Stoffstrombereichs als eine Art „Nebeneffekt“ ebenfalls zu einem gewissen Grad durchgeführt. Es sei darauf hingewiesen, dass die primäre und sekundäre Frage, welche die Problemstellung charakterisieren, nicht mit der Zielsetzung im Sinne des Optimierungsproblems verwechselt werden dürfen. Das heißt, es wird keine multikriterielle Optimierung durchgeführt.

Abbildungen 3.3 und 3.4 verdeutlichen im Bezug auf die primäre Fragestellung exemplarisch, wie sich Engpässe im Stoffstrombereich auf den Projektablauf im Rückbaubereich auswirken. Es wird dasselbe Projekt wie in den Abbildungen 3.1 und 3.2 betrachtet. Abbildung 3.3 zeigt das Gantt-Diagramm eines Schedules, zu dessen Ausführung unbegrenzte Lagerkapazitäten zur Verfügung stehen. Die erneuerbaren Ressourcenkapazitäten sind gegenüber Abbildung 3.2 unverändert, weshalb beide Gantt-Diagramme übereinstimmen und es sich um eine Lösung des (RCPSP) handelt. Unterhalb des Gantt-Diagramms ist das Auslastungsprofil des exemplarisch ausgewählten Lagers L1 dargestellt. Abbildung 3.4 zeigt einen Schedule für dasselbe Projekt, wobei hier nur begrenzte Lagerkapazitäten zur Verfügung stehen. Dies ist durch die horizontale Linie im Auslastungsprofil von L1 kenntlich gemacht. Die Verarbeitungskapazitäten sind gegenüber Abbildung 3.3 unverändert. Im Vergleich ist erkennbar, dass die Lagerengpässe das betrachtete Projekt um 538 Stunden verzögern. Der in Abbildung 3.4 gezeigte Schedule ist eine Lösung des in diesem Kapitel eingeführten *Schedulingproblems unter Ressourcen- und Stoffstromrestriktionen (RMCPSP)* (engl. resource- and material-flow-constrained project scheduling problem). Da in diesem Schedule Informationen zur zeitabhängigen Ressourcen- und Lagerauslastung im Stoffstrombereich enthalten sind, wird auch die sekundäre Fragestellung beantwortet.



**Abbildung 3.3:** Gantt-Diagramm und Auslastungsprofil zu einem Projekt mit unbegrenzten Lagerkapazitäten; Lösung des (RCPSP).





**Abbildung 3.4:** Gantt-Diagramm und Auslastungsprofil zu einem Projekt mit begrenzten Lagerkapazitäten; Lösung des in diesem Kapitel vorgestellten (RMCPSP).

Sowohl in Abbildung 3.3 als auch in Abbildung 3.4 ist ersichtlich, dass die Entleerung der Lager und damit die Stoffverarbeitung nach dem mit der vertikalen Linie eingezeichneten Projektende fortgesetzt wird. Dies verdeutlicht, dass die Stoffverarbeitung nicht zum Rückbauprojekt gezählt wird, da sie lediglich eine sekundäre Stellung hat.

## 3.2 Der Rückbau kerntechnischer Anlagen als Anwendungsbeispiel

In Europa wird der Rückbau kerntechnischer Anlagen (im Folgenden als „kerntechnischer Rückbau“ bezeichnet) in den kommenden Jahren zunehmend an Bedeutung gewinnen, da die vorhandenen Kernkraftwerke das Ende ihrer Lebensdauer erreichen (Europäische Kommission 2016, Abschnitt 5.2). Volk u. a. (2019, S. 240) stellen fest, dass dies auch weltweit gilt – insbesondere in den USA und in Japan ist bis zum Jahr 2047 mit einer umfangreichen Reduktion der mit Kernkraft erzeugten elektrischen Leistung zu rechnen. Die Europäische Kommission (2016, Abbildung 27) schätzt die durchschnittlichen Kosten auf 0,3 bis 1,3 Milliarden Euro pro Rückbau einer kerntechnischer Anlage, wobei diese Schätzungen aufgrund fehlender Erfahrungen auf diesem Gebiet erheblichen Unsicherheiten unterliegen. Schätzungen zur Dauer eines Rückbaus variieren ebenfalls beträchtlich und sind nur in geringem Umfang zu finden. Beispielsweise wurde für den Rückbau des Kernkraftwerks Greifswald mit einem Zeitraum von 18 Jahren gerechnet (Thierfeldt und Schartmann 2009, S. 54). Die E.ON Kernkraft GmbH (2008, S. 19) schätzte die Rückbaudauer für das Kernkraftwerk Stade im Jahr 2008 auf zehn Jahre ein, sodass eine Fertigstellung bis 2015 anvisiert wurde. Neueren Angaben des Betreibers zufolge werde mit einer Fertigstellung bis 2026 gerechnet (Preussen Elektra 2020).

Der kerntechnische Rückbau weist aufgrund der Einzigartigkeit seiner Rahmenbedingungen einen Projektcharakter auf. So unterscheiden sich die Anlagen in vielen Fällen bereits in ihrer Bauweise. Darüber hinaus ist der individuell vorherrschende radiologische Ausgangszustand zu berücksichtigen. Mit gewöhnlichen Rückbauprojekten sind kerntechnische Rückbauprojekte aufgrund der hohen Sicherheitsanforderungen sowie ihrer Komplexität und Größe nur bedingt vergleichbar (Hübner 2019, S. 11). An dieser Stelle wird nicht weiter auf die allgemeinen Rahmenbedingungen des kerntechnischen Rückbaus eingegangen, sondern auf eine umfassende Darlegung in Hübner (2019, S. 13ff) verwiesen. Darin wird auch diskutiert, welche verschiedenen Stilllegungsstrategien für kerntechnische Anlagen unterschieden werden und wie sich der Rückbau in den gesamten Stilllegungsprozess einfügt.

#### 3.2.1 Freisetzung von Stoffströmen

Beim Rückbau kerntechnischer Anlagen werden Stoffe freigesetzt. Das Atomgesetz unterscheidet zwischen radioaktiven Reststoffen und radioaktiven Abfällen, wobei Reststoffe „schadlos verwertet“ und Abfälle „geordnet beseitigt“ werden müssen (§9a Abs. 1 AtG). Die EnBW Kernkraft GmbH (2018, S. 24) betrachtet radioaktiven Abfall als Teil der radioaktiven Reststoffe und definiert Letztere als „anfallende Stoffe, bewegliche Gegenstände, Anlagen und Anlagenteile [...], die kontaminiert oder aktiviert sind oder bei denen eine Kontamination oder Aktivierung nicht auszuschließen ist“. Daneben fallen auch nichtradioaktive Reststoffe an, d. h. Stoffe, „die weder kontaminiert noch aktiviert sind“ (EnBW Kernkraft GmbH 2018, S. 25). Während die Kontamination weitgehend eine Oberflächeneigenschaft ist, die durch Dekontaminationsmaßnahmen entfernt werden kann, ist die Aktivierung eine Volumeneigenschaft, die sich gewöhnlich nicht entfernen lässt (Thierfeldt und Schartmann 2009, S. 77). Abhängig von ihrer Entstehungsart werden anfallende Stoffe auch in Primär- und Sekundärreststoffe bzw. -abfälle unterteilt. Während Primärreststoffe aus der Anlage selbst stammen (z. B. Anlagenteile oder Bauschutt), entstehen Sekundärreststoffe erst im Zuge der Stilllegungsverfahren (z. B. Dekontaminationsflüssigkeiten oder verbrauchte Werkzeuge) (Thierfeldt und Schartmann 2009, S. 75). In dieser Arbeit umfasst der Begriff *Stoffe* im Kontext des kerntechnischen Rückbaus alle Arten anfallender radioaktiver oder nichtradioaktiver Primär- oder Sekundärreststoffe oder -abfälle.

Zur Bezifferung der Mengen freigesetzter Stoffe wird auch von *Massen* gesprochen. So benennt die EnBW Kernkraft GmbH (2018, S. 25) die Gesamtmasse des sich in der Stilllegung befindlichen Kernkraftwerks Philippsburg, Block 2, mit ca. 782 500 Megagramm (Mg; 1 Mg = 1 Tonne). Beim Rückbau des Kernkraftwerks Greifswald wird die Gesamtmasse aufgrund der materialintensiven Bauweise mit 1 800 000 Mg angegeben (Thierfeldt und Schartmann 2009, S. 44). In einer Studie zum Rückbau des italienischen Kernkraftwerks Trino wurde die Gesamtmasse der anfallenden Stoffe auf ca. 18 810 Mg geschätzt (Brusa u. a. 2002, S. 14). Es handelt sich im Vergleich zu den Kernkraftwerken Philippsburg und Greifswald um eine deutlich kleinere Anlage. Hieran ist ersichtlich, wie stark sich die jeweiligen Rückbauprojekte voneinander unterscheiden können.

Allen Projekten gemein ist in der Regel das Ziel, möglichst viele Stoffe in den Stoffkreislauf zurückzuführen (Brusa u. a. 2002, S. 1; Hübner 2019, S. 128). Dies ist „ein ökologisch und ökonomisch vordringliches Gebot“, da die Zwischen- und Endlagerkapazitäten knapp und teuer sind (Thierfeldt und Schartmann 2009, S. 44). Laut BMUB (2021, S. 24) sollte „zur Reduzierung des Abfallvolumens [...] bereits bei der Entstehung von Reststoffen eine Trennung in verwertbare Stoffe und radioaktive Abfälle erfolgen“. Beim Umgang mit Stoffen, die der atom- und strahlenschutzrechtlichen Überwachung unterliegen, d. h. aus kerntechnischen Anlagen stammen, sind zwei Fälle zu unterscheiden:

- Stoffe, „bei denen aufgrund der Betriebshistorie und aufgrund der Nutzung eine Kontamination oder Aktivierung ausgeschlossen ist“: Sie können, falls sie nicht aus dem Kontrollbereich stammen, *herausgegeben* werden (BMUB 2021, S. 23). „Geeignete Messungen“ müssen eine Kontaminierungs- und Aktivierungsfreiheit sicherstellen (BMUB 2021, S. 23).
- Stoffe, die von den §§ 31 bis 42 der Strahlenschutzverordnung (StrlSchV, Fassung vom 29.11.2018) erfasst sind: Sie können *freigegeben* werden, falls die Einhaltung des im § 31 StrlSchV vorgeschriebenen Dosiskriteriums durch sogenannte *Freimessungen* nachgewiesen wird (BMUB 2021, S. 21). Bei der Freigabe unterscheidet § 32 StrlSchV zwischen einer uneingeschränkten Freigabe, die „keiner Festlegungen zur künftigen Verwendung, Verwertung, Beseitigung, des Innehabens der freizugebenden Stoffe und Gegenstände oder deren Weitergabe an Dritte“ bedarf, und der spezifischen Freigabe, bei der die „künftige Verwendung, Verwertung, Beseitigung, das Innehaben der freizugebenden Stoffe und Gegenstände oder deren Weitergabe an Dritte eingeschränkt“ ist. Unter einer solchen Einschränkung wird dabei, wie § 36 StrlSchV konkretisiert, beispielsweise die Beseitigung auf Deponien oder in einer Verbrennungsanlage verstanden. Falls eine Freigabe nicht möglich ist, gelten die gleichen Randbedingungen wie für Betriebsabfälle (BMUB 2021, S. 24). Stoffe können innerhalb des kerntechnischen Stoffkreislaufs wiederverwertet oder -verwendet werden oder müssen, sofern es keine „objektiv[en] Verwertungsmöglichkeiten“ (BMUB 2021, S. 24) gibt, als radioaktive Abfälle angesehen werden (EnBW Kernkraft GmbH 2018, S. 25; Thierfeldt und Schartmann 2009, S. 108).

Die genannten Optionen zum Umgang mit Stoffen, d. h. die Herausgabe, die uneingeschränkte Freigabe, die spezifische Freigabe, die Wiederverwertung im kerntechnischen Stoffkreislauf und die Beseitigung als radioaktiver Abfall, werden in Anlehnung an die EnBW Kernkraft GmbH (2018, S. 25) als *Entsorgungsziele* bezeichnet. Von der 782 500 Megagramm umfassenden Gesamtmasse des Kernkraftwerks Philippsburg, Block 2, besteht der überwiegende Teil aus Gebäuden und baulichen Anlageteilen, die laut EnBW Kernkraft GmbH (2018, S. 25) herausgegeben werden sollen. Den radioaktiven Reststoffen seien ca. 15 600 Mg zuzuordnen und es werde mit Sekundärreststoffen von ca. 350 Mg gerechnet, d. h., 15 950 Mg Stoffe sind von den §§ 31 bis 42 StrlSchV erfasst. Davon sollen 11 600 Mg freigegeben werden, 500 Mg dem kerntechnischen Stoffkreislauf zugeführt werden und 4200 Mg als radioaktiver Abfall beseitigt werden (EnBW Kernkraft GmbH 2018, S. 25). Es ist nicht ersichtlich, weshalb die Angaben zur Entstehung und zur Verwendung der Stoffe in Summe nicht übereinstimmen. Dennoch lässt sich feststellen, dass der Anteil

radioaktiver Massen an der Gesamtmasse im unteren einstelligen Prozentbereich liegt (Thierfeldt und Schartmann 2009, S. 79).

#### 3.2.2 Verarbeitung von Stoffströmen

Nach Thierfeldt und Schartmann (2009, S. 104ff) unterteilt sich der Ablauf des Freigabeverfahrens in eine Voruntersuchung zur radiologischen Charakterisierung des Materials, einer Durchführung weiterer Maßnahmen zur Herstellung der Freigabefähigkeit (d. h. insbesondere Dekontaminationsmaßnahmen) und einer Entscheidungsmessung zum Nachweis, ob die Freigabewerte unterschritten werden. Für die einzelnen Schritte kommen zahlreiche Verfahren zum Einsatz. In der nachfolgenden Auflistung werden einige Verfahren genannt, wobei jedoch kein Anspruch auf Vollständigkeit erhoben wird. Die Darlegungen basieren auf Thierfeldt und Schartmann (2009).

- Wenn das Material nicht in der geeigneten Größe vorliegt, ist eine Nachzerlegung erforderlich, beispielsweise bei Rohren oder größeren Komponenten. Wenn innere Oberflächen nicht zugänglich sind, werden Bauteile aufgeschnitten. Es wird zwischen thermischen (z. B. autogenes Brennschneiden, Plasmaschmelzschnitten, Laserschneiden, etc.) und mechanischen (z. B. Sägeverfahren, Trennschneiden, Trennschleifen, Wasserabrasivstrahlschneiden, etc.) *Zerlegeverfahren* unterschieden. Eine ausführliche Auflistung und Beschreibung von Zerlegeverfahren ist in Thierfeldt und Schartmann (2009, S. 129ff) zu finden.
- Zur Reduktion von Kontaminationen in Kreisläufen und auf Oberflächen werden *Dekontaminationsverfahren* eingesetzt. Es wird zwischen chemischen (z. B. Wasser, Säuren, etc.) und mechanischen (z. B. Bürsten, Saugen, Abraspeln, Schaben, Abfräsen, Sandstrahlen, etc.) Dekontaminationsverfahren unterschieden. Auch das Beschichten betroffener Flächen kommt infrage. Weitere Verfahren, wie z. B. das Elektropolieren oder die Laserablation können weder den chemischen noch den mechanischen Verfahren zugeordnet werden. Eine ausführliche Auflistung und Beschreibung von Dekontaminationsverfahren ist in Thierfeldt und Schartmann (2009, S. 139ff) zu finden.
- Die Auswahl geeigneter *Messverfahren* ist unter anderem von der Art der nachzuweisenden Kontamination, ihrer räumlichen Verteilung und der Materialart sowie -geometrie abhängig. Es stehen beispielsweise Kontaminationsmessgeräte für Messungen an Oberflächen und spektrometrische Verfahren basierend auf Probenentnahmen zur Verfügung. Freimessanlagen sind dazu geeignet, komplette Komponenten mit festgelegter Geometrie innerhalb einer Messkammer zu untersuchen. Sie sind empfindlich und erreichen hohe Durchsätze. Für große Flächen bieten sich spezielle Detektoren an, die in gewissem Abstand auf die zu messenden Flächen gerichtet werden können. Eine ausführliche Auflistung und Beschreibung von Messverfahren ist in Thierfeldt und Schartmann (2009, S. 142ff) zu finden.

- Wenn Material nicht freigegeben werden kann, sind *Konditionierungsverfahren* vorgesehen. Diese Verfahren zielen darauf ab, die Endlagerbedingungen zu erfüllen und somit einen langfristig sicheren Einschluss der Radioaktivität zu erreichen. Im Einzelnen sind Konditionierungsverfahren auf „die Volumenreduktion, die Trocknung, die sichere Einbettung und Fixierung der Abfälle in Gebinden und die Herstellung sowie [den] Verschluss der Gebinde selbst“ (Thierfeldt und Schartmann 2009, S. 144) ausgerichtet. Zur Volumenreduktion werden Abfälle beispielsweise gepresst. Zur sicheren Einbettung werden Hohlräume in Containern oder Fässern mit metallischen Reststoffen vergossen.

Zwischen dem Einsatz einzelner Verfahren, wie sie in der Auflistung genannt wurden, befinden sich Transport-, Verpackungs-, Sammel-, Sortier-, Übergabe- und Ausschleusevorgänge (EnBW Kernkraft GmbH 2018, S. 17f; EnBW Kernkraft GmbH o. D. S. 5). Wenn eine Dekontamination nicht wirtschaftlich ist und die Freigabewerte nicht deutlich überschritten werden, kommt möglicherweise eine Abklinglagerung infrage (Thierfeldt und Schartmann 2009, S. 107). Aufgrund der beabsichtigten Reduzierung des Abfallvolumens sind die angewendeten Verfahren in ihrer Häufigkeit und Abfolge grundsätzlich nicht eingeschränkt. Es ist durchaus möglich, dass sich Dekontaminationsverfahren und Messungen wiederholen und Freigabeoptionen nachträglich geändert werden müssen (Thierfeldt und Schartmann 2009, S. 107). Da die Abfälle erhebliche Unterschiede aufweisen (Thierfeldt und Schartmann 2009, S. 82), können auch die Stoffströme unterschiedlich ausfallen.

Die Anwendung von Zerlege-, Dekontaminations-, Mess- oder Konditionierungsverfahren auf Stoffe sowie deren Transport, Verpackung, Übergabe oder Ausschleusung können als *Verarbeitungsschritte* im Sinne dieser Arbeit verstanden werden (vgl. Abschnitt 2.1.2). In der Regel sind die dafür benötigten Verarbeitungskapazitäten begrenzt. So wird beispielsweise im Kernkraftwerk Philippsburg zur Ausschleusung eine am Reaktorgebäude angebrachte Containerschleuse benötigt (EnBW Kernkraft GmbH 2018, S. 17). Zur Durchführung der genannten Verfahren werden Arbeitskräfte und Werkzeuge bzw. Anlagen benötigt, die beispielsweise aufgrund mangelnder Verfügbarkeit auf dem Markt (Laufs 2018, S. 339) oder wegen Platzmangels innerhalb der Anlagen nicht unbegrenzt zur Verfügung stehen. So gestalteten sich die Rückbauarbeiten am Versuchsreaktor Jülich „unter anderem durch die sehr beengten Platzverhältnisse in der Anlage schwierig und waren mit Verzögerungen gegenüber dem Zeitplan verbunden“ (Thierfeldt und Schartmann 2009, S. 56).

Die verfügbaren Flächen müssen neben den speziell für den Rückbau benötigten Werkzeugen und Anlagen auch die Stoffe selbst aufnehmen. Die EnBW Kernkraft GmbH (2018, S. 38) unterscheidet Lagerflächen zur Bereitstellungslagerung, zur Pufferlagerung und zur Abklinglagerung. Während die Abklinglagerung längerfristig ist (Laufs 2018, S. 336), ist davon auszugehen, dass die Bereitstellungs- und Pufferlagerung für die operative Planung des Rückbaus von Relevanz sind. So dienen die Pufferlager einer „kontinuierlichen und wirtschaftlichen Auslastung der nachfolgenden Reststoffbearbeitungsstationen“ (Laufs 2018, S. 339). Bereitstellungslager lassen sich nicht eindeutig von Pufferlagern abgrenzen. Laufs (2018, S. 350) spricht bei der Lagerung hochradioaktiver Abfälle von Zwischenlagern und bei der Lagerung schwach- und mittelradioaktiver Abfälle von Bereitstellungslagern. Für das Kernkraftwerk Trino zeigen Brusa u. a. (2002, S. 7) einen Grundriss der Gebäude zur

Reststoffbearbeitung, in welchem die verschiedenen Stoffstrompfade und Bearbeitungsstationen eingezeichnet sind. Es ist deutlich zu erkennen, dass die verfügbaren Flächen begrenzt sind und mit Engpässen zu rechnen ist. Grundsätzlich ist auch davon auszugehen, dass die verfügbaren Flächen je nach Rückbaufortschritt variieren (E.ON Kernkraft GmbH 2008, S. 14; ARGE Stilllegung Biblis 2020, S. 31).

Die EnBW Kernkraft GmbH (2018, S. 38) errichtet für den Rückbau des Kernkraftwerks Philippsburg ein sogenanntes Reststoffbearbeitungszentrum, das die Verarbeitung der in Philippsburg abgebauten Anlagenteile räumlich getrennt von den kerntechnischen Anlagen übernimmt (siehe auch Laufs 2018, S. 340). Auch in der Studie zum Rückbau des Kernkraftwerks Trino ist von einer separat errichteten „waste management facility“ die Rede. Andere Rückbauprojekte, wie das Kernkraftwerk Obrigheim, erlauben jedoch die Reststoffbearbeitung innerhalb der bestehenden Gebäude (Laufs 2018, S. 28). „Der Abbau von Anlagenteilen, ggf. inklusive ihrer Bearbeitung im Abbaubereich (z. B. Zerlegung auf Transportmaß), wird grundsätzlich von der weiteren Bearbeitung radioaktiver Stoffe entkoppelt“, so die EnBW Kernkraft GmbH (2018, S. 18). Die Verarbeitung von Stoffen kann auch extern erfolgen (EnBW Kernkraft GmbH 2018, S. 18; Thierfeldt und Schartmann 2009, S. 53).

#### 3.2.3 Eignung zum Einsatz eines Optimierungswerkzeugs

Thierfeldt und Schartmann (2009, S. 153) geben an, dass Kosteneinsparungen beim Rückbau kerntechnischer Anlagen möglich sind, „wenn vorhandene Ressourcen so effizient wie möglich genutzt werden“. Im Hinblick auf die Projektdauer stellen sie fest: „Da bis zur Hälfte der Gesamtkosten eines Projekts auf Personalkosten und sonstige Kosten, die direkt mit der Abwicklungsdauer verbunden sind, entfallen können, werden alle Faktoren vermieden, die ein Projekt verzögern könnten“ (Thierfeldt und Schartmann 2009, S. 34). Als sonstige Kosten sind in diesem Zusammenhang beispielsweise Sicherheitsvorkehrungen, regelmäßige Prüfungen oder Behebungen von altersbedingten Schäden an der Infrastruktur zu verstehen (Thierfeldt und Schartmann 2009, S. 153). Auch Laufs (2018, S. 336) bestätigt, dass der Rückbau möglichst schnell erfolgen und gleichzeitig die Umweltbelastungen sowie das Aufkommen an radioaktiven Abfällen so gering wie möglich gehalten werden sollten. Aufgrund dieser mit dem kerntechnischen Rückbau verbundenen Zielsetzungen sowie der komplexen Rahmenbedingungen und Einschränkungen, insbesondere in Bezug auf die Stoffströme, sind kerntechnische Rückbauprojekte ein geeignetes Anwendungsbeispiel für das in dieser Arbeit entwickelte Optimierungswerkzeug. Belegt wird die Notwendigkeit zum Einsatz geeigneter Planungsmethoden durch die bei einigen abgeschlossenen Rückbauprojekten beobachteten großen Abweichungen von der ursprünglichen Planung (Hübner 2019, S. 13).

Das nachfolgend vorgestellte Modell zur stoffstrombasierten Planung von Rückbauprojekten ist durch kerntechnische Rückbauprojekte inspiriert. Deshalb werden die im nächsten Abschnitt beschriebenen Modellanforderungen teilweise durch den kerntechnischen Rückbau motiviert. Die anschließende Modellentwicklung erfolgt auf generische Weise, um zu

unterstreichen, dass ein Einsatz auch für Rückbauprojekte in anderen Bereichen möglich ist.

### 3.3 Modellanforderungen, -einschränkungen und -abgrenzungen

Während in Abschnitt 3.1 die Rahmenbedingungen der behandelten Problemstellung erläutert wurden, befasst sich dieser Abschnitt mit deren Konkretisierung im Hinblick auf die Entwicklung eines Modells. Es wird systematisch zwischen Modellanforderungen, -einschränkungen und -abgrenzungen unterschieden. *Modellanforderungen* sind Erfordernisse, die ein Modell zur stoffstrombasierten Rückbauplanung erfüllen soll, um die Eignung des stoffstrombasierten Optimierungswerkzeugs für möglichst viele praktische Anwendungen sicherzustellen. Komplementär zu den Modellanforderungen begrenzen *Modelleinschränkungen* die Allgemeingültigkeit und stellen sicher, dass das Modell handhabbar und auf den Untersuchungsrahmen dieser Arbeit fokussiert bleibt. Modelleinschränkungen sind begründet, falls hinreichende Evidenz besteht oder falls entsprechende Modellerweiterungen mit in der Literatur bekannten Konzepten umsetzbar sind (vgl. Abschnitt 4.5). Andernfalls begründet sich im Umkehrschluss eine Modellanforderung. Wenn beispielsweise nicht geklärt ist, ob die Stofffreisetzung in der Praxis zum Start, während der Ausführung oder zum Ende eines Rückbauvorgangs erfolgt, begründet dies die Modellanforderung, alle drei genannten Arten der Stofffreisetzung modellieren zu können (vgl. Modellanforderung (4)). Während Modellanforderungen und Modelleinschränkungen sich im Kontext der behandelten Problemstellung bewegen, dienen *Modellabgrenzungen* der klaren Unterscheidung zu Problemstellungen aus anderen Planungsaufgaben.

Folgende **Anforderungen** werden an ein Modell zur stoffstrombasierten Planung von Rückbauprojekten gestellt:

- (1) Stoffströme sollen aus räumlicher Perspektive mit frei wählbaren Aufspaltungen und Zusammenführungen modelliert werden können. Sie sollen beliebige Lager und Verarbeitungsschritte durchlaufen können, wobei explizit auch Zyklen möglich sein sollen. Es ist zu beachten, dass diese Zyklen der Stoffströme nicht mit den in Abschnitt 2.2.3 angesprochenen Zyklen in Vorgangsknotennetzplänen zu verwechseln sind: Zyklen der Stoffströme sind physischer Art, während Zyklen in Vorgangsknotennetzplänen sich auf die Zeitstruktur beziehen.
- (2) Die durch jeden Rückbauvorgang freigesetzten Stoffströme sollen individuell beschrieben werden können. Insbesondere soll es möglich sein, für einen Rückbauvorgang verschiedenartige Stoffströme modellieren zu können. Beispielsweise eine bestimmte Menge an Stoffen, welche in den Wertstoffkreislauf rückgeführt werden kann und gleichzeitig eine andere Menge an Stoffe, welche gesondert beseitigt werden muss. Diese Modellanforderung resultiert aus der Absicht, individuelle stoffliche

Charakteristika, welche sich aus der Nutzungsvergangenheit eines Stoffs ergeben, in die Planung einbeziehen zu können.

- (3) Engpässe im Stoffstrombereich, d. h. begrenzte Lager- und Verarbeitungskapazitäten, sollen individuell und flexibel modelliert werden können. Zur Modellierung der begrenzten Lagerkapazitäten soll eine frei wählbare Menge an Lagern spezifiziert und mit individuellen Kapazitäten ausgestattet werden können. Zur Modellierung der begrenzten Verarbeitungskapazitäten soll ein geeignetes Konzept erarbeitet werden, mit welchem die Verursacher einzelner Kapazitätsbeschränkungen (z. B. begrenzt verfügbare Fachkräfte oder begrenzt verfügbare Bearbeitungsstationen) spezifiziert und mit individuellen Kapazitäten ausgestattet werden können. Dabei ist zu berücksichtigen, dass hierbei gegebenenfalls ein Austausch zwischen Rückbau- und Stoffstrombereich stattfindet, indem beispielsweise Fachkräfte sowohl bei der Ausführung eines Rückbauvorgangs als auch bei der Stoffverarbeitung eingesetzt werden.
- (4) Stofffreisetzungen sollen frei wählbar zum Start, gleichmäßig während der Ausführung und zum Ende eines Rückbauvorgangs erfolgen können.

Folgende **Einschränkungen** werden für ein Modell zur stoffstrombasierten Planung von Rückbauprojekten getroffen:

- (1) Es sollen keine zeitlichen Höchstabstände zwischen Verarbeitungsschritten oder zwischen Rückbauvorgängen modelliert werden. Dies schließt ein, dass auch keine Fälligkeitstermine für Rückbauvorgänge zu berücksichtigen sind. Diese Einschränkung ermöglicht eine übersichtlichere und intuitivere Darstellung der in Kapitel 5 vorgestellten Lösungsverfahren. Sie verringert auch die Laufzeitkomplexität der Algorithmen. Konzepte zur Modellierung zeitlicher Höchstabstände werden in der Literatur umfangreich behandelt und können auf das in dieser Arbeit entwickelte Modell übertragen werden. Auf Details wird in Abschnitt 4.5.1 eingegangen.
- (2) Es wird davon ausgegangen, dass die Stoffe keine Haltbarkeitsbeschränkungen aufweisen. Somit ist die Modellierung von Höchstlagerdauern nicht erforderlich. Erforderliche Erweiterungen des Modells zur Abbildung von Höchstlagerdauern werden in ebenfalls in Abschnitt 4.5.1 behandelt.
- (3) Es wird davon ausgegangen, dass keine Mindestbestände für Lager zu beachten sind. Dies ist Rückbauprojekten inhärent, da sie darauf ausgerichtet sind, sämtliche freigesetzten Stoffe von der Baustelle zu entfernen, d. h. zur Stoffstromsenke zu führen. Es müssen keine Reservebestände gehalten werden, wie beispielsweise im produktionswirtschaftlichen Umfeld. Aus modelltheoretischer Sicht beträgt der Mindestbestand jedes Lagers null. Wie in Abschnitt 4.5.3 gezeigt wird, können echte Mindestbestände mit denselben Konzepten modelliert werden, die auch für Höchstbestände (d. h. Lagerkapazitäten) benötigt werden.
- (4) Es wird ausschließlich die Zielsetzung zur Minimierung der Projektdauer betrachtet. Wie im vorangegangenen Abschnitt 3.2 dargelegt, wird beim kerntechnischen Rück-



bau eine möglichst schnelle Projektdurchführung beabsichtigt. Dies reduziert hohe Personalkosten und Kosten zur Aufrechterhaltung der Infrastruktur. Auch für andere Rückbauprojekte wird angenommen, dass sich die Projektdauer weitestgehend proportional zu den Projektkosten verhält und somit ein adäquater Stellvertreter für die aus ökonomischer Sicht zentrale Zielsetzung der Kostenminimierung ist. Da das Lösungsverfahren zur Minimierung der Projektdauer auch für andere reguläre Zielsetzungen geeignet ist (vgl. Abschnitt 2.2.8), können die entwickelten Konzepte innerhalb dieses Rahmens übertragen werden.

- (5) Es wird davon ausgegangen, dass Verarbeitungs- und Lagerkapazitäten im Stoffstrombereich über die gesamte Dauer des Rückbauprojekts konstant sind, d. h. in jeder Periode im selben Umfang zur Verfügung stehen. Anders ausgedrückt, werden keine zeitlichen Variabilitäten von Verarbeitungs- und Lagerkapazitäten modelliert, wie beispielsweise geplante Abgänge von Arbeitskräften oder die Schaffung eines neuen Lagers nach Erreichen eines bestimmten Projektfortschritts. In Abschnitt 4.5.2 wird gezeigt, dass zeitlich variable Verarbeitungs- und Lagerkapazitäten ohne tiefgreifende Modellerweiterungen abgebildet werden können.

In dieser Arbeit liegt der Fokus primär auf der Ablaufplanung des Rückbauprojekts, wie in Abschnitt 3.1 dargelegt. Diese Planungsaufgabe ist von logistischen und produktionswirtschaftlichen Planungsaufgaben abzugrenzen, welche auf taktischer oder operativer Ebene angesiedelt sind. Eine Vermischung verschiedener Planungsaufgaben über alle Ebenen ist zu vermeiden, um die Größe und Komplexität eines Modells und des damit beschriebenen mathematischen Problems beherrschen zu können. Außerdem ist auch im Hinblick auf die Anwendungsorientierung davon auszugehen, dass verschiedene Planungsaufgaben verschiedenen Stellen innerhalb einer Organisation zugeordnet sind und somit vollintegrierte Modelle in ihrem praktischen Nutzen begrenzt wären. Daher gelten für das zu entwickelnde Modell folgende **Abgrenzungen** zu anderen Planungsaufgaben:

- (1) Umrüstzeiten werden nicht berücksichtigt. Diese Einschränkung ist mit der in Abschnitt 3.1 vorgenommenen Abstufung von primärer und sekundärer Fragestellung begründet. Da der exakte zeitliche Ablauf der Stoffverarbeitung nur auf sekundärer Ebene betrachtet wird, ist davon auszugehen, dass Umrüstzeiten nicht in exakter zeitlicher Auflösung den einzelnen Verarbeitungsschritten zugeordnet werden müssen. Stattdessen können die erforderlichen Umrüstzeiten über gewisse Zeiträume modell-exogen abgeschätzt und auf die Dauern der Verarbeitungsschritte umgelegt werden. Bei der operativen Maschinenbelegungsplanung für den Stoffstrombereich können dann die tatsächlich erforderlichen Umrüstvorgänge einbezogen werden. Hierbei handelt es sich jedoch um eine produktionswirtschaftliche Fragestellung, für die andere Modelle existieren. Aus denselben Gründen werden auch keine Transportzeiten oder -wege modelliert.
- (2) Es werden keine Entscheidungen zwischen verschiedenen Lagerflächen getroffen. Dies ist Aufgabe der logistischen Planung. Stattdessen ist für jeden Stoffstrom modell-exogen vorzugeben, welche Lager er durchläuft. Lager sind im vorliegenden Modell nicht räumlicher, sondern organisatorischer Art. Ein Lager im Sinne des Modells

kann sich beispielsweise über drei verschiedene Flächen innerhalb eines Gebäudes erstrecken. Es wird jedoch organisatorisch als ein einzelnes Lager angesehen, wenn auf den drei Flächen sämtliche Stoffe gelagert werden, die für einen bestimmten Verarbeitungsschritt vorgesehen sind.

- (3) Ein- und Auslagerungsreihenfolgen und damit verbundene Blockaden von Objekten in Lagern werden nicht berücksichtigt. Außerdem bleibt die Geometrie von Objekten unberücksichtigt. Lagerbestände und -kapazitäten werden über *Stoffeinheiten* (z. B. Container, Tonnen, Kubikmeter, etc.) beschrieben, die sich am benötigten Lagervolumen (bei dreidimensionaler Kapazitätsbeschränkung) bzw. an der benötigten Lagerfläche (bei zweidimensionaler Kapazitätsbeschränkung) orientieren. Es wird davon ausgegangen, dass jede Stoffeinheit gleich viel Lagervolumen bzw. Lagerfläche beansprucht. Die Beschreibung der freigesetzten Stoffe mit geeigneten Stoffeinheiten erfolgt im Rahmen einer modellexogenen Entscheidung. Theoretisch ist eine Stoffeinheit so zu wählen, dass sie dem größten gemeinsamen Teiler aller zur Beschreibung der Stoffströme erforderlichen Mengenangaben entspricht. Sofern sich Stoffströme nicht vermischen, ist die reale Beschaffenheit einer Stoffeinheit aus modelltheoretischer Sicht irrelevant. Beispielsweise kann die Stoffeinheit für sämtliche Betonstoffströme einem Kubikmeter und für sämtliche Stahlstoffströme einem Container entsprechen. Lediglich bei einer Vermischung in einem Pufferlager ist es erforderlich, eine gemeinsame Stoffeinheit festzulegen, mit der der Lagerbestand bemessen wird (siehe Abschnitt 4.4.4 für ein Beispiel zur Festlegung der Stoffeinheiten). Fragestellungen im Zusammenhang mit der Beschaffenheit von Stoffen, der Stapelung, der Gewichtsrestriktionen, der Blockaden, etc. sind Gegenstand logistischer Planungsprobleme.

## 3.4 Das Schedulingproblem unter Ressourcen- und Stoffstromrestriktionen (RMCPSP)

In Abschnitt 3.4.1 wird das *Stoffstrommodell* vorgestellt, welches zur Beschreibung der bei einem Rückbauprojekt freigesetzten Stoffströme dient. Anschließend wird in Abschnitt 3.4.2 das betrachtete Schedulingproblem unter Ressourcen- und Stoffstromrestriktionen (RMCPSP) formal definiert. Da es sich hierbei um eine konzeptionelle Formulierung mit hohem Abstraktionsgrad handelt, werden in Abschnitt 3.4.3 darauf aufbauende Modellierungs- und Lösungsansätze skizziert.

### 3.4.1 Stoffstrommodell

Analog zur Vorgehensweise bei der Diskretisierung der Zeit (siehe Abschnitt 2.2.1) können auch Stoffmengen ohne Beschränkung der Allgemeinheit diskretisiert werden. Dies erfolgt mittels Wahl einer Stoffeinheit, die dem größten gemeinsamen Teiler aller Mengenangaben entspricht. Die Diskretisierung vereinfacht einerseits die Modellierung und die Problem-

formulierung. Andererseits stellt sie auch die effiziente Lösbarkeit von Instanzen sicher, da sie Stoffe abzählbar macht und somit implizit eine Obergrenze für die Granularität beschreibt, mit welcher Stoffe geplant werden können. Diese Granularität wird bei der Modellierung des Stoffstrombereichs in Abschnitt 4.2.1 relevant.

Die *Stoffströme* im Sinne des Modells werden mithilfe zweier Typen von Problemparametern beschrieben: Die Menge der (paarweise verschiedenen) *Stoffstrompfade*  $\mathcal{W}$  und die Anzahl  $f_{i,w}^y$  der Stoffeinheiten, die jeder Rückbauvorgang  $i \in V$  freisetzt und die jeden Stoffstrompfad  $w \in \mathcal{W}$  durchlaufen – kurz als (*Stoff-*)*freisetzung* bezeichnet.

Ein Stoffstrompfad wird als Folge von Verarbeitungsschritten und Lagern definiert. Jeder Stoffstrompfad endet mit einer Senke, da sämtliche freigesetzten Stoffströme von der Baustelle entfernt werden sollen, wie in Modelleinschränkung (3) beschrieben. Die Senke bildet die Systemgrenze und ist nicht entscheidungsrelevant, weshalb es ausreicht, im Modell von einer einzigen Senke  $s$  zu sprechen (selbst wenn es, wie im kerntechnischen Rückbau, unterschiedliche Entsorgungsziele gibt). In Stoffstrompfaden können dieselben Verarbeitungsschritte und Lager beliebig oft enthalten sein, wodurch sich Zyklen in den Stoffströmen modellieren lassen und Modellanforderung (1) Rechnung getragen wird. Ohne Beschränkung der Allgemeinheit wird definiert, dass sich Verarbeitungsschritte und Lager in jedem Stoffstrompfad abwechseln. Außerdem muss jeder Stoffstrompfad mit einem Lager beginnen. Es handelt sich hierbei jeweils um dasjenige Lager, in das der Rückbauvorgang die Stoffeinheiten freisetzt. Falls in der Realität kein Lager vor einem Verarbeitungsschritt existiert, kann dies mit einem Lager der Kapazität null modelliert werden. Implizit erfordert ein solches Lager der Kapazität null, dass Stoffe ohne Unterbrechung weiterverarbeitet werden müssen. Für Details wird auf Abschnitt 4.4.2 verwiesen.

Formal werden Stoffstrompfade wie folgt definiert: Die Menge der Verarbeitungsschritte wird als  $\mathcal{P}$  und die Menge der Lager als  $\mathcal{R}^y$  bezeichnet. Um Verarbeitungsschritte zu indizieren, wird die Variable  $p$  verwendet und um Lager zu indizieren, wird  $k$  verwendet. Da Lager in Kapitel 4 mittels kumulativer Ressourcen modelliert werden, stimmt ihre Notation mit der in Abschnitt 2.2.6 verwendeten Notation für kumulative Ressourcen überein. Für einen Stoffstrompfad  $w \in \mathcal{W}$  gilt,  $w := (k_1(w), p_1(w), k_2(w), p_2(w), \dots, s)$ . Die Anzahl der enthaltenen Lager und Verarbeitungsschritte ist beliebig, solange sie abwechselnd enthalten sind. Mit  $k_1(w)$  wird auf das erste Lager in Stoffstrompfad  $w$  referenziert, mit  $p_1(w)$  auf den ersten Verarbeitungsschritt und so weiter. Lager  $k_1(w)$  wird als das dem Verarbeitungsschritt  $p_1(w)$  vorgeschaltete Lager bezeichnet und Lager  $k_2(w)$  als das dem Verarbeitungsschritt  $p_1(w)$  nachgeschaltete Lager und so weiter. Entsprechend der oben getroffenen Definitionen ist der kürzestmögliche Stoffstrompfad  $(k_1(w), p_1(w), s)$ . Noch kürzer wäre nur der Pfad  $(s)$ , der jedoch nicht entscheidungsrelevant ist und deshalb im Modell ignoriert werden kann.

Die Kombination aus Stoffstrompfaden und Stofffreisetzungen, wobei Letztere vorgangsspezifische Problemparameter sind, erfüllt Modellanforderung (2). Jedem Rückbauvorgang können individuell und in beliebigen Mengen Stoffe zugeordnet werden, die eine beliebige Abfolge an Verarbeitungsschritten durchlaufen und dazwischen in beliebigen Lagern ge-

puffert werden. Zum besseren Verständnis wird das Stoffstrommodell anhand des Beispiels 3.1 veranschaulicht.

**Beispiel 3.1:** Angenommen, der durch einen Rückbauvorgang 1 freigesetzte Stoffstrom besteht aus fünf Stoffeinheiten. Dieser Stoffstrom wird nachfolgend beschrieben, wobei  $W_x$  Stoffstrompfade,  $L_x$  Lager und  $P_x$  Verarbeitungsschritte bezeichnen. Es ist zu beachten, dass es sich bei  $x$  um eine laufende Nummer der exemplarischen Stoffstrompfade, Lager bzw. Verarbeitungsschritte handelt. In der oben verwendeten Notation  $k_x(w)$  bzw.  $p_x(w)$  bezieht sich der Index  $x$  hingegen auf die Position des Lagers bzw. des Verarbeitungsschritts im Stoffstrompfad.

- Rückbauvorgang 1 setzt eine Stoffeinheit frei, die Stoffstrompfad  $W1 = (L1, P1, L1, P2, L3, P3, L5, P7, s)$  durchläuft.
- Rückbauvorgang 1 setzt zwei Stoffeinheiten frei, die Stoffstrompfad  $W2 = (L4, P4, L5, P5, L5, P8, s)$  durchlaufen.
- Rückbauvorgang 1 setzt eine Stoffeinheit frei, die Stoffstrompfad  $W3 = (L4, P4, L5, P8, s)$  durchläuft.
- Rückbauvorgang 1 setzt eine Stoffeinheit frei, die Stoffstrompfad  $W4 = (L2, P6, s)$  durchläuft.

Abbildung 3.5 zeigt das Flussnetzwerk zu diesem Stoffstrom. Nach der Freisetzung durch Rückbauvorgang 1 werden eine Stoffeinheit in  $L1$ , drei Stoffeinheiten in  $L4$  und eine Stoffeinheit in  $L2$  eingelagert. Danach stehen diese Stoffeinheiten bereit für ihre Verarbeitung in  $P1$ ,  $P4$  bzw.  $P6$  (jeweils gemäß den Stoffstrompfaden). Dazu werden sie aus den vorgeschalteten Lagern  $L1$ ,  $L4$  bzw.  $L2$  ausgelagert, verarbeitet und in die nachgeschalteten Lager  $L1$  bzw.  $L5$  eingelagert. Eine Ausnahme hiervon bildet die Stoffeinheit mit dem Stoffstrompfad  $W4$ , die die Senke erreicht. Die übrigen Stoffeinheiten werden weiterverarbeitet und umgelagert, bis sie ebenfalls die Senke erreichen. Dadurch entsteht eine inkrementelle Bewegung der Stoffe in Richtung der Senke, welche neuen Platz in den Lagern schafft. Es ist ersichtlich, dass Stoffströme in diesem Beispiel zusammenfließen, auseinanderfließen und Zyklen bilden. Das Flussnetzwerk ist als Veranschaulichung dieser physische Stoffflüsse zu verstehen.  $\square$

#### 3.4.2 Konzeptionelle Problemformulierung

In Abschnitt 3.1 wurde die in dieser Arbeit betrachtete Problemstellung über eine primäre und sekundäre Frage beschrieben. Primär sind Erkenntnisse über die Auswirkungen von Engpässen im Stoffstrombereich auf die Zielsetzung und den Schedule im Rückbaubereich von Interesse. Sekundär sind Informationen über den detaillierten zeitlichen Ablauf der Stoffverarbeitung von Interesse. Dieser Abstufung entsprechend wird die stoffstrombasierte Planung von Rückbauprojekten in zwei Planungsebenen unterteilt: Die *primäre Planungsebene* befasst sich mit der Rückbauplanung und die *sekundäre Planungsebene*

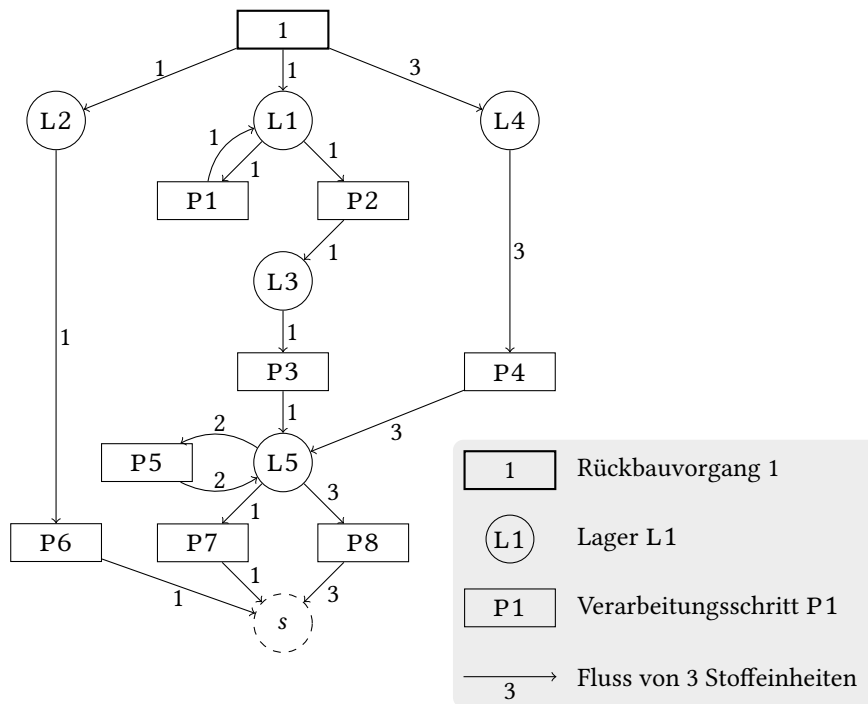


Abbildung 3.5: Flussnetzwerk zu Beispiel 3.1.

befasst sich mit der Stoffstromplanung. Das nachfolgend vorgestellte (RMCPSP) ist auf der primären Planungsebene angesiedelt. Verbindungen zur sekundären Planungsebene werden mittels abstrakter Funktionen dargestellt. Hier wird deutlich, dass es sich beim (RMCPSP) um eine hochgradig konzeptionelle Problemformulierung handelt, die lediglich der Problembeschreibung und dem Problemverständnis dient.

Einige der in Kapitel 2 vorgestellten Modellbestandteile sowie deren Notationen werden übernommen. Das zu planende Rückbauprojekt wird in die nichtunterbrechbaren Rückbauvorgänge  $i = 0, \dots, I + 1$ , zusammengefasst als Menge  $V$ , unterteilt. Rückbauvorgänge 0 und  $I + 1$  sind fiktiv und bilden den Projektstart bzw. das -ende ab. Die Dauer jedes Rückbauvorgangs  $i \in V$  wird mit  $d_i \in \mathbb{Z}_{\geq 0}$  angegeben, wobei  $d_0 = d_{I+1} := 0$  gilt. Das (RMCPSP) befasst sich gemäß Modelleinschränkung (4) mit der Suche nach einem Schedule  $S = (S_i)_{i \in V}$ , welcher die Projektdauer  $S_{I+1}$  minimiert.

Eine Menge von Vorrangbeziehungen  $E \subset V^2$  ist für Paare von Rückbauvorgängen definiert. Die Vorrangbeziehungen  $(i, i') \in E$  implizieren Vorrangrestriktionen, die als Ungleichungen

$$S_{i'} \geq S_i + d_i \quad (\forall (i, i') \in E) \quad (3.1)$$

formuliert werden (vgl. Abschnitt 2.2.2). Das (RMCPSP) und die im weiteren Verlauf dieser Arbeit entwickelten Lösungsverfahren können aufgrund der topologischen Sortierbarkeit ohne Weiteres auf das Konzept der zeitlichen Mindestabstände erweitert werden (vgl. Abschnitt 2.3.1). Zeitliche Höchstabstände werden gemäß Modelleinschränkung (1) nicht berücksichtigt.

Zur Durchführung des Rückbauprojekts wird eine Menge  $\mathcal{R}^\alpha$  an begrenzt verfügbaren erneuerbaren Ressourcen benötigt (vgl. Abschnitt 2.2.4). Für jede erneuerbare Ressource  $k \in \mathcal{R}^\alpha$  ist eine erneuerbare Ressourcenkapazität  $R_k^\alpha \in \mathbb{Z}_{\geq 0}$  gegeben. Die erneuerbare Ressourcennutzung wird für jeden Rückbauvorgang  $i \in V$  und jede erneuerbare Ressource  $k \in \mathcal{R}^\alpha$  mit  $r_{ik}^\alpha \in \mathbb{Z}_{\geq 0}$  angegeben. Die Anzahl der durch alle Rückbauvorgänge zu einem Zeitpunkt  $t$  genutzten Einheiten einer erneuerbaren Ressource  $k \in \mathcal{R}^\alpha$  ergibt sich bei gegebenem Schedule  $S$  zu

$$r_k^\alpha(S, t) := \sum_{i \in \mathcal{A}^\alpha(S, t)} r_{ik}^\alpha. \quad (3.2)$$

Für die Definition der aktiven Menge  $\mathcal{A}^\alpha(S, t)$  wird auf Gleichung (2.9) verwiesen.

Bei der Beschreibung der Problemstellung in Abschnitt 3.1 wurde dargelegt, dass die Engpässe im Stoffstrombereich aus begrenzten Verarbeitungs- und Lagerkapazitäten resultieren. Zur Modellierung der begrenzten Verarbeitungskapazitäten ist es naheliegend, den bestehenden und wissenschaftlich ausgereiften Modellbestandteil der erneuerbaren Ressourcen zu verwenden. Damit lassen sich beispielsweise begrenzt verfügbare Fachkräfte oder Bearbeitungsstationen als Verursacher der Kapazitätsbeschränkung modellieren. Gemäß Modellanforderung (3) soll ein Austausch von Ressourcen zwischen Rückbau- und Stoffstrombereich ermöglicht werden. Daher greift der Stoffstrombereich auf dieselbe Menge der erneuerbaren Ressourcen  $\mathcal{R}^\alpha$  zu, wie die Rückbauvorgänge. Um dies formal auszudrücken, wird Funktion  $\Phi_k^\alpha(t)$  eingeführt. Sie beschreibt die Anzahl der zu einem Zeitpunkt  $t$  im Stoffstrombereich genutzten Einheiten einer erneuerbaren Ressource  $k \in \mathcal{R}^\alpha$ . Es handelt sich hierbei um eine Information aus der sekundären Planungsebene. Deshalb wird zunächst kein Mechanismus zur Auswertung der Funktion  $\Phi_k^\alpha(t)$  definiert, sondern sie wird als abstrakt angesehen. Auf Ansätze zur konkreten Modellierung wird im nachfolgenden Abschnitt 3.4.3 eingegangen. Unter Zusammenführung der Ressourcennutzungen im Rückbau- und Stoffstrombereich werden die erneuerbaren Ressourcenrestriktionen somit als Ungleichungen

$$r_k^\alpha(S, t) + \Phi_k^\alpha(t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0) \quad (3.3)$$

formuliert. Es ist zu beachten, dass sich mit dieser Modellierung Rückbau- und Stoffstrombereich zwar erneuerbare Ressourcen teilen können, aber nicht müssen. Die tatsächlichen Ressourcennutzungen sind frei wählbare Problemparameter, wodurch Projektverantwortliche auch getrennte Ressourcenpools modellieren können.

Die begrenzten Lagerkapazitäten bilden das Komplement zu den begrenzten Verarbeitungskapazitäten. Nur das Zusammenspiel beider Kapazitätsbeschränkungen kann einen Rückkopplungseffekt auf das Rückbauprojekt haben. Dies wird bei der Betrachtung beider Extremfälle deutlich: Selbst wenn überhaupt keine Verarbeitungskapazitäten für freigesetzte Stoffe zur Verfügung stehen, aber die Lagerkapazitäten unbegrenzt sind, wird die Ausführung des Rückbauprojekts nicht beeinträchtigt. Dasselbe gilt für den umgekehrten Fall, in welchem unbegrenzte Kapazitäten zur sofortigen Verarbeitung jeder Stoffeinheit zur Verfügung stehen, aber keine Lagerkapazitäten. Zur Modellierung der begrenzten Lagerkapazitäten wird für jedes Lager  $k \in \mathcal{R}^\gamma$  eine *Lagerkapazität*  $R_k^\gamma \in \mathbb{Z}_{\geq 0}$  definiert. Die

Lagerkapazität und die Stoffmengen, die ein Lager durchlaufen, müssen in derselben Stoffeinheit beschrieben werden. Da es sich bei der Menge der erneuerbaren Ressourcen  $\mathcal{R}^\alpha$  und bei der Menge der Lager  $\mathcal{R}^Y$  sowie bei ihren jeweiligen Kapazitäten  $R_k^\alpha$  bzw.  $R_k^Y$  um Problemparameter handelt, wird Modellanforderung (3) erfüllt: Engpässe im Stoffstrombereich können individuell und flexibel modelliert werden.

Entsprechend Modellanforderung (4) sollen die Stofffreisetzungen frei wählbar zum Start, gleichmäßig während der Ausführung und zum Ende eines Rückbauvorgangs erfolgen können. Dazu wird das in Abschnitt 2.2.6 vorgestellte Konzept der Nutzungsarten auf die Stofffreisetzungen übertragen: Für jeden Rückbauvorgang  $i \in V$  und jeden Stoffstrompfad  $w \in \mathcal{W}$  wird die Stofffreisetzung  $f_{iw}^Y$  durch den dreidimensionalen Vektor  $(f_{iw}^{Y,start}, f_{iw}^{Y,lin}, f_{iw}^{Y,end})$  ersetzt. Dabei bezeichnet  $f_{iw}^{Y,start} \in \mathbb{Z}_{\geq 0}$  die Freisetzung zum Start von  $i$ ,  $f_{iw}^{Y,lin} \in \mathbb{Z}_{\geq 0}$  die gleichmäßige Freisetzung während der Ausführung von  $i$  und  $f_{iw}^{Y,end} \in \mathbb{Z}_{\geq 0}$  die Freisetzung zum Ende von  $i$ . Zusammengefasst wird von hierbei von drei *Freisetzungsarten* gesprochen.

Die Anzahl der durch einen Rückbauvorgang  $i \in V$  mit gegebenem Startzeitpunkt  $S_i$  zu einem Zeitpunkt  $t \geq S_i$  freigesetzten Stoffeinheiten, die Stoffstrompfad  $w \in \mathcal{W}$  durchlaufen, ergibt sich zu

$$f_{iw}^Y(S_i, t) := \begin{cases} f_{iw}^{Y,start} & \text{falls } t = S_i, \\ f_{iw}^{Y,start} + \lfloor (t - S_i) f_{iw}^{Y,lin} / d_i \rfloor & \text{falls } S_i < t < S_i + d_i, \\ f_{iw}^{Y,start} + f_{iw}^{Y,lin} + f_{iw}^{Y,end} & \text{falls } t \geq S_i + d_i. \end{cases} \quad (3.4)$$

Falls  $t = S_i$ , ist die Freisetzung zum Vorgangsstart erfolgt. Falls  $S_i < t < S_i + d_i$ , sind die Freisetzung zum Vorgangsstart und die anteilige Freisetzung während der Vorgangsausführung erfolgt. Letztere ergibt sich aus der Multiplikation der seit Vorgangsstart vergangenen Zeit  $(t - S_i)$  und der Freisetzung pro Periode  $f_{iw}^{Y,lin} / d_i$ . Die Abrundungsfunktion  $\lfloor \cdot \rfloor$  ist aufgrund der Diskretisierung der Stoffmengen erforderlich. Dadurch wird sichergestellt, dass Stoffeinheiten zu keinem Zeitpunkt aufgeteilt werden, sondern erst als freigesetzt gelten, wenn sie vollständig zur Verfügung stehen. Falls  $t \geq S_i + d_i$ , sind alle drei Freisetzungsarten erfolgt.

Die Freisetzungen sind, wie an den Indizes der Notation erkennbar, für Stoffstrompfade definiert. Um ihre Auswirkungen auf die begrenzte Lagerkapazität quantifizieren zu können, ist eine Umschreibung auf die Freisetzungen in die jeweiligen Lager erforderlich. Dazu wird für ein Lager  $k \in \mathcal{R}^Y$  über alle Stoffstrompfade  $w \in \mathcal{W}$  aufsummiert, die mit  $k$  beginnen. Ein Stoffstrompfad  $w$  beginnt mit  $k$ , falls  $k_1(w) = k$  gilt ( $k_1(w)$  referenziert auf das erste Lager in  $w$ ). Die Anzahl der durch einen Rückbauvorgang  $i \in V$  mit gegebenem Startzeitpunkt  $S_i$  zu einem Zeitpunkt  $t \geq S_i$  in ein Lager  $k \in \mathcal{R}^Y$  freigesetzten Stoffeinheiten ergibt sich formal zu

$$f_{ik}^Y(S_i, t) := \sum_{w \in \mathcal{W} | k_1(w) = k} f_{iw}^Y(S_i, t). \quad (3.5)$$

Die gesamte Anzahl der durch Rückbauvorgang  $i$  in Lager  $k$  freigesetzten Stoffeinheiten wird als  $f_{ik}^Y(\cdot, \infty)$  bezeichnet. Da dieser Wert unabhängig vom Startzeitpunkt  $S_i$  ist, wird in der Notation ein  $\cdot$  verwendet.

Die Anzahl der durch alle Rückbauvorgänge zu einem Zeitpunkt  $t$  in ein Lager  $k \in \mathcal{R}^Y$  freigesetzten Stoffeinheiten ergibt sich bei gegebenem Schedule  $S$  zu

$$f_k^Y(S, t) := \sum_{i \in \mathcal{A}^Y(S, t)} f_{ik}^Y(S_i, t). \quad (3.6)$$

Für die Definition der aktiven Menge  $\mathcal{A}^Y(S, t)$  wird auf Gleichung (2.20) verwiesen.

In ein Lager  $k$  können nur solange Stoffe freigesetzt werden, bis dessen Kapazität ausgeschöpft ist. Diese zu einem Zeitpunkt  $t$  verfügbare Lagerkapazität wird mit Funktion  $\Gamma_k^Y(t)$  ausgedrückt, welche einerseits von der Lagerkapazität  $R_k^Y$  und andererseits davon, wie schnell zuvor freigesetzte Stoffeinheiten verarbeitet und zur Senke bewegt werden können, abhängt. Auch hierbei handelt es sich um eine Information aus der sekundären Planungsebene, weshalb  $\Gamma_k^Y(t)$  als abstrakt anzusehen ist. Die verfügbare Lagerkapazität impliziert Lagerrestriktionen für das Scheduling, die als Ungleichungen

$$f_k^Y(S, t) \leq \Gamma_k^Y(t) \quad (\forall k \in \mathcal{R}^Y, t \geq 0) \quad (3.7)$$

formuliert werden.

Die formale Beschreibung des (RMCPSP) lautet zusammengefasst:

$$\left. \begin{array}{ll} \min_S & z(S) := S_{I+1} \\ \text{u. d. R.} & S_{i'} \geq S_i + d_i \quad (\forall (i, i') \in E); \\ & r_k^\alpha(S, t) + \Phi_k^\alpha(t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0); \\ & f_k^Y(S, t) \leq \Gamma_k^Y(t) \quad (\forall k \in \mathcal{R}^Y, t \geq 0); \\ & S_0 = 0; \\ & S_i \geq 0 \quad (i = 1, \dots, I + 1). \end{array} \right\} \quad (\text{RMCPSP})$$

Eine Instanz des (RMCPSP) wird als  $\mathcal{I}_{\text{RMCPSP}}$  bezeichnet. Tabelle 3.1 fasst die zur Formulierung des (RMCPSP) eingeführte Notation zusammen. Im ersten Abschnitt der Tabelle werden alle Problemparameter aufgelistet, aus denen sich eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  zusammensetzt. Eine Belegung der Problemparameter resultiert aus modellexogenen Entscheidungen. Im zweiten Abschnitt der Tabelle werden die Entscheidungsvariablen und Funktionen aufgelistet, deren Belegung auf modellendogenen Entscheidungen beruht.

### 3.4.3 Modellierungs- und Lösungsansätze

Das (RMCPSP) dient wie erläutert lediglich der konzeptionellen Problemformulierung. Es bewegt sich auf der primären Planungsebene der Rückbauplanung und drückt mit Hilfe



Notation	Bedeutung
<b>Problemparameter (modellexogen)</b>	
$i \in V$	Rückbauvorgang
$d_i \in \mathbb{Z}_{\geq 0}$	Dauer des Rückbauvorgangs $i$
$(i, i') \in E$	Vorrangbeziehung
$k \in \mathcal{R}^\alpha$	erneuerbare Ressource
$r_{ik}^\alpha \in \mathbb{Z}_{\geq 0}$	Nutzung der erneuerbaren Ressource $k \in \mathcal{R}^\alpha$ durch Rückbauvorgang $i$
$R_k^\alpha \in \mathbb{Z}_{\geq 0}$	Kapazität der erneuerbaren Ressource $k \in \mathcal{R}^\alpha$
$p \in \mathcal{P}$	Verarbeitungsschritt
$k \in \mathcal{R}^Y$	Lager
$R_k^Y \in \mathbb{Z}_{\geq 0}$	Kapazität des Lagers $k \in \mathcal{R}^Y$
$w \in \mathcal{W}$	Stoffstrompfad
$f_{iw}^{Y,start} \in \mathbb{Z}_{\geq 0}$	Stofffreisetzung in Stoffstrompfad $w \in \mathcal{W}$ zum Start, ...
$f_{iw}^{Y,lin} \in \mathbb{Z}_{\geq 0}$	... gleichmäßig während der Ausführung bzw. ...
$f_{iw}^{Y,end} \in \mathbb{Z}_{\geq 0}$	... zum Ende des Rückbauvorgangs $i$
<b>Entscheidungsvariablen und Funktionen (modellendogen)</b>	
$S_i$	Startzeitpunkt des Rückbauvorgangs $i$
$r_k^\alpha(S, t)$	Anzahl der durch alle Rückbauvorgänge zu einem Zeitpunkt $t$ genutzten Einheiten der erneuerbaren Ressource $k \in \mathcal{R}^\alpha$
$\Phi_k^\alpha(t)$	Anzahl der im Stoffstrombereich zu einem Zeitpunkt $t$ genutzten Einheiten der erneuerbaren Ressource $k \in \mathcal{R}^\alpha$
$f_{iw}^Y(S_i, t)$	Anzahl der durch Rückbauvorgang $i$ zu einem Zeitpunkt $t$ freigesetzten Stoffeinheiten, die Stoffstrompfad $w \in \mathcal{W}$ durchlaufen
$f_{ik}^Y(S_i, t)$	Anzahl der durch Rückbauvorgang $i$ zu einem Zeitpunkt $t$ in Lager $k \in \mathcal{R}^Y$ freigesetzten Stoffeinheiten
$f_k^Y(S, t)$	Anzahl der durch alle Rückbauvorgänge zu einem Zeitpunkt $t$ in Lager $k \in \mathcal{R}^Y$ freigesetzten Stoffeinheiten
$\Gamma_k^Y(t)$	verfügbare Lagerkapazität zu einem Zeitpunkt $t$

**Tabelle 3.1:** Eingeführte Notation zur Formulierung des (RMCPSP).

der abstrakten Funktionen die Verbindung zur sekundären Planungsebene der Stoffstromplanung aus. Um das Problem zu lösen, ist eine Konkretisierung erforderlich, die mit einer Erweiterung des bisher vorgestellten Basismodells einhergeht. Dabei ist insbesondere zu klären, wie sich die Schnittstellen zwischen Rückbau- und Stoffstrombereich konkret formulieren lassen und wie die Ablaufplanung im Stoffstrombereich modelliert wird.

Da der Stoffstrombereich dem Rückbaubereich nachgeschaltet ist, wäre die Aufspaltung in ein zweistufiges Problem ein naheliegender Ansatz. In einer ersten Stufe wird ermittelt, welchen Durchsatz der Stoffstrombereich mit den gegebenen Verarbeitungs- und Lagerkapazitäten erreichen kann. Diese Informationen werden dann in der zweiten Stufe als Eingabe für das (RMCPSP) verwendet, um einen Schedule für den Rückbaubereich zu berechnen. Die abstrakten Funktionen  $\Phi_k^\alpha(t)$  und  $\Gamma_k^\gamma(t)$  wären somit als Problemparameter zu betrachten, welche in der ersten Stufe bestimmt wurden.

Da sich beide Planungsbereiche jedoch gegenseitig beeinflussen, wäre bei der Aufspaltung in ein zweistufiges Problem eine starke Einschränkung der Freiheitsgrade hinzunehmen. Darüber hinaus widerspricht der unidirektionale Ansatz, zuerst die Stoffverarbeitung und danach den Rückbau zu planen, der realen Problemstellung. Schließlich wird bei der Rückbauplanung spezifiziert, welche Stoffmengen zu welcher Zeit im Stoffstrombereich zu erwarten sind. Deshalb erfordert die Planung im Stoffstrombereich eine direkte Anbindung an den Rückbaubereich. Umgekehrt ist es nur mit den Informationen aus dem Stoffstrombereich möglich, im Rückbaubereich sicherzustellen, dass keine Überlastungen auftreten. Die Planung im Rückbaubereich erfordert daher ebenfalls eine direkte Anbindung an den Stoffstrombereich. Würden beide Bereiche getrennt betrachtet, müsste eine wiederholte Anpassung der jeweiligen Pläne erfolgen, was die Gefahr von Inkonsistenzen und Ineffizienzen birgt. Dies gilt insbesondere, da sich beide Bereiche dieselben erneuerbaren Ressourcen, wie beispielsweise Arbeitskräfte, teilen können. Daraus folgt, dass beide Planungsbereiche sinnvollerweise in einem integrierten Schedulingproblem zusammengefasst werden sollten.

Ein geeigneter Ansatz ist die Ausweitung des vorgangsbasierten Modells auf den Stoffstrombereich. Dabei wird die Verarbeitung von Stoffen in Form sogenannter *Stoffstromvorgänge* modelliert, welche zusätzlich zu den Rückbauvorgängen eingeplant werden. Dies ermöglicht die Verwendung gängiger Modellbestandteile von Schedulingproblemen, wie Zeitbeziehungen oder Ressourcennutzungen, und fördert damit die Intuitivität des Modells. Lager können mittels der in Abschnitt 2.2.6 vorgestellten kumulativen Ressourcen modelliert werden. Es ist zu beachten, dass die Stoffverarbeitung weniger die Eigenschaften eines Projekts, sondern aufgrund ihrer vom Repetitionstyp einer Serienfertigung ähnelnden Art eher die Eigenschaften einer Produktion erfüllt. Wie in Abschnitt 2.4.2 angesprochen, lassen sich die Methoden und Modelle aus dem Bereich des Projektschedulings jedoch auf den produktionswirtschaftlichen Kontext übertragen. Diesen Ansatz verfolgt Kapitel 4, wo die Entwicklung eines *Modells zur integrierten Planung von Rückbauprojekt und Stoffströmen* im Detail beschrieben wird.

## 3.5 Existierende Ansätze in der Literatur

Analog zur Struktur des vorangegangenen Abschnitts 3.4 wird für den Vergleich mit existierenden Ansätzen in der Literatur zwischen dem Stoffstrommodell (Abschnitt 3.5.1) und dem Schedulingproblem (Abschnitt 3.5.2) unterschieden. Ein Verständnis der in Kapitel 2 vermittelten Grundlagen ist erforderlich, um eine Einordnung in die Literatur vornehmen zu können. Auf Literatur zu geeigneten Lösungsverfahren wird hier nicht eingegangen, sondern auf Kapitel 5 verwiesen. Einige der nachfolgend herausgearbeiteten Unterschiede zu Ansätzen in der Literatur werden nach der Transformation des (RMCPSP) in das Modell zur integrierten Planung von Rückbauprojekt und Stoffströmen verständlicher. An entsprechenden Stellen sind deshalb Verweise auf weiterführende Erläuterungen in Kapitel 4 eingefügt.

### 3.5.1 Stoffstrommodell

Bei der Betrachtung des Flussnetzwerks in Abbildung 3.5 ist erkennbar, dass das vorgestellte Stoffstrommodell Ähnlichkeiten zu den in Abschnitt 2.4.3 erläuterten graphentheoretischen Modellen aus dem Produktionsmanagement aufweist. Diese Ähnlichkeiten werden im Folgenden erörtert.

Ähnlich zu den Input-Output-Graphen nach Kloock (1969), wo Knoten Produktionsstellen und Kanten deren Verflechtungen darstellen, können auch die Verarbeitungsschritte in Abbildung 3.5 als Produktionsstellen interpretiert werden. Im Gegensatz zu Input-Output-Graphen enthält das Stoffstrommodell dieser Arbeit neben den Verarbeitungsschritten noch Lager, die ebenfalls als Knoten dargestellt werden. Es handelt sich bei Abbildung 3.5 somit um einen bipartiten Graphen. Die Kanten beschreiben die Verflechtungen, sind wie bei Input-Output-Graphen mit den jeweiligen Mengeneinheiten bewertet und können Zyklen bilden. Die Quelle und die Senke in Abbildung 3.5 können als Beschaffungs- bzw. Absatzstelle interpretiert werden. Es ist jedoch zu beachten, dass die Stoffverarbeitung bei Rückbauprojekten nicht wie bei einer Auftragsfertigung absatzseitig (d. h. seitens der Senke) getrieben ist, sondern seitens der Quelle. Sobald ein Stoff zur Verfügung steht, kann seine Verarbeitung beginnen. Die Stoffverarbeitung kann somit eher mit einer Vorratsfertigung verglichen werden, bei der keine Entscheidungsfreiheit über die Wahl der Beschaffungszeitpunkte besteht. Stattdessen resultieren die Beschaffungszeitpunkte aus dem Schedule des Rückbaubereichs.

Wie in Abschnitt 2.4.3 bereits festgestellt, eignen sich Gozintographen nicht zur Beschreibung von Stoffströmen, da sie keine Zyklen zulassen. Somit wird an dieser Stelle nicht weiter auf sie eingegangen.

Das bipartite Flussnetzwerk in Abbildung 3.5 hat auch Ähnlichkeiten zu Petri-Netzen, wie bereits beim grafischen Vergleich zu erkennen ist. Analog zur Theorie der Petri-Netze sind die Verarbeitungsschritte als Transitionen anzusehen, da sie aktiv eine Zustandveränderung herbeiführen. Demgegenüber entsprechen Lager den Stellen, welche passive

Zustände darstellen. Ähnlich zu den Marken in Petri-Netzen bewegen sich auch Stoffe durch das Netzwerk, indem sie von Verarbeitungsschritten aus einem Lager entnommen und in ein anderes Lager gelegt werden. Es handelt sich somit um ein Verhalten, das wie bei Petri-Netzen mit dem Begriff des Schaltens beschrieben werden kann. Dennoch wird in dieser Arbeit bewusst darauf verzichtet, das Stoffstrommodell über die Petri-Netz-Theorie zu motivieren. Das liegt im Wesentlichen daran, dass der Petri-Netz-Formalismus der dynamischen Modellierung von Stoffströmen dient, während das Stoffstrommodell dieser Arbeit die Stoffströme eines Rückbauprojekts dem (RMCPSP) über Problemparameter zugänglich macht. Das dynamische Verhalten wird erst mit dem im nachfolgenden Kapitel vorgestellten Modell, welches den Stoffstrombereich in die Rückbauplanung integriert, abgebildet. Petri-Netze bringen mit ihrem Fokus auf die Schaltvorgänge auch Einschränkungen mit sich, die sie für das Stoffstrommodell dieser Arbeit ungeeignet machen. So ist es ein zentraler Bestandteil der Modellspezifikation von Petri-Netzen, dass beim Schalten einer Transition Marken in alle nachfolgenden Stellen gelegt werden. Beispielsweise wird beim Schalten von  $tr_1$  in Abbildung 2.5 eine Marke in  $st_2$  und eine Marke in  $st_3$  gelegt. Im Stoffstrommodell können jedoch auch Situationen auftreten, in denen ein Stoff nach einer Verzweigung lediglich einen von mehreren Wegen einschlägt. Deshalb ist das Flussnetzwerk in Abbildung 3.5 nicht mit Petri-Netzen zu verwechseln.

Bestimmte Überschneidungen zwischen dem Stoffstrommodell und den State-Task-Netzwerken nach Kondili u. a. (1993) lassen sich ebenfalls herausarbeiten. State-Task-Netzwerke werden mit bipartiten Graphen veranschaulicht, wobei die Zustände mit Lagern und die Aufgaben mit Verarbeitungsschritten vergleichbar sind. Allerdings wurden State-Task-Netzwerke dazu entwickelt, Stoffarten zu präzisieren. Deshalb sind sie so definiert, dass jeder Zustand einer bestimmten Stoffart entsprechen muss. Im Stoffstrommodell dieser Arbeit hingegen werden keine Einschränkungen zur Stoffart getroffen. Stattdessen können beliebige Stoffarten im selben Lager gelagert werden. Lager beschreiben hier organisatorische Zustände, während bei State-Task-Netzwerken der Fokus auf stofflichen Zuständen liegt.

Zusammengefasst lässt sich festhalten, dass das in Abschnitt 3.4.1 vorgestellte Stoffstrommodell verschiedene Konzepte der in Abschnitt 2.4.3 vorgestellten graphentheoretischen Modelle aus dem Produktionsmanagement vereint. Dennoch sollte es mit keinem dieser graphentheoretischen Modelle verwechselt werden. Das Flussnetzwerk in Abbildung 3.5 lässt am besten als einen Input-Output-Graphen beschreiben, der um die Lager als zusätzliche Knoten erweitert wurde.

Es sei außerdem darauf hingewiesen, dass das Flussnetzwerk in Abbildung 3.5 ausschließlich der Veranschaulichung der Stoffströme einer bestimmten Probleminstanz  $\bar{I}_{\text{RMCPSP}}$  dient. Das Stoffstrommodell im engeren Sinne besteht aus der Menge der Stoffstrompfade  $\mathcal{W}$  und den freigesetzten Mengen  $f_{iw}^y$  für alle Vorgänge  $i \in V$  und alle Stoffstrompfade  $w \in \mathcal{W}$ . Grafisch betrachtet werden die Stoffströme im (RMCPSP) somit über lineare Graphen beschrieben. Dies ist erforderlich, um die individuelle Zuordnung von Stoffen zu Verarbeitungsschritten aufrechtzuerhalten. Würden Stoffe stattdessen ausschließlich über ein Flussnetzwerk beschrieben, könnten die individuellen Verarbeitungswege nicht mehr nachvollzogen werden. Hierin liegt ein wesentlicher Unterschied zu einigen stoffstromba-

sierten Planungsmethoden in der Literatur, die Stoffe als *homogen* betrachten. In einem solchen homogenen Modell kann ein Vorgang einen Stoff verarbeiten bzw. verbrauchen, sobald der Stoff zur Verfügung steht. Dies ist häufig eine adäquate Annahme für Planungsprobleme im stoffstrombasierten Produktionsmanagement, denen eine Massenfertigung zugrunde liegt. Die Stoffe, die hier an einzelnen Produktionsstellen zu Zwischen- und Endprodukten verarbeitet werden, weisen dieselben Eigenschaften auf und können nach demselben Muster verarbeitet werden. In Rückbauprojekten – insbesondere im kerntechnischen Rückbau – ist jedoch davon auszugehen, dass Stoffe individuelle Charakteristika aufweisen (beispielsweise abhängig von ihrem radiologischen Ausgangszustand) und entsprechend individuell verarbeitet werden müssen. Solche Stoffe werden als *inhomogen* bezeichnet. Mit der Absicht, individuelle Charakteristika modellieren zu können, wurde auch Modellanforderung (2) begründet. Für das Schedulingmodell bedeutet dies, dass implizit eine zeitliche Abhängigkeit zwischen den auf jede Stoffeinheit anzuwendenden Verarbeitungsschritten besteht. Die Gesamtheit dieser zeitlichen Abhängigkeiten wird nachfolgend als *stoffstrominduzierte Zeitstruktur* bezeichnet. Angenommen, es muss in zwei Rückbauvorgängen je eine Wand demontiert werden, wovon jedoch nur eine der beiden Wände aufgrund einer vorliegenden Kontamination gesondert behandelt werden muss. Dann ist es zur korrekten Modellierung zwingend erforderlich, die Verbindung zwischen dem jeweiligen Rückbauvorgang und dem dadurch freigesetzten Stoffstrom aufrechtzuerhalten. Ansonsten könnte die Verarbeitung der kontaminierten Wand bereits starten, obwohl diese noch nicht rückgebaut wurde. Dadurch würde die Realitätstreue des Modells verfälscht. In Abschnitt 4.3.1 wird detaillierter auf die Notwendigkeit der stoffstrominduzierten Zeitstruktur eingegangen.

Homogene Stoffe werden beispielsweise in den Beiträgen von Blömer und Günther (1998), Schwindt und Trautmann (2000), Neumann, Schwindt und Trautmann (2005) oder Briskorn, Choi u. a. (2010) behandelt. Abgesehen von Briskorn, Choi u. a. (2010) werden in allen genannten Beiträgen Produktionsplanungsprobleme aus der Prozessindustrie behandelt, wo Stoffe verschiedene Reaktoren und Tanks durchlaufen. Briskorn, Choi u. a. (2010) motivieren ihr Problem mit einer Umladeeinrichtung, bei welcher LKWs Stoffe anliefern und wieder abholen. Ihr Ziel ist die zeitliche Planung der LKW-Abfertigung unter der Restriktion, dass Stoffe erst abgeholt werden können, wenn die entsprechende Menge zuvor angeliefert wurde. Schwindt und Trautmann (2000) beschreiben das den homogenen Stoffen zugrundeliegende Modellierungskonzept wie folgt: „Precedences between tasks [= Rückbau- und Stoffstromvorgänge] resulting from the production structure are taken into account by the constraints on the minimum inventories of input products, an approach which offers more flexibility compared to the linking of operations of consecutive levels by minimum time lags [...]. Such a linking would require a fixed matching between producing and consuming tasks.“ Bei inhomogenen Stoffen ist jedoch gerade diese feste Zuordnung der Verarbeitungsschritte, welche nacheinander auf Stoffe angewendet werden, erforderlich. Modelle mit fester Zuordnung werden beispielsweise von Voudouris und Grossmann (1996) oder Neumann und Schwindt (1997) vorgestellt. Beide Beiträge befassen sich mit Produktionsplanungsproblemen. Im Gegensatz zu den zuvor genannten Problemstellungen aus der Prozessindustrie liegen unterschiedliche Erzeugnisstrukturen für Produkte vor, die sich einen gemeinsamen Maschinenpool teilen. Dementsprechend

müssen Zeitbeziehungen zwischen einzelnen Produktionsschritten eingeführt werden, die deren individuelle Erzeugnisstruktur widerspiegeln. Dies ist vergleichbar mit der Verarbeitung inhomogener Stoffe und der dadurch hervorgerufenen stoffstrominduzierten Zeitstruktur. Die Publikationen von Voudouris und Grossmann (1996) und Neumann und Schwindt (1997) fokussieren sich jedoch wie viele andere Beiträge auf diesem Gebiet auf die Maschinenbelegungsplanung und nicht auf Engpässe im Stoffstrombereich. Deshalb berücksichtigen sie im Gegensatz zu dieser Arbeit auch keine Lagerkapazitäten.

Welche Auswirkungen die stoffstrominduzierte Zeitstruktur auf das Schedulingproblem und dessen Lösbarkeit hat, wird in den nachfolgenden Kapiteln an verschiedenen Stellen aufgegriffen. In Abschnitt 4.5.3 wird erläutert, welche Anpassungen erforderlich wären, um homogene Stoffe abzubilden.

#### 3.5.2 Schedulingproblem

Zahlreiche Veröffentlichungen in der Schedulingliteratur, welche sich mit Stoffströmen befassen, behandeln Produktionsplanungsprobleme. Es kann auf Burkard u. a. (1998), Blömer und Günther (1998), Blömer und Günther (2000), Belaid u. a. (2012), Boysen u. a. (2013), Baumann und Trautmann (2014) und Briskorn und Zeise (2019) verwiesen werden, um nur einige exemplarische Beiträge aus den vergangenen Jahrzehnten zu nennen. Die Beiträge unterscheiden sich teils beträchtlich hinsichtlich ihrer Modellbestandteile. Beispielsweise kann eine zwischengeschaltete Lagerung von Stoffen erlaubt sein oder nicht. Oder es können zur Produktion nur eine Maschine oder mehrere Maschinen zur Verfügung stehen. Von besonderer Bedeutung sind Stoffströme in der Prozessindustrie, wie Übersichtsarbeiten von Méndez u. a. (2006, mit Fokus auf sogenannte Batchprozesse) und Kallrath (2002) zeigen.

Im Kontext der Produktionsplanung, oder genauer, der Maschinenbelegungsplanung, werden häufig Begriffe wie Job-shop, Flow-shop und Open-shop verwendet. Dabei handelt es sich um Problemklassen, die unterschiedliche Produktionssysteme abbilden. Für ausführliche Informationen wird auf Pinedo (2018, S. 15) verwiesen. Da sich diese Arbeit primär mit der Planung eines Rückbauprojekts befasst, basiert die Formulierung des hier vorgestellten Schedulingproblems auf gängigen Modellen der Projektablaufplanung. Als Grundlage dient das in Abschnitt 2.2.4 vorgestellte (RCPSP), welches in der Literatur ausführlich behandelt wurde. Bei der thematischen Einordnung ist zu beachten, dass das (RCPSP) aufgrund seines generischen Charakters zahlreiche weitere Probleme aus dem Gebiet der Maschinenbelegungsplanung umfasst. Das gilt beispielsweise für die genannten Job-shop-, Flow-shop- und Open-shop-Probleme (Kolisch 1996a, S. 1; Neumann, Schwindt und Trautmann 2005, S. 500). Gerade deshalb gilt das (RCPSP) laut Kolisch (1996a) als eines der wichtigsten Schedulingprobleme. Aufgrund dieser modelltheoretischen Mächtigkeit sind die Grenzen seiner verschiedenen Anwendungsgebiete fließend, wie bereits in Abschnitt 2.4.2 erläutert. Da es sich beim (RMCPS) um eine Generalisierung des (RCPSP) handelt, sind auch hier Überschneidungen zur Produktionsplanung vorhanden. Lediglich die Begrifflichkeiten unterscheiden sich. Beispielsweise wird bei der Projektplanung von Vorgängen (engl.

activities) und Ressourcen gesprochen, während bei der Produktionsplanung in der Regel von Arbeitsgängen (engl. jobs) und Maschinen gesprochen wird.

Das (RCPSp/c) ist mit den kumulativen Ressourcen eine Generalisierung des (RCPSp), welche die Berücksichtigung begrenzter Lagerkapazitäten im Scheduling ermöglicht. Deshalb verwenden Schwindt und Trautmann (2000), Neumann, Schwindt und Trautmann (2005) und Schwindt, Fink u. a. (2007) diese Problemklasse in ihren Publikationen, um stoffstrombasierte Produktionsplanungsprobleme zu formulieren und zu lösen. Hierbei handelt es sich um diejenigen Beiträge, die die größte inhaltliche Nähe zu dieser Arbeit aufweisen.

Schwindt und Trautmann (2000, S. 503) sehen die Vorteile ihres Ansatzes in der effizienten Modellierung und Lösung großer Probleminstanzen. Die Autoren untersuchen einen exemplarischen Anwendungsfall aus der chemischen Industrie, wo Stoffe in vorbestimmten Gruppen, den sogenannten Batches, verarbeitet werden. Die Stoffe durchlaufen eine gegebene Produktionsstruktur aus Verarbeitungseinheiten und Lagern. Die Produktionsstruktur kann mit einem State-Task-Netzwerk abgebildet werden (obwohl die Autoren diesen Begriff nicht verwenden). Die Verarbeitung eines bestimmten Batches wird als „Operation“ bezeichnet. Ziel ist die Planung der Produktion einer gegebenen Menge von Endprodukten, sodass Mindest- und Höchstabstände zwischen einzelnen Operations, sowie bestimmte Verarbeitungs- und Lagerkapazitäten eingehalten werden und die Produktionsdauer minimiert wird. Schwindt und Trautmann (2000) behandeln die Operations wie Vorgänge im Sinne der Projektplanung. Die begrenzten Verarbeitungskapazitäten werden mittels erneuerbarer Ressourcen modelliert und die begrenzten Lagerkapazitäten mittels kumulativer Ressourcen. Es wird deutlich, dass die von Schwindt und Trautmann (2000) betrachtete Kombination aus Verarbeitungseinheiten und Lagern einige Überschneidungen zu den Konzepten dieser Arbeit aufweist. Dennoch lassen sich Unterschiede herausarbeiten, die in den folgenden Absätzen dargelegt werden.

Zunächst gehen Schwindt und Trautmann (2000) davon aus, dass Stoffe homogen sind und deshalb keine stoffstrominduzierte Zeitstruktur vorliegt, wie bereits im vorangegangenen Abschnitt 3.5.1 dargelegt. Außerdem nehmen sie an, dass die Verarbeitungsdauer von Stoffen unabhängig von der Batchgröße, d. h. der verarbeiteten Stoffmenge, ist. Ihre Annahme lässt sich praktisch motivieren: Wenn Stoffe, wie in der chemischen Industrie üblich, in Reaktoren verarbeitet werden, ist es unerheblich, in welchem Ausmaß die Reaktoren befüllt sind. Die Reaktion benötigt stets eine konstante Dauer. In der Stoffverarbeitung bei Rückbauprojekten ist diese Annahme jedoch oft nicht zutreffend. Zum Beispiel ist bei der Dekontamination von Stahlrohren im kerntechnischen Rückbau davon auszugehen, dass eine größere Menge an Stahlrohren auch zu einer größeren Dauer der Dekontamination führt. Die Verarbeitungsdauer skaliert linear oder auf andere Weise mit der verarbeiteten Stoffmenge.

Die gewichtigste Unterschied zwischen dem Modell von Schwindt und Trautmann (2000) und dem (RMCPSP) liegt in der organisatorischen Einordnung der Stoffverarbeitung. Weil Schwindt und Trautmann (2000) ein Produktionsproblem planen, sind die Stoffe unmittelbarer Gegenstand ihres Planungsproblems. Dadurch, dass sie die Produktion

als Projekt betrachten, kann ihr Modell auch folgendermaßen beschrieben werden: Die Verarbeitung der Stoffe *ist* das Projekt und somit primäres Untersuchungsobjekt. In der Problemstellung dieser Arbeit gilt jedoch: Die Verarbeitung der Stoffe ist ein separater Planungsbereich und kann das Projekt verzögern. Sie ist lediglich sekundäres Untersuchungsobjekt. Dieser Unterschied hat wesentliche Auswirkungen auf die Zielfunktion und das Zustandekommen des Zielfunktionswertes, d. h. der Projektdauer. Bei Schwindt und Trautmann (2000) kann die Stoffverarbeitung auf dem kritischen Pfad liegen und daher die Projektdauer direkt erhöhen. Im (RMCPSP) ist die Stoffverarbeitung hingegen über die Restriktionen eingebunden und kann somit nur indirekt im Fall eines Überlaufens der Lager die Projektdauer erhöhen. In Abschnitt 4.3.2 wird diese besondere Problemcharakteristik detailliert betrachtet. Die Diskrepanz zwischen den Anwendungsumgebungen des von Schwindt und Trautmann (2000) betrachteten Problems und des (RMCPSP) ist auch an den Problemparametern zu erkennen. Während in Schwindt und Trautmann (2000) ein Nachfragevektor als Planungsgrundlage vorgibt, welche Endprodukte in welchen Mengen zu fertigen sind, werden in dieser Arbeit Stoffströme ausgehend von den Stofffreisetzungen durch Rückbauvorgänge beschrieben.

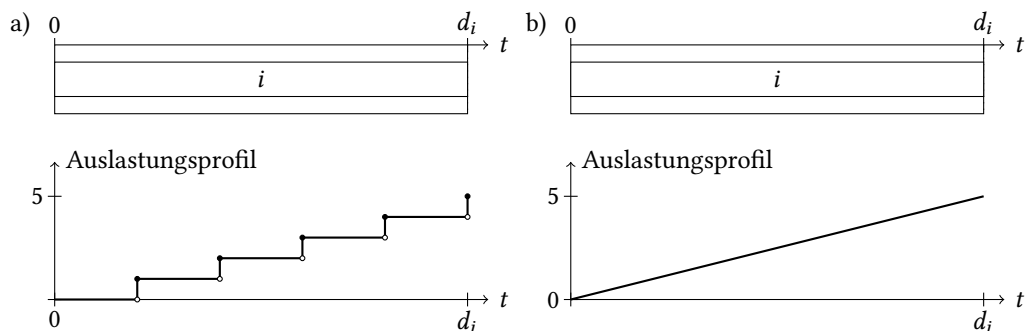
Mit der unterschiedlichen organisatorischen Einordnung der Stoffströme lässt sich auch begründen, dass Schwindt und Trautmann (2000) Umrüstzeiten berücksichtigen, während sie in dieser Arbeit gemäß Modellabgrenzung (1) keine Rolle spielen. Im Gegensatz zu Schwindt und Trautmann (2000) werden hier die auf der sekundären Planungsebene angesiedelten Stoffströme nicht mit exakter zeitlicher Auflösung geplant.

Die Batchgröße ist in Schwindt und Trautmann (2000) als Ergebnis eines vorgeschalteten Planungsproblems oder schlicht aufgrund der Reaktorgrößen vorgegeben. Die Verarbeitung eines Batches kann aufgrund von Arbeitspausen unterbrochen werden, muss aber direkt danach wieder fortgesetzt werden. Solche Arbeitspausen werden in dieser Arbeit nicht berücksichtigt, was ebenfalls aus der sekundären Stellung der Stoffverarbeitung resultiert. Der Begriff des Batches wird in der Literatur nicht immer einheitlich verwendet, wie der Beitrag von Voß und Witt (2007) zeigt. Die Autoren modellieren eine Problemstellung aus der Stahlherstellung als Mehr-Modus-Schedulingproblem (MRCPPSP). Allerdings verstehen sie unter einem Batch eine Folge von Verarbeitungsschritten, die auf einer Maschine verrichtet werden, ohne sie umrüsten zu müssen. Sie berücksichtigen auch keine Lagerkapazitäten, weshalb ihr Modell nur eingeschränkt mit dem (RMCPSP) vergleichbar ist.

Schwindt, Fink u. a. (2007) diskutieren ein nahezu identisches Problem wie Schwindt und Trautmann (2000), stellen jedoch ein alternatives, prioritätsregelbasiertes Lösungsverfahren vor. Neumann, Schwindt und Trautmann (2005) befassen sich mit einer Variante des Problems in Schwindt und Trautmann (2000), bei welcher die Stoffströme nicht in Form von Batches sondern kontinuierlich verarbeitet werden. Eine Stoffverarbeitung bzw. einen Stoffstrom definieren sie als kontinuierlich, wenn der Arbeitsfortschritt aller Vorgänge mit konstanter Rate ansteigt und folglich die Ein- und Auslagerungen ebenfalls mit konstanter Rate erfolgen (vgl. Neumann, Schwindt und Trautmann 2005, S. 496 und S. 499). Die Beiträge zum kumulativen Ressourcenprofil sind dann linear. Kumulative Ressourcen, die sich durch diese Nutzungsart auszeichnen, bezeichnen sie als „kontinuierliche



kumulative Ressourcen“. Erfolgt die Verarbeitung hingegen in Form von Batches, sprechen Neumann, Schwindt und Trautmann (2005) von einer diskreten Stoffverarbeitung bzw. einem diskreten Stoffstrom. Der Arbeitsfortschritt eines Vorgangs steigt in einem solchen Fall zum Vorgangsstart oder -ende mit unendlich großer Rate an, d. h. springt von null auf 100 %. Folglich geschehen die Ein- und Auslagerungen ebenfalls mit unendlich großer Rate, was sich an Sprüngen im kumulativen Ressourcenprofil bemerkbar macht. Kumulative Ressourcen, die entsprechend der Definitionen in Abschnitt 2.2.6 zum Vorgangsstart oder Vorgangsende genutzt werden, würden Neumann, Schwindt und Trautmann (2005) als „diskret“ bezeichnen. In dieser Arbeit wird bewusst darauf verzichtet, Stoffströme oder kumulative Ressourcen mit den Begriffen „diskret“ und „kontinuierlich“ zu charakterisieren. Stattdessen wurde das Konzept der Nutzungsarten eingeführt. Der Grund hierfür ist, dass Stoffeinheiten in dieser Arbeit diskret sind, und somit strenggenommen keine kontinuierlichen Stoffströme im Sinne von Neumann, Schwindt und Trautmann (2005) vorliegen können. Bei einer gleichmäßigen Stofffreisetzung über die Dauer eines Rückbauvorgangs sind die Beiträge zum Auslastungsprofil eines Lagers aufgrund der Abrundungsfunktion in Gleichung (3.4) nicht linear, sondern stufenförmig. Die Stufen sind allerdings kleinteilig und entsprechen jeweils einer Stoffeinheit, weshalb sich die Beiträge der linearen Form annähern. Die gleichmäßige Freisetzung während der Vorgangsausführung ist daher näherungsweise dasselbe Modellierungskonzept, wie die kontinuierlichen Stoffströme in Neumann, Schwindt und Trautmann (2005). Abbildung 3.6 illustriert den Unterschied zwischen dem Auslastungsprofil eines Lagers, wenn eine Stofffreisetzung von fünf Stoffeinheiten wie in dieser Arbeit gleichmäßig während der Ausführung eines Vorgangs erfolgt (vgl. Gleichung (3.4) und Abbildung 3.6a) und, wenn eine Stofffreisetzung von fünf Stoffeinheiten wie im Modell von Neumann, Schwindt und Trautmann (2005) kontinuierlich erfolgt (vgl. Abbildung 3.6b).



**Abbildung 3.6:** Gantt-Diagramme und Auslastungsprofile: a) bei gleichmäßiger Stofffreisetzung während der Vorgangsausführung wie in dieser Arbeit; b) bei kontinuierlichen Stoffströmen nach Neumann, Schwindt und Trautmann (2005).

Zusammenfassend lässt sich festhalten, dass stoffstrombasierte Schedulingprobleme in der existierenden Literatur die Planung der Stoffströme stets als primäres Untersuchungsobjekt betrachten. Die sekundäre Ansiedlung des Stoffstrombereichs, welcher das primär geplante Rückbauprojekt nur im Fall einer Überlastung beeinflusst, stellt eine wesentliche

Neuerung dar. Konkret lässt sich dies im (RMCPSP) am neuartigen Konzept der Lagerrestriktionen (3.7) erkennen. Diese Restriktionen besagen, dass nur solange Stoffströme freigesetzt werden können, wie sie vom Stoffstrombereich (d. h. der Gesamtheit aus Verarbeitungsschritten und Lagern mit den jeweiligen Kapazitäten) aufgenommen werden können. Im Vergleich dazu sind in der existierenden Literatur die Stoffe unmittelbar in die Vorrang- und Ressourcenrestriktionen sowie die Zielsetzung des Schedulingproblems eingebunden.

Neben den methodisch mit dieser Arbeit verwandten Beiträge von Schwindt und Trautmann (2000), Neumann, Schwindt und Trautmann (2005) und Schwindt, Fink u. a. (2007) existiert auch Literatur, welche sich spezifisch mit der Ablaufplanung von Rückbauprojekten befasst. Hervorzuheben sind die Dissertationen von Schultmann (1998), Bartels (2009), Volk (2017) und Hübner (2019) (sowie die Artikel, in welchen die Modelle der Dissertationen vorgestellt werden, wie Bartels u. a. (2011) und Hübner u. a. (2021)). Während Schultmann (1998) und Volk (2017) die Planung allgemeiner Rückbauprojekte behandeln, fokussieren sich Bartels (2009) und Hübner (2019) auf den Anwendungsfall des kerntechnischen Rückbaus. Die Modelle von Schultmann (1998), Volk (2017) und Hübner (2019) enthalten keine kumulativen Ressourcen und ermöglichen daher keine Berücksichtigung beschränkter Lagerkapazitäten. Volk (2017, S. 264) und Hübner (2019, S. 531) heben jedoch in ihrem Ausblick hervor, dass eine Erweiterung um kumulative Ressourcen die Realitätsnähe des Modells erhöhen würde. Bartels (2009) berücksichtigt zwar kumulative Ressourcen, allerdings mit einigen Einschränkungen: Die Nutzungen von kumulativen Ressourcen sind nur zum Vorgangsstart und -ende möglich (Bartels 2009, S. 15). Es wird davon ausgegangen, dass für jeden freisetzenden Rückbauvorgang (als „Demontagevorgang“ bezeichnet) ein zugehöriger Dekontaminationsvorgang (als „korrespondierender Dekontaminationsvorgang“ bezeichnet) existiert, der sämtliche freigesetzte Stoffeinheiten des Rückbauvorgangs verarbeitet (Bartels 2009, S. 52). Zwischen Rückbau und Dekontamination werden die Stoffe in einem Lager gepuffert, wobei für das gesamte Projekt nur ein einziges Lager zur Verfügung steht (Bartels 2009, S. 104). Bartels (2009) beschränkt sich demnach auf Stoffstrompfade, die alle aus demselben Lager und einem nachfolgenden Verarbeitungsschritt bestehen. In dieser Arbeit sind hingegen Stoffstrompfade mit beliebiger Länge und beliebiger Anzahl an Verarbeitungsschritten und Lagern möglich.

## 4 Integrierte Planung von Rückbauprojekt und Stoffströmen

Wie in den Modellierungsansätzen in Abschnitt 3.4.3 dargelegt, dient das (RMCPSP) der konzeptionellen Abgrenzung zwischen primärer und sekundärer Planungsebene. Zur Lösung des (RMCPSP) ist eine Konkretisierung erforderlich, die mit einer Erweiterung des bisher vorgestellten Basismodells einhergeht. Ein geeigneter Ansatz hierfür ist es, den Stoffstrombereich analog zum Rückbaubereich vorgangsbasiert zu modellieren. Dadurch werden das Rückbauprojekt und die Stoffströme integriert geplant, d. h., die Abstufung zwischen primärer und sekundärer Ebene wird aufgelöst. Unter Verwendung des vorgangsbasierten Modells lässt sich das *Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen (RCPSP/M)* formulieren, welches in Abschnitt 4.1 vorgestellt wird.

Um eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  dem integrierten Schedulingproblem zugänglich zu machen, ist eine *Transformation* erforderlich. Konkret handelt es sich dabei um einen Algorithmus, der die Struktur der Problemparameter umwandelt, sodass daraus eine Instanz des (RCPSP/M) entsteht – kurz als  $\mathcal{I}_{\text{RCPSP/M}}$  bezeichnet. Die Problemstellung soll bei der Umwandlung möglichst realitätsgetreu abgebildet werden, ohne die Lösbarkeit übermäßig zu beeinträchtigen. Unter Berücksichtigung dieses Spannungsfelds werden in Abschnitt 4.2 sogenannte Transformationsschemas vorgestellt, die die beschriebene Umwandlung der Problemparameter vornehmen. Als Herausforderung gilt hierbei, dass die vom Repetitionstyp einer Serienfertigung ähnelnde Stoffverarbeitung nicht ohne Weiteres geeignet dazu geeignet ist, mit Modellen aus der der Projektplanung abgebildet werden zu können (vgl. Abschnitt 2.4.2). So bestehen bei der Entwicklung der Transformationsschemas verschiedene Ausgestaltungsmöglichkeiten, deren jeweilige Vor- und Nachteile erörtert werden.

In Abschnitt 4.3 wird auf Charakteristika des (RCPSP/M) eingegangen, womit die im vorangegangenen Kapitel vorgenommene Abgrenzung zur Literatur vertieft wird. Abschnitt 4.4 befasst sich mit Aspekten, die bei der Modellierung einer realen Problemstellung zu beachten sind. Modellerweiterungen werden in Abschnitt 4.5 behandelt. Das Kapitel schließt mit einem Überblick und einer Zusammenfassung in Abschnitt 4.6.

## 4.1 Das Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen (RCPSP/M)

Wie die Problembezeichnung (RCPSP/M) bereits impliziert, wird auf die bekannten Modellbestandteile der RCPSP-Klasse zurückgegriffen. Dadurch wird die Anwendbarkeit gängiger Lösungsverfahren vereinfacht. Dennoch weist das (RCPSP/M) einige Besonderheiten gegenüber den aus der Literatur bekannten Problemen (RCPSP) und (RCPSP/c) auf. Deshalb ist die Bezeichnung mit dem Zusatz „M“ versehen, welcher für den englischsprachigen Begriff „material“ steht.

Das zentrale Konzept zur integrierten Planung von Rückbauprojekt und Stoffströmen ist die vorgangsbasierte Modellierung des Stoffstrombereichs. Dadurch können sämtliche aus der Literatur bekannten vorgangsbasierten Modellbestandteile, wie beispielsweise erneuerbare und kumulative Ressourcennutzungen, auf den Stoffstrombereich übertragen werden. Zusätzlich zu den bereits für das (RMCPSP) definierten Rückbauvorgängen  $i = 0, \dots, I + 1$ , zusammengefasst als Menge  $V$ , besteht das zu planende Projekt nun aus Stoffstromvorgängen  $j = 1^o, \dots, (J + 1)^o$ , zusammengefasst als Menge  $V^o$ . Das Superskript „o“ deutet dabei in Anlehnung an den von Schwindt und Trautmann (2000) und Neumann, Schwindt und Trautmann (2005) verwendeten Begriff „operation“ auf einen Zusammenhang zum Stoffstrombereich hin. Auf eine Verwendung des Superskripts „m“, welches als Initiale von „material“ geeigneter erscheinen mag, wird aufgrund einer möglichen Verwechslung mit der Notation für das Mehr-Modus-Schedulingproblem verzichtet. Um Rückbauvorgänge zu indizieren, werden wie bisher die Variablen  $i, i', i_1, i_2, \dots$  verwendet. Um Stoffstromvorgänge zu indizieren, werden die Variablen  $j, j', j_1, j_2, \dots$  verwendet. Stoffstromvorgänge sind genauso wie Rückbauvorgänge per Definition nichtunterbrechbar. Diese Vorgabe basiert auf der Modellspezifikation des (RCPSP) und ist Voraussetzung für die Anwendbarkeit gängiger Lösungsverfahren. Der fiktive Stoffstromvorgang  $(J + 1)^o$  bildet das Ende der Stoffverarbeitung ab. Demgegenüber existiert kein fiktiver Stoffstromvorgang, der den Start der Stoffverarbeitung abbildet. Das liegt daran, dass in Rückbauprojekten der Stoffstrombereich dem Rückbaubereich natürlicherweise nachgeordnet ist und somit jeder Stoffstromvorgang mindestens einen realen Rückbauvorgang als Vorgänger hat. Die Dauer jedes Stoffstromvorgangs  $j \in V^o$  wird mit  $d_j \in \mathbb{Z}_{\geq 0}$  angegeben, wobei  $d_{(J+1)^o} := 0$  gilt. Das (RCPSP/M) befasst sich mit der Suche nach einem Schedule  $(S, S^o)$ , wobei  $S = (S_i)_{i \in V}$  der Schedule der Rückbauvorgänge und  $S^o = (S_j^o)_{j \in V^o}$  der Schedule der Stoffstromvorgänge ist. Es wird analog zum (RMCPSP) die Zielsetzung verfolgt, die Dauer des Rückbauprojekts  $S_{I+1}$  zu minimieren.

Stoffstromvorgänge stehen in enger Verbindung zu den Verarbeitungsschritten, sind jedoch nicht mit ihnen zu verwechseln. Verarbeitungsschritte dienen der statischen Beschreibung von Stoffströmen in einer Instanz  $\mathcal{I}_{\text{RMCPSP}}$  und können innerhalb eines Stoffstrompfads oder über verschiedene Stoffstrompfade beliebig oft enthalten sein. In diesem Kapitel rückt jedoch die zeitliche Planung des Stoffstrombereichs in den Fokus, weshalb ein Übergang von einer statischen zur einer dynamischen Modellierung der Stoffströme erfolgt. Hierfür werden die Stoffstromvorgänge verwendet, welche die Verarbeitungsschritte um

eine zeitliche Dimension erweitern. Jeder Stoffstromvorgang existiert nur einmal in einer Instanz  $\bar{I}_{\text{RCPSP/M}}$  und ist mit einer bestimmten Stoffmenge und einer bestimmten Dauer assoziiert. Anders ausgedrückt, kann ein Verarbeitungsschritt als Klasse von Stoffstromvorgängen betrachtet werden und ein Stoffstromvorgang als konkrete Ausprägung eines Verarbeitungsschritts. Für den Anwendungsfall des kerntechnischen Rückbaus lässt sich der Unterschied zwischen Verarbeitungsschritten und Stoffstromvorgängen wie folgt verdeutlichen: Ein Verarbeitungsschritt beschreibt eine Technologie, z. B. „Sandstrahldekontamination“. Ein Stoffstromvorgang beschreibt die Anwendung einer Technologie auf eine bestimmte Stoffmenge, z. B. „Sandstrahldekontamination von zwei Tonnen Stahlrohren“. Zur Vereinfachung der Ausdrucksweise wird der Begriff des *Ableitens* verwendet. Der Stoffstromvorgang „Sandstrahldekontamination von zwei Tonnen Stahlrohren“ ist vom Verarbeitungsschritt „Sandstrahldekontamination“ abgeleitet. Falls eine explizite Darstellung dieser Beziehung benötigt wird, wird die Stoffstromvorgangsbezeichnung um  $[p]$  ergänzt, wobei  $p$  derjenige Verarbeitungsschritt ist, von dem der Stoffstromvorgang abgeleitet wurde. Beispielsweise bezeichnet  $1^o[P1]$  den Stoffstromvorgang  $1^o$ , der von Verarbeitungsschritt P1 abgeleitet wurde.

Nachfolgend wird die Formulierung des (RCPSP/M) hergeleitet. Die im (RMCPSP) definierten Vorrangrestriktionen (3.1) zwischen Rückbauvorgängen gelten im (RCPSP/M) unverändert.

Wie beim Vergleich mit existierenden Ansätzen in der Literatur in Abschnitt 3.5.1 hergeleitet, muss im (RCPSP/M) eine stoffstrominduzierte Zeitstruktur abgebildet werden. Dies wurde damit begründet, dass Stoffe individuelle Charakteristika aufweisen und deshalb inhomogen sind. Zunächst ist dafür eine feste Zuordnung der Stoffstromvorgänge zu den Rückbauvorgängen, die die jeweiligen Stoffe freisetzen, erforderlich. Diese Zuordnung wird mit einer Menge von *Freisetzungsbeziehungen* modelliert. Eine Freisetzungsbeziehung zwischen einem Rückbauvorgang  $i$  und einem Stoffstromvorgang  $j$  besteht aus einer Anordnungsbeziehung  $(i, j)$  und einem zeitlichen Mindestabstand  $d_{ij}^{rel} \in \mathbb{Z}_{\geq 0}$  (das Superskript „rel“ steht für „release“). Zeitliche Höchstabstände werden gemäß Modelleinschränkung (1) nicht berücksichtigt, weshalb der Zeitabstand auch nicht wie in Abschnitt 2.2.3 mit  $\delta$  bezeichnet wird. Die Menge aller Anordnungsbeziehungen ist  $E^{rel} \subseteq V \times V^o$ . Die zeitlichen Mindestabstände werden als Matrix  $d^{rel} := (d_{ij}^{rel})_{i \in V, j \in V^o}$  zusammengefasst. Die als Tupel  $(E^{rel}, d^{rel})$  gegebenen Freisetzungsbeziehungen implizieren *Freisetzungsrestriktionen* für das (RCPSP/M), die als Ungleichungen

$$S_j^o \geq S_i + d_{ij}^{rel} \quad (\forall (i, j) \in E^{rel}) \quad (4.1)$$

formuliert werden. Üblicherweise sind bei der Modellierung einer Problemstellung die zeitlichen Mindestabstände  $d_{ij}^{rel}$  so zu wählen, dass Stoffstromvorgang  $j$  starten kann, sobald die von ihm verarbeitete Stoffeinheit freigesetzt wurde. Allerdings sind auch Ausnahmen denkbar, wenn beispielsweise Transportzeiten oder Abklingzeiten einbezogen werden sollen.

Neben der durch die Freisetzungsbeziehungen sichergestellten Zuordnung der Stoffstromvorgänge zu den Rückbauvorgängen ist auch eine Zuordnung der Stoffstromvorgänge untereinander erforderlich, um die stoffstrominduzierte Zeitstruktur abzubilden. Diese

Zuordnung wird mit einer Menge von *stoffstrominduzierten Zeitbeziehungen* modelliert. Eine stoffstrominduzierte Zeitbeziehung zwischen zwei Stoffstromvorgängen  $j$  und  $j'$  besteht aus einer Anordnungsbeziehung  $(j, j')$  und einem zeitlichen Mindestabstand  $d_{jj'}^{flow} \in \mathbb{Z}_{\geq 0}$ . Zeitliche Höchstabstände werden gemäß Modelleinschränkung (1) auch hier nicht berücksichtigt. Die Menge aller Anordnungsbeziehungen ist  $E^{flow} \subset (V^o)^2$ . Die zeitlichen Mindestabstände werden als Matrix  $d^{flow} := (d_{jj'}^{flow})_{j,j' \in V^o, j \neq j'}$  zusammengefasst. Die als Tupel  $(E^{flow}, d^{flow})$  gegebenen stoffstrominduzierten Zeitbeziehungen implizieren *stoffstrominduzierte Zeitrestriktionen* für das (RCPSP/M), die als Ungleichungen

$$S_{j'}^o \geq S_j^o + d_{jj'}^{flow} \quad (\forall (j, j') \in E^{flow}) \quad (4.2)$$

formuliert werden. Stoffstrominduzierte Zeitbeziehungen sind für eine Instanz  $\mathcal{I}_{RCPSP/M}$  so zu erzeugen, dass die Struktur eines Stoffstroms auf die Stoffstromvorgänge übertragen wird. Üblicherweise sind die zeitlichen Mindestabstände so zu wählen, dass der nachfolgende Stoffstromvorgang  $j'$  starten kann, sobald die von ihm verarbeitete Stoffeinheit vom vorangehenden Stoffstromvorgang  $j$  zur Verfügung gestellt wurde. Um sicherzustellen, dass der fiktive Stoffstromvorgang  $(J+1)^o$  das Ende der Stoffverarbeitungen darstellt, müssen außerdem geeignete fiktive stoffstrominduzierte Zeitbeziehungen eingeführt werden.

Die Zeitstruktur einer Instanz  $\mathcal{I}_{RCPSP/M}$  lässt sich in Form eines Vorgangsknotennetzplans  $N^{temp}$  darstellen. Jeder Vorgang  $i \in V$  und jeder Stoffstromvorgang  $j \in V^o$  wird als Knoten abgebildet. Jede Vorrangbeziehung  $(i, i') \in E$  wird als mit  $d_i$  bewertete Kante abgebildet. Jede Anordnungsbeziehung  $(i, j) \in E^{rel}$  und  $(j, j') \in E^{flow}$  wird als mit  $d_{ij}^{rel}$  bzw.  $d_{jj'}^{flow}$  bewertete Kante abgebildet. Zyklen sind nicht zulässig, da sie eine positive Länge hätten und die Problem Instanz dadurch unzulässig wäre (vgl. Abschnitt 2.2.3). Zusätzlich zu den bereits in (2.1) und (2.2) definierten Mengen  $Pred(i)$  und  $Succ(i)$  lassen sich folgende direkte Vorgänger- und Nachfolgermengen definieren:

$$Succ^o(i) := \{j \in V^o \mid \exists (i, j) \in E^{rel}\}, \quad (4.3)$$

$$Pred(j) := \{i \in V \mid \exists (i, j) \in E^{rel}\}, \quad (4.4)$$

$$Pred^o(j) := \{j' \in V^o \mid \exists (j', j) \in E^{flow}\} \text{ und} \quad (4.5)$$

$$Succ^o(j) := \{j' \in V^o \mid \exists (j, j') \in E^{flow}\}. \quad (4.6)$$

Das Superskript „o“ verdeutlicht, dass es sich um eine Teilmenge der Stoffstromvorgänge handelt. Ohne das Superskript handelt es sich um eine Teilmenge der Rückbauvorgänge. Eine Menge  $Pred^o(i)$  existiert nicht, da Rückbauvorgänge keine Stoffstromvorgänge als Vorgänger haben können.

Darüber hinaus lassen sich auch transitive Vorgänger- und Nachfolgermengen definieren. Dabei sind

$$\overline{Succ^o}(i) := \{j \in V^o \setminus \{(J+1)^o\} \mid \exists \text{ ein Pfad von } i \text{ nach } j \text{ in } N^{temp}\} \text{ und} \quad (4.7)$$

$$\overline{Succ^o}(j) := \{j' \in V^o \setminus \{(J+1)^o\} \mid \exists \text{ ein Pfad von } j \text{ nach } j' \text{ in } N^{temp}\} \quad (4.8)$$

von besonderer Bedeutung. Diese Menge können auch nach der in Gleichung (2.4) beschriebenen Logik rekursiv berechnet werden. Zu beachten ist, dass der fiktive Stoffstromvorgang

$(J + 1)^o$  explizit aus  $\overline{Succ^o}(i)$  bzw.  $\overline{Succ^o}(j)$  ausgeschlossen ist, um die algorithmischen Formulierungen in Kapitel 5 zu vereinfachen.

Die vorgangsbasierte Modellierung des Stoffstrombereichs ermöglicht es, die in Abschnitt 3.4.2 durch die Funktion  $\Phi_k^\alpha(t)$  abstrakt ausgedrückte erneuerbare Ressourcennutzung im Stoffstrombereich zu konkretisieren. Dazu werden für die Stoffstromvorgänge die aktive Menge

$$\mathcal{A}^\alpha(S^o, t) := \{j \in V^o \mid S_j^o \leq t < S_j^o + d_j\} \quad (4.9)$$

und die Funktion

$$r_k^\alpha(S^o, t) := \sum_{j \in \mathcal{A}^\alpha(S^o, t)} r_{jk}^\alpha \quad (4.10)$$

definiert. Beide Definitionen sind analog zur Formulierung der erneuerbaren Ressourcennutzungen durch Rückbauvorgänge. Auf ein Superskript „o“ in der Notation wird verzichtet, da das Funktionsargument  $S^o$  bereits eindeutig den Zusammenhang zum Stoffstrombereich impliziert. Unter Zusammenführung der Ressourcennutzungen im Rückbau- und Stoffstrombereich werden die erneuerbaren Ressourcenrestriktionen als Ungleichungen

$$r_k^\alpha(S, t) + r_k^\alpha(S^o, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0) \quad (4.11)$$

formuliert.

Die Lager im (RMCPSP) werden im (RCPSM) als kumulative Ressourcen modelliert. In Abschnitt 2.2.6 wurde erläutert, dass für kumulative Ressourcen grundsätzlich die Angabe eines Höchst- und eines Mindestbestands möglich ist. Die stoffstrominduzierte Zeitstruktur stellt jedoch bereits sicher, dass Stoffe nicht aus einem Lager entnommen werden können, bevor sie zur Verfügung stehen. Deshalb wird bei korrekt erzeugter stoffstrominduzierter Zeitstruktur implizit ein Mindestbestand von  $\underline{R}_k^Y = 0$  für alle kumulativen Ressourcen  $k \in \mathcal{R}^Y$  eingehalten. Darüber hinaus ist entsprechend Modelleinschränkung (3) kein echter Mindestbestand  $\underline{R}_k^Y > 0$  zu berücksichtigen. Somit ist es ausreichend, für jede kumulative Ressource  $k \in \mathcal{R}^Y$  die Lagerkapazität  $R_k^Y \in \mathbb{Z}$  als kumulative Ressourcenkapazität (= Höchstbestand) zu betrachten. Auf die Überstreichung in der Notation (d. h.  $\overline{R}_k^Y$ ) wird verzichtet.

Die mit den Stoffströmen einhergehenden Lagerbewegungen werden für jeden Stoffstromvorgang  $j \in V^o$  und jede kumulative Ressource  $k \in \mathcal{R}^Y$  über die kumulative Ressourcennutzung  $r_{jk}^Y$  beschrieben. Um für Stoffstromvorgänge zwischen den Nutzungsarten unterscheiden zu können, wird  $r_{jk}^Y$  durch den dreidimensionalen Vektor  $(r_{jk}^{Y,start}, r_{jk}^{Y,lin}, r_{jk}^{Y,end})$  ersetzt. Dabei bezeichnet  $r_{jk}^{Y,start} \in \mathbb{Z}$  die Ressourcennutzung zum Start von  $j$ ,  $r_{jk}^{Y,lin} \in \mathbb{Z}$  die gleichmäßige Ressourcennutzung während der Ausführung von  $j$  und  $r_{jk}^{Y,end} \in \mathbb{Z}$  die Ressourcennutzung zum Ende von  $j$ . Diese Problemparameter sind für eine Instanz  $\overline{\mathcal{I}}_{RCPSM}$  so zu wählen, dass sie den zeitlichen Verlauf der Stoffströme realitätsgetreu abbilden.

Wenn beispielsweise Stoffe gleichmäßig während der Vorgangsausführung verarbeitet und an ein Lager  $k$  weitergereicht werden, müssen die entsprechenden Stoffstromvorgänge die kumulative Ressource  $k$  gleichmäßig über die Vorgangsausführung nutzen. Dabei ist zu beachten, dass die stoffstrominduzierte Zeitstruktur und die kumulativen Ressourcennutzungen in Einklang miteinander stehen sollten. Die Auslagerung durch einen nachfolgenden Stoffstromvorgang darf erst erfolgen, wenn die Einlagerung durch den vorangehenden Stoffstromvorgang bzw. die Freisetzung der Stoffe erfolgt ist. Zum besseren Verständnis dieses Zusammenhangs wird auf die in den nachfolgenden Abschnitten vorgestellten Transformationsschemas verwiesen.

Die Anzahl der durch einen Stoffstromvorgang  $j \in V^o$  mit gegebenem Startzeitpunkt  $S_j^o$  zu einem Zeitpunkt  $t \geq S_j^o$  genutzten Einheiten einer kumulativen Ressource  $k \in \mathcal{R}^V$  ergibt sich zu

$$r_{jk}^Y(S_j^o, t) := \begin{cases} r_{jk}^{Y,start} & \text{falls } t = S_j^o, \\ r_{jk}^{Y,start} + \left\lfloor (t - S_j^o) r_{jk}^{Y,lin} / d_j \right\rfloor & \text{falls } r_{jk}^{Y,lin} \geq 0 \wedge S_j^o < t < S_j^o + d_j, \\ r_{jk}^{Y,start} + \left\lfloor (t + 1 - S_j^o) r_{jk}^{Y,lin} / d_j \right\rfloor & \text{falls } r_{jk}^{Y,lin} < 0 \wedge S_j^o < t < S_j^o + d_j, \\ r_{jk}^{Y,start} + r_{ik}^{Y,lin} + r_{jk}^{Y,end} & \text{falls } t \geq S_j^o + d_j. \end{cases} \quad (4.12)$$

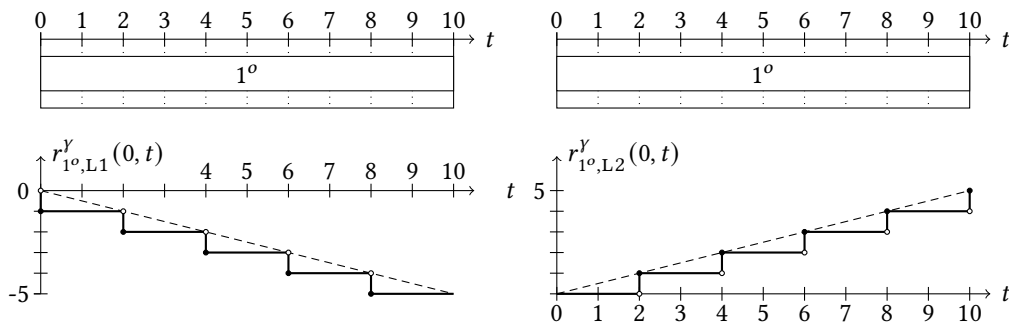
Für die gleichmäßige Nutzung während der Ausführung des Stoffstromvorgangs  $j$  wird hier unterschieden, ob  $r_{jk}^{Y,lin}$  positiv oder negativ ist, d. h. ob  $j$  in  $k$  ein- oder auslagert. Diese Unterscheidung resultiert aus der Diskretisierung der Stoffmengen. Wenn Stoffeinheiten eingelagert werden, wird davon ausgegangen, dass sie erst als eingelagert gelten, sobald ihre Verarbeitung vollständig abgeschlossen ist. Wenn Stoffeinheiten ausgelagert werden, wird hingegen davon ausgegangen, dass sie sofort als ausgelagert gelten, sobald ihre Verarbeitung beginnt. Deshalb ist der Zustand zum Zeitpunkt  $(t + 1 - S_j^o)$  ausschlaggebend. Zu beachten ist, dass die Abrundungsfunktion immer die nächstkleinere ganze Zahl liefert. Das bedeutet, der Betrag eines negativen Wertes wird aufgerundet. Beispiel 4.1 veranschaulicht, wie Gleichung (4.12) zustande kommt.

**Beispiel 4.1:** Ein Stoffstromvorgang  $1^o$  mit Dauer  $d_{1^o} = 10$  lagert gleichmäßig fünf Stoffeinheiten aus einer kumulativen Ressource  $L1^1$  aus und gleichzeitig fünf Stoffeinheiten in eine kumulative Ressource  $L2$  ein. Es gilt somit  $r_{1^o,L1}^{Y,lin} = -5$  und  $r_{1^o,L2}^{Y,lin} = 5$ . Abbildung 4.1 zeigt den Verlauf der in Gleichung (4.12) definierten Funktionen  $r_{1^o,L1}^Y(0, t)$  und  $r_{1^o,L2}^Y(0, t)$ . Für  $t = 1$  gilt beispielsweise  $r_{1^o,L1}^Y(0, 1) := \lfloor (1 + 1 - 0) \cdot (-5)/10 \rfloor = -1$  und  $r_{1^o,L2}^Y(0, 1) := \lfloor (1 - 0) \cdot 5/10 \rfloor = 0$ .

Zum Vergleich ist in Abbildung 4.1 mit gestrichelten Linien eingezeichnet, welcher Verlauf der Funktionen bei kontinuierlichen Stoffeinheiten zu beobachten wäre (= linearer Verlauf). Es ist erkennbar, dass die Unterscheidung zwischen positivem und negativem  $r_{jk}^{Y,lin}$  in Gleichung (4.12) zu einer Treppenfunktion führt, die sowohl für Ein- als auch Auslagerungen von unten an den linearen Verlauf angelegt ist.  $\square$

<sup>1</sup> Exemplarische kumulative Ressourcen werden fortan genau wie Lager mit  $Lx$  bezeichnet. Damit wird die modelltheoretische Identität von Lagern und kumulativen Ressourcen ausgedrückt.





**Abbildung 4.1:** Gantt-Diagramme und Verlauf der kumulativen Ressourcennutzungen zu Beispiel 4.1.

Die gesamte Anzahl der durch Stoffstromvorgang  $j$  genutzten Einheiten einer kumulativen Ressource  $k$  wird als  $r_{jk}^Y(\cdot, \infty)$  bezeichnet. Da dieser Wert unabhängig vom Startzeitpunkt  $S_j^o$  ist, wird in der Notation ein  $\cdot$  verwendet. Ein Stoffstromvorgang  $j$  gilt bezüglich einer kumulativen Ressource  $k$  als *einlagernd*, falls  $r_{jk}^Y(\cdot, \infty) > 0$ . Umgekehrt gilt er als *auslagernd*, falls  $r_{jk}^Y(\cdot, \infty) < 0$ . Falls  $r_{jk}^Y(\cdot, \infty) = 0$ , gilt  $j$  als *neutral*. Dies umfasst zwei Fälle: Entweder Stoffstromvorgang  $j$  nutzt die kumulative Ressource  $k$  nicht, d. h., er lagert weder ein noch aus. Oder Stoffstromvorgang  $j$  lagert dieselbe Stoffmenge aus einer kumulativen Ressource  $k$  aus, die er auch wieder in  $k$  einlagert. Letzterer Fall wird ausführlich in Abschnitt 4.4.1 untersucht.

Die vorgangsbasierte Modellierung des Stoffstrombereichs ermöglicht es nun, die in Abschnitt 3.4.2 durch die Funktion  $\Gamma_k^Y(t)$  abstrakt ausgedrückte verfügbare Lagerkapazität im Stoffstrombereich zu konkretisieren. Dazu wird für die Stoffstromvorgänge die aktive Menge bezüglich der kumulativen Ressourcen

$$\mathcal{A}^Y(S^o, t) := \{j \in V^o \mid 0 \leq S_j^o \leq t\} \quad (4.13)$$

definiert. Die Anzahl der durch alle Stoffstromvorgänge zu einem Zeitpunkt  $t$  genutzten Einheiten einer kumulativen Ressource  $k \in \mathcal{R}^Y$  ergibt sich bei gegebenem Schedule  $S^o$  zu

$$r_k^Y(S^o, t) := \sum_{j \in \mathcal{A}^Y(S^o, t)} r_{jk}^Y(S_j^o, t). \quad (4.14)$$

Unter Zusammenführung der Stofffreisetzungen im Rückbaubereich und der kumulativen Ressourcennutzungen im Stoffstrombereich werden die Lagerrestriktionen (3.7) nun ersetzt durch kumulative Ressourcenrestriktionen

$$f_k^Y(S, t) + r_k^Y(S^o, t) \leq R_k^Y \quad (\forall k \in \mathcal{R}^Y, t \geq 0). \quad (4.15)$$

Die Verwendung der unterschiedlichen Symbole  $f$  und  $r$  in den Notationen verdeutlicht, dass sich Rückbau- und Stoffstromvorgänge hinsichtlich der kumulativen Ressourcen unterscheiden. Rückbauvorgänge können nur freisetzen, d. h. kumulative Ressourcen positiv nutzen. Stoffstromvorgänge können ein- und auslagern, d. h. kumulative Ressourcen

positiv und negativ nutzen. Um diese Unterscheidung auch sprachlich hervorzuheben, wird bei Rückbauvorgängen stets von Freisetzungen anstelle von Nutzungen gesprochen.

Für einen gegebenen Schedule  $(S, S^o)$  und alle Zeitpunkte  $t \geq 0$  beschreibt die linke Seite der kumulativen Ressourcenrestriktionen (4.15) das *kumulative Ressourcenprofil* der kumulativen Ressource  $k$ . Aufgrund der modelltheoretischen Identität von Lagern und kumulativen Ressourcen entspricht dies dem Auslastungsprofil des Lagers  $k$ . Nachfolgend wird für die Beschreibung realer Problemstellungen weiterhin von Lagern gesprochen, wohingegen bei der Modellierung und Lösung im Zusammenhang mit dem (RCPSP/M) der Begriff der kumulativen Ressource verwendet wird.

Die formale Beschreibung des (RCPSP/M) lautet zusammengefasst:

$$\begin{array}{ll}
 \min_{(S, S^o)} & z(S, S^o) := S_{I+1} \\
 \text{u. d. R.} & S_{i'} \geq S_i + d_i \quad (\forall (i, i') \in E); \\
 & S_j^o \geq S_i + d_{ij}^{rel} \quad (\forall (i, j) \in E^{rel}); \\
 & S_{j'}^o \geq S_j^o + d_{jj'}^{flow} \quad (\forall (j, j') \in E^{flow}); \\
 & r_k^\alpha(S, t) + r_k^\alpha(S^o, t) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t \geq 0); \\
 & f_k^\gamma(S, t) + r_k^\gamma(S^o, t) \leq R_k^\gamma \quad (\forall k \in \mathcal{R}^\gamma, t \geq 0); \\
 & S_0 = 0; \\
 & S_i \geq 0 \quad (i = 1, \dots, I + 1); \\
 & S_j^o \geq 0 \quad (j = 1^o, \dots, (J + 1)^o).
 \end{array} \quad \left. \vphantom{\begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}} \right\} \text{(RCPSP/M)}$$

Tabelle 4.1 listet die zur Formulierung des (RCPSP/M) eingeführte Notation auf, sofern sie nicht bereits in Tabelle 3.1 enthalten ist.

Beim Vergleich der beiden Problemformulierungen ist ersichtlich, dass das (RCPSP/c) und das (RCPSP/M) zahlreiche Überschneidungen aufweisen. Genauer ausgedrückt, kann das (RCPSP/M) als eine Spezifizierung des (RCPSP/c) betrachtet werden, welche auf die stoffstrombasierte Planung eines Rückbauprojekts angepasst wurde. So erlaubt die Formulierung des (RCPSP/M) lediglich Zeitstrukturen, die von den Rückbauvorgängen in Richtung der Stoffstromvorgänge ausgerichtet sind und nicht umgekehrt. Das Problem der integrierten Planung von Rückbauprojekt und Stoffströmen hätte auch als (RCPSP/c) formuliert werden können (falls Zeitrestriktionen statt den Vorrangrestriktionen verwendet werden). Die mit dem (RCPSP/M) vorgenommene Spezifizierung fördert jedoch das Verständnis der Problemstruktur. Aus diesem Grund werden auch Stoffstromvorgänge getrennt von Rückbauvorgängen modelliert, obwohl diese Unterscheidung modelltheoretisch nicht notwendig wäre. Genauso wäre die Unterscheidung zwischen Vorrangbeziehungen, Freisetzungsbefziehungen und stoffstrominduzierten Zeitbeziehungen nicht notwendig. Diese auf den Anwendungsfall zugeschnittenen Konzepte dienen dazu, die weiteren Darlegungen intuitiver zu gestalten.

Notation	Bedeutung
<b>Problemparameter (modellexogen)</b>	
$j \in V^0$	Stoffstromvorgang
$d_j \in \mathbb{Z}_{\geq 0}$	Dauer des Stoffstromvorgangs $j$
$(i, j) \in E^{rel}$	Anordnungsbeziehung zur Freisetzungsbeziehung
$d_{ij}^{rel} \in \mathbb{Z}_{\geq 0}$	zeitlicher Mindestabstand zwischen dem Start von $i$ und $j$ zur Freisetzungsbeziehung
$(j, j') \in E^{flow}$	Anordnungsbeziehung zur stoffstrominduzierten Zeitbeziehung
$d_{ij}^{flow} \in \mathbb{Z}_{\geq 0}$	zeitlicher Mindestabstand zwischen dem Start von $j$ und $j'$ zur stoffstrominduzierten Zeitbeziehung
$r_{jk}^\alpha \in \mathbb{Z}_{\geq 0}$	Nutzung der erneuerbaren Ressource $k \in \mathcal{R}^\alpha$ durch Stoffstromvorgang $j$
$k \in \mathcal{R}^Y$	kumulative Ressource
$R_k^Y \in \mathbb{Z}_{\geq 0}$	Kapazität (= Höchstbestand) der kumulativen Ressource $k \in \mathcal{R}^Y$
$r_{jk}^{Y,start} \in \mathbb{Z}$	Nutzung der kumulativen Ressource $k \in \mathcal{R}^Y$ zum Start, ...
$r_{jk}^{Y,lin} \in \mathbb{Z}$	... gleichmäßig während der Ausführung bzw. ...
$r_{jk}^{Y,end} \in \mathbb{Z}$	... zum Ende des Stoffstromvorgangs $j$
<b>Entscheidungsvariablen und Funktionen (modellendogen)</b>	
$S_j^0$	Startzeitpunkt des Stoffstromvorgangs $j$
$r_k^\alpha(S^0, t)$	Anzahl der durch alle Stoffstromvorgänge zu einem Zeitpunkt $t$ genutzten Einheiten der erneuerbaren Ressource $k \in \mathcal{R}^\alpha$
$r_{jk}^Y(S_j^0, t)$	Anzahl der durch Stoffstromvorgang $j$ zu einem Zeitpunkt $t$ genutzten Einheiten der kumulativen Ressource $k \in \mathcal{R}^Y$
$r_k^Y(S^0, t)$	Anzahl der durch alle Stoffstromvorgänge zu einem Zeitpunkt $t$ genutzten Einheiten der kumulativen Ressource $k \in \mathcal{R}^Y$

**Tabelle 4.1:** Eingeführte Notation zur Formulierung des (RCSP/M).

## 4.2 Transformationsschemas

In einer Instanz  $\mathcal{I}_{\text{RMCPSP}}$  sind Stoffströme über Stoffstrompfade und Stofffreisetzungen beschrieben (vgl. Abschnitt 3.4.1). Um das Rückbauprojekt und Stoffströme integriert planen zu können, werden Stoffstromvorgänge benötigt, die die Lösungsmenge des Schedulingproblems um die zeitliche Dimension der Stoffströme (d. h. die Startzeitpunkte  $S^0$ ) erweitern. Für die Stoffstromvorgänge muss festgelegt werden, wie lange sie dauern, welche erneuerbare Ressourcen sie in welchem Umfang nutzen, welche kumulative Ressourcen sie in welchem Umfang und nach welcher Nutzungsart nutzen und wie sie sich über die stoffstrominduzierte Zeitstruktur in das Projekt eingliedern. Zur systematischen Belegung dieser Problemparameter dienen *Transformationsschemas*. Sie wandeln eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  in eine Instanz  $\mathcal{I}_{\text{RCSP/M}}$  um. Ein Transformationsschema ist so zu gestalten, dass die im vorangegangenen Abschnitt 4.1 genannten Anforderungen an eine Instanz  $\mathcal{I}_{\text{RCSP/M}}$  eingehalten werden. Beispielsweise muss sichergestellt sein, dass die stoffstrominduzierte Zeitstruktur und die kumulativen Ressourcennutzungen korrekt aufeinander abgestimmt

sind und der Anwendungsfall damit realitätsgetreu modelliert wird. Ein Transformationsschema ist nicht als Bestandteil eines Lösungsverfahrens zu interpretieren, sondern als Hilfsmittel bei der Modellierung des dem produktionswirtschaftlichen Kontext zuordenbaren Stoffstrombereichs mit Konzepten des Projektschulings. Es ist durchaus möglich, die Problemstellung auch ohne Verwendung eines Transformationsschemas direkt als Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  zu modellieren, was bei großen Stoffmengen jedoch mit hohem Aufwand verbunden sein kann.

In Kapitel 3 wurde die Problemstellung nicht ausreichend konkret beschrieben, um eindeutig festlegen zu können, wie der Stoffstrombereich modelliert wird. Anders ausgedrückt gibt es mehrere Möglichkeiten, eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  in eine Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  umzuwandeln. Deshalb gibt es mehrere Ausgestaltungsmöglichkeiten für ein Transformationsschema. Diese Flexibilität bei der Ausgestaltung des Transformationsschemas ist beabsichtigt. Nur eingehende Kenntnisse der praktischen Entscheidungssituation geben Aufschluss darüber, wie Stoffströme im Detail zu modellieren sind. Je nach Anwendung können sich Unterschiede ergeben. Außerdem kann die Modellierung des Stoffstrombereichs Auswirkungen auf die Leistungsfähigkeit der in Kapitel 5 vorgestellten Lösungsverfahren haben. Somit würde es dem Ziel dieser Arbeit, ein für möglichst viele Entscheidungssituationen geeignetes Optimierungswerkzeug zu entwickeln, widersprechen, wenn das (RMCPSP) stärker konkretisiert worden wäre. Da nachfolgend verschiedene Ausgestaltungsmöglichkeiten der Transformationsschemas untersucht werden, ist dennoch sichergestellt, dass ein Großteil der Anwendungsfälle abgedeckt wird – entweder, indem ein vorgeschlagenes Transformationsschema direkt übernommen wird, oder indem es als Ansatzpunkt für eine Weiterentwicklung herangezogen wird.

Die Ausgestaltungsmöglichkeiten für das Transformationsschema definieren sich im Wesentlichen über die Logik, nach welcher Stoffstromvorgänge erzeugt werden. Zwei Varianten für eine solche Logik werden in den Abschnitten 4.2.1 und 4.2.2 vorgestellt und in den Abschnitten 4.2.3 und 4.2.4 tiefergehend untersucht. Die beiden Varianten bewegen sich in einem Spannungsfeld aus Größe und Realitätstreue der erzeugten Instanz, wie gezeigt werden wird. In Abschnitt 4.2.5 wird ein Überblick über weitere Ausgestaltungsmöglichkeiten für Transformationsschemas gegeben.

#### 4.2.1 Granulares Transformationsschema

Der Grundgedanke des *granularen Transformationsschemas* ist es, die in einer Instanz  $\mathcal{I}_{\text{RMCPSP}}$  über Stoffstrompfade und Stofffreisetzungen beschriebenen Stoffströme auf Basis der kleinsten Einheit zu modellieren. Eine solche Ausgestaltungsmöglichkeit für das Transformationsschema ist dadurch motiviert, dass die Stoffmengen diskret sind (vgl. Abschnitt 3.4.1). Von jedem Verarbeitungsschritt, der auf jede einzelne Stoffeinheit angewendet wird, wird daher ein *granularer Stoffstromvorgang* abgeleitet. Dies ist mit dem Vorteil verbunden, dass jede Stoffeinheit zeitlich unabhängig von anderen Stoffeinheiten eingeplant werden kann und das Modell eine hohe Planungsflexibilität ermöglicht. Voraussetzung ist jedoch, dass auch der modellierte Anwendungsfall eine solche Planungsflexibilität zulässt. Falls

es bei der Stoffverarbeitung Mindestmengen gibt, für die beispielsweise eine Maschine angefahren wird, ist das granulare Transformationsschema entsprechend anzupassen. Nachfolgend wird davon ausgegangen, dass die Stoffverarbeitung für jede Stoffeinheit flexibel gestartet und unterbrochen werden kann. Grundsätzlich erscheint diese Annahme plausibel, da der Stoffstrombereich in dieser Arbeit nur sekundär geplant und aus übergeordneter Perspektive betrachtet wird. Über die exakte zeitliche Ausgestaltung der Stoffverarbeitung wird im Rahmen der operativen Maschinenbelegungsplanung entschieden, wo dann auch Stoffstromvorgänge gruppiert oder verschoben werden können. Für die in dieser Arbeit betrachtete stoffstrombasierte Planung eines Rückbauprojekts ist es lediglich von Bedeutung, die Rückkopplungen der Engpässe im Stoffstrombereich auf das Rückbauprojekt abzubilden.

Eine mit dem granularen Transformationsschema erzeugte Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  weist eine spezielle Struktur auf (d. h., die Belegung der Problemparameter folgt einem bestimmten Muster). Um diese spezielle Struktur in der Notation hervorzuheben, wird von einer Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  gesprochen. Das granulare Transformationsschema ist formal in Algorithmus 2 beschrieben und wird als `TransformationSchemeGra` bezeichnet. Die Zeilenreferenzen in diesem Abschnitt beziehen sich auf Algorithmus 2. Zunächst wird das granulare Transformationsschema verbal erläutert. Am Ende dieses Abschnitts wird an die in Beispiel 3.1 beschriebene Instanz  $\mathcal{I}_{\text{RMCPSP}}$  angeknüpft und gezeigt, wie die Erzeugung der dazugehörigen Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  abläuft.

`TransformationSchemeGra` bekommt eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  als Eingabe übergeben. Im (RMCPSP) ist der Stoffstrombereich nur abstrakt modelliert, weshalb in  $\mathcal{I}_{\text{RMCPSP}}$  keine Informationen zu den Dauern der Stoffverarbeitung und zu den erneuerbaren Ressourcennutzungen im Stoffstrombereich vorhanden sind. Da Verarbeitungsschritte als Klasse von Stoffstromvorgängen betrachtet werden, wird angenommen, dass alle von einem Verarbeitungsschritt  $p \in \mathcal{P}$  abgeleiteten granularen Stoffstromvorgänge dieselbe Dauer  $d_p \in \mathbb{Z}_{\geq 0}$  und dieselben erneuerbaren Ressourcennutzungen  $r_{pk}^\alpha \in \mathbb{Z}_{\geq 0}$  aufweisen. Diese Annahme vereinfacht die systematische Umwandlung der Instanzen. Bei  $d_p$  und  $r_{pk}^\alpha$  handelt es sich ebenfalls um eine Eingabe für `TransformationSchemeGra`. Es gibt viele weitere Ausgestaltungsmöglichkeiten zur Festlegung der Dauern und erneuerbaren Ressourcennutzungen der Stoffstromvorgänge. Von Projektverantwortlichen können diese Problemparameter auch individuell vergeben werden.

In Zeile 1 wird die neue Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  erzeugt, die im weiteren Verlauf von `TransformationSchemeGra` zusammengestellt wird. In Zeile 2 wird der fiktive Stoffstromvorgang  $(J+1)^o$ , der das Ende der Stoffverarbeitung abbildet, erzeugt. Ab Zeile 3 werden die Lager als kumulative Ressourcen betrachtet. Es sei erneut darauf hingewiesen, dass kumulative Ressourcen und Lager aus modelltheoretischer Sicht identisch sind.

In den darauffolgenden Schleifen wird über jede Stoffeinheit  $u = 1, \dots, f_{iw}^{\gamma, \text{start}} + f_{iw}^{\gamma, \text{lin}} + f_{iw}^{\gamma, \text{end}}$ , die jeder Rückbauvorgang  $i \in V$  in jeden Stoffstrompfad  $w \in \mathcal{W}$  freisetzt, iteriert. In Zeile 7 wird die Hilfsvariable `step` initialisiert. Dabei handelt es sich um einen Zähler für Verarbeitungsschritte  $p_1(w), p_2(w), \dots$  im Stoffstrompfad  $w$ . Dieser Zähler wird in Zeile 19 schrittweise erhöht, bis die Senke  $s$  des Stoffstrompfads  $w$  erreicht ist und damit

---

**Algorithmus 2:** TransformationSchemeGra

---

**Eingabe:** Instanz  $\mathcal{I}_{\text{RMCPSP}}$ ; Dauern  $d_p$  und ern. Ressourcennutzungen  $r_{pk}^\alpha$  für alle  $p \in \mathcal{P}$  und  $k \in \mathcal{R}^\alpha$

```

1 erzeuge eine neue Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$ ;
2 erzeuge den fiktiven Stoffstromvorgang  $(J+1)^o$ ;
3 betrachte jedes Lager  $k \in \mathcal{R}^Y$  als kumulative Ressource;
4 für Rückbauvorgang  $i := 1$  bis  $I$  tue
5   für jeden Stoffstrompfad  $w \in W$  tue
6     für Stoffeinheit  $u := 1$  bis  $f_{iw}^{Y,start} + f_{iw}^{Y,lin} + f_{iw}^{Y,end}$  tue
7       initialisiere  $step := 1$ ;
8       solange Senke  $s$  nicht erreicht tue
9         leite einen neuen Stoffstromvorgang  $j'$  von  $p_{step}(w)$  ab;
10        setze  $d_{j'} := d_{p_{step}(w)}$ ;
11        setze  $r_{j'k}^\alpha := r_{p_{step}(w),k}^\alpha$  für jedes  $k \in \mathcal{R}^\alpha$ ;
12        wenn  $step = 1$  dann
13          erzeuge  $(i, j') \in E^{rel}$  und setze  $d_{ij'}^{rel}$  gemäß (4.16);
14        sonst
15          erzeuge  $(j, j') \in E^{flow}$  und setze  $d_{jj'}^{flow} := d_j$ ;
16          setze  $r_{j,k_{step}(w)}^{Y,end} := 1$ ;
17        Ende
18        setze  $r_{j',k_{step}(w)}^{Y,start} := -1$ ;
19        setze  $j := j'$  und  $step := step + 1$ ;
20      Ende
21    erzeuge  $(j, (J+1)^o) \in E^{flow}$  und setze  $d_{j,(J+1)^o}^{flow} := d_j$ ;
22  Ende
23 Ende
24 Ende
25 vervollständige  $\mathcal{I}_{\text{RCPSP/M-gra}}$  mit allen übrigen Problemparametern aus  $\mathcal{I}_{\text{RMCPSP}}$ ;
Ausgabe: Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$ 

```

---

über alle Verarbeitungsschritte in  $w$  iteriert wurde. In Zeile 9 wird vom betrachteten Verarbeitungsschritt  $p_{step}(w)$  ein neuer granularer Stoffstromvorgang  $j'$  abgeleitet. Dabei ist  $j'$  eine Hilfsvariable für den jeweils neu erzeugten granularen Stoffstromvorgang. Die Dauer und die erneuerbaren Ressourcennutzungen werden, wie in den Zeilen 10 und 11 beschrieben, von Verarbeitungsschritt  $p_{step}(w)$  an  $j'$  übergeben.

In Zeile 12 erfolgt eine Fallunterscheidung: Falls der erste Verarbeitungsschritt im Stoffstrompfad  $w$  betrachtet wird, ist die Erzeugung einer Freisetzungsbeziehung erforderlich. Dazu wird in Zeile 13 zunächst die Anordnungsbeziehung  $(i, j')$  zwischen dem Rückbauvorgang  $i$  und dem neu erzeugten Stoffstromvorgang  $j'$  zu  $E^{rel}$  hinzugefügt. Der zeitliche Mindestabstand  $d_{ij'}^{rel}$  wird so gewählt, dass Stoffstromvorgang  $j'$  starten kann, sobald die von ihm verarbeitete Stoffeinheit freigesetzt wurde. Somit ist  $d_{ij'}^{rel}$  von der zeitlichen Verteilung der Stofffreisetzungen abhängig. Formal gilt deshalb in Zeile 13

$$d_{ij'}^{rel} := \min \{ \tau \in \mathbb{Z}_{\geq 0} \mid f_{iw}^y(0, \tau) \geq u \}. \quad (4.16)$$

Dabei ist  $f_{iw}^y(0, \tau)$  die Anzahl der durch Rückbauvorgang  $i$  zum Zeitpunkt ( $\tau$  Perioden nach dem Start von  $i$ ) freigesetzten Stoffeinheiten, die Stoffstrompfad  $w$  durchlaufen. Sie muss mindestens so groß sein wie  $u$ . Der Minimumsausdruck stellt sicher, dass der zeitliche Mindestabstand nur gerade so groß wie erforderlich gewählt wird. Falls alle Stoffeinheiten von  $i$  zu seinem Start freigegeben werden, liefert Gleichung (4.16) immer  $d_{ij'}^{rel} := 0$ . Falls alle Stoffeinheiten von  $i$  zu seinem Ende freigegeben werden, liefert Gleichung (4.16) immer  $d_{ij'}^{rel} := d_i$ .

Falls die in Zeile 12 formulierte Bedingung nicht zutrifft, d. h. nicht der erste Verarbeitungsschritt im Stoffstrompfad  $w$  betrachtet wird, ist die Erzeugung einer stoffstrominduzierten Zeitbeziehung erforderlich. Dazu wird in Zeile 15 zunächst die Anordnungsbeziehung  $(j, j')$  zu  $E^{flow}$  hinzugefügt. Die Variable  $j$  ist gemäß Zeile 19 immer auf denjenigen Stoffstromvorgang gesetzt, der in der vorangegangenen Iteration von TransformationSchemeGra erzeugt wurde. Der zeitliche Mindestabstand wird auf die Dauer des Vorgängers gesetzt, d. h.  $d_{jj'}^{flow} := d_j$ . Dies lässt sich darauf zurückführen, dass ausschließlich granulare Stoffstromvorgänge geplant werden. Jeder granulare Stoffstromvorgang  $j$  stellt die Verarbeitung einer diskreten Stoffeinheit dar und dauert genauso lange, wie diese Verarbeitung dauert. Somit kann davon ausgegangen werden, dass die verarbeitete Stoffeinheit erst zum Endzeitpunkt von  $j$  zur Verfügung steht. Der Nachfolger  $j'$  kann daher frühestens zu diesem Zeitpunkt starten. Bildlich gesprochen formen die Anordnungsbeziehungen  $(j, j') \in E^{flow}$  eine Kette aus den Stoffstromvorgängen zu jeder Stoffeinheit. Das Ende jeder Kette wird über eine zusätzliche stoffstrominduzierte Zeitbeziehung mit dem fiktiven Stoffstromvorgang  $(J + 1)^o$  verknüpft, um zu gewährleisten, dass  $(J + 1)^o$  das Ende der Stoffverarbeitung abbildet (vgl. Zeile 21).

Jeder granulare Stoffstromvorgang verarbeitet eine Stoffeinheit, die er zu seinem Start aus einer kumulativen Ressource auslagert und zu seinem Ende in eine kumulative Ressource einlagert. Damit sind die über die Ein- und Auslagerungen abgebildeten Stoffströme auch konsistent zu den stoffstrominduzierten Zeitbeziehungen: Am Ende jedes Stoffstromvorgangs  $j$  wird die verarbeitete Stoffeinheit eingelagert und der Nachfolger  $j'$  kann starten,

welcher die Stoffeinheit zu seinem Start auslagert. In einer Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  liegen für die Stoffstromvorgänge folglich nur Nutzungen zum Start  $r_{jk}^{y,\text{start}} \in \{-1, 0, 1\}$  und Nutzungen zum Ende  $r_{jk}^{y,\text{end}} \in \{-1, 0, 1\}$  vor. Diese Nutzungen werden in den Zeilen 16 und 18 gesetzt. In Zeile 18 wird die Auslagerung des in der jeweiligen Iteration erzeugten Stoffstromvorgangs  $j'$  aus der ihm vorgeschalteten kumulativen Ressource  $k_{\text{step}}(w)$  gesetzt. In der Zeile 16 wird die Einlagerung des in der vorangegangenen Iteration erzeugten Stoffstromvorgang  $j$  in die ihm nachgeschaltete kumulative Ressource  $k_{\text{step}}(w)$  rückwirkend gesetzt. Diese rückwirkende Belegung macht Sinn, da der letzte zu einem Stoffstrompfad  $w$  erzeugte Stoffstromvorgang keine Stoffeinheit mehr einlagert (auf ihn folgt die Senke). Somit wäre eine weitere Fallunterscheidung erforderlich, die mit der rückwirkenden Belegung vermieden wird. Die rückwirkende Anweisung ist nur im Sonst-Block der Fallunterscheidung von Zeile 12 enthalten, da der im Dann-Block betrachtete Stoffstromvorgang  $j'$  noch keinen zuvor erzeugten Stoffstromvorgang  $j$  hat.

In Zeile 25 wird die erzeugte Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  mit allen übrigen Problemparametern aus der Ursprungsinstanz  $\mathcal{I}_{\text{RMCPSP}}$  vervollständigt. Dabei handelt es sich um die Menge der Rückbauvorgänge  $V$ , die Menge der erneuerbaren Ressourcen  $\mathcal{R}^\alpha$  und deren Kapazitäten  $R_k^\alpha$ , die Menge der Stoffstrompfade  $\mathcal{W}$ , die erneuerbaren Ressourcennutzungen  $r_{ik}^\alpha$ , die Dauern  $d_i$ , die Vorrangbeziehungen  $E$ , die Stofffreisetzungen  $(f_{iw}^{y,\text{start}}, f_{iw}^{y,\text{lin}}, f_{iw}^{y,\text{end}})$  und die kumulativen Ressourcenkapazitäten  $R_k^Y$ .

**Beispiel 4.2 (Fortsetzung des Beispiels 3.1):** Nachfolgend werden die Schritte von TransformationSchemeGra anhand der in Beispiel 3.1 beschriebenen Instanz  $\mathcal{I}_{\text{RMCPSP}}$  erläutert. Ergänzend zu den im Beispiel 3.1 gegebenen Problemparametern seien nun folgende gegeben: Für Rückbauvorgang 1 gilt  $d_1 = 10$ . Die Verarbeitung jeder Stoffeinheit in jedem Verarbeitungsschritt P1, ..., P8 dauert zwei Perioden und benötigt eine Einheit einer erneuerbaren Ressource R1, d. h.  $d_p = 2$  und  $r_{p,R1}^\alpha = 1$  für alle  $p \in \{P1, \dots, P8\}$ . Der Rückbauvorgang 1 setzt alle Stoffe gleichmäßig während seiner Ausführung in die Stoffstrompfade  $w \in \{W1, \dots, W4\}$  frei. Für die Freisetzungsarten gilt daher  $f_{1,W1}^{y,\text{lin}} = 1$ ,  $f_{1,W2}^{y,\text{lin}} = 2$ ,  $f_{1,W3}^{y,\text{lin}} = 1$  und  $f_{1,W4}^{y,\text{lin}} = 1$  (vgl. Beispiel 3.1). Die übrigen Freisetzungsarten sind null. Es sei angemerkt, dass die gleichmäßige Freisetzung einer einzigen Stoffeinheit über die Vorgangsausführung äquivalent zur Freisetzung dieser Stoffeinheit am Ende des Vorgangs ist (aufgrund der Abrundungsfunktion in Gleichung (3.4)).

Da Beispiel 3.1 die fünf Lager L1, ..., L5 enthält, werden in Zeile 3 die gleichnamigen kumulativen Ressourcen erzeugt. Anschließend wird Stoffstrompfad W1 betrachtet,  $u := 1$  und  $\text{step} := 1$  gesetzt. Vom ersten Verarbeitungsschritt P1 wird in Zeile 9 der granulare Stoffstromvorgang  $1^o[P1]$  abgeleitet. Seine Dauer wird auf  $d_{1^o} := 2$  und seine erneuerbare Ressourcennutzung auf  $r_{1^o,R1}^\alpha := 1$  gesetzt. Eine Anordnungsbeziehung  $(1, 1^o)$  wird in Zeile 13 erzeugt und zu  $E^{\text{rel}}$  hinzugefügt. Dann wird

$$d_{1,1^o}^{\text{rel}} := \min \left\{ \tau \in \mathbb{Z}_{\geq 0} \mid f_{1,W1}^Y(0, \tau) \geq 1 \right\}.$$

bestimmt. Da  $f_{1,W1}^Y(0, \tau) = 0$  für  $\tau = 0, \dots, 9$  und erst  $f_{1,W1}^Y(0, 10) = 1$ , wird  $d_{1,1^o}^{\text{rel}} := 10$  gesetzt. Dies lässt sich auch verbal begründen: Da die einzige von Rückbauvorgang



1 in W1 freigesetzte Stoffeinheit erst zehn Perioden nach seinem Start vollständig zur Verfügung steht (aufgrund der Abrundungsfunktion in Gleichung (3.4)), muss derjenige Stoffstromvorgang, der diese Stoffeinheit verarbeitet, auch mindestens zehn Perioden nach dem Start des Rückbauvorgangs 1 starten. In Zeile 18 wird die Auslagerung einer Stoffeinheit durch Stoffstromvorgang  $1^o$  aus der ihm vorgeschalteten kumulativen Ressource L1 gesetzt, d. h.  $r_{1^o, L1}^{y, start} := -1$ .

In der nächsten Iteration ( $step = 2$ ) wird in Zeile 9 vom zweiten Verarbeitungsschritt in W1 der granulare Stoffstromvorgang  $2^o$  [P2] abgeleitet. Seine Dauer wird auf  $d_{2^o} := 2$  und seine erneuerbare Ressourcennutzung auf  $r_{2^o, R1}^\alpha := 1$  gesetzt. In Zeile 15 wird eine Anordnungsbeziehung ( $1^o, 2^o$ ) erzeugt und zu  $E^{flow}$  hinzugefügt. Der zeitliche Mindestabstand wird auf  $d_{1^o, 2^o}^{flow} := d_{1^o} = 2$  gesetzt. In Zeile 16 wird rückwirkend die Einlagerung einer Stoffeinheit durch  $1^o$  in die kumulative Ressource L1 gesetzt, d. h.  $r_{1^o, L1}^{y, end} := 1$ .

Im weiteren Verlauf des granularen Transformationsschemas werden für die beiden verbleibenden Verarbeitungsschritte in W1 die beiden Stoffstromvorgänge  $3^o$  [P3] und  $4^o$  [P7] erzeugt. Dann werden die Stoffstromvorgänge  $5^o$  [P4],  $6^o$  [P5] und  $7^o$  [P3] für die erste Stoffeinheit erzeugt, die Stoffstrompfad W2 durchläuft. Für die zweite Stoffeinheit, die Stoffstrompfad W2 durchläuft, werden die Stoffstromvorgänge  $8^o$  [P4],  $9^o$  [P5] und  $10^o$  [P3] erzeugt. Dann werden die Stoffstromvorgänge  $11^o$  [P4] und  $12^o$  [P5] für diejenige Stoffeinheit erzeugt, die Stoffstrompfad W3 durchläuft. Und zuletzt wird der Stoffstromvorgang  $13^o$  [P4] für diejenige Stoffeinheit erzeugt, die Stoffstrompfad W4 durchläuft.

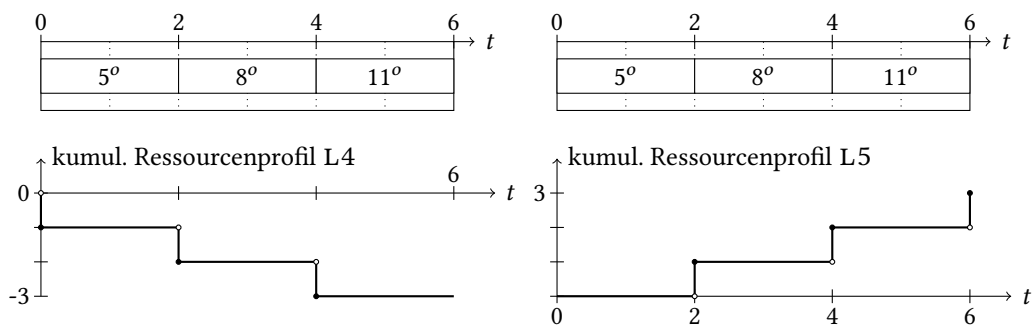
Da für Stoffstrompfad W2 gilt, dass die erste freigesetzte Stoffeinheit fünf Perioden nach dem Start von Rückbauvorgang 1 zur Verfügung steht und die zweite freigesetzte Stoffeinheit zehn Perioden, werden  $d_{1, 5^o}^{rel} := 5$  und  $d_{1, 8^o}^{rel} := 10$  als zeitliche Mindestabstände gesetzt. Für die übrigen zeitlichen Mindestabstände zu den Freisetzungsbearbeitungen gilt  $d_{1, 11^o}^{rel} = d_{1, 13^o}^{rel} := 10$ , weil in die Stoffstrompfade W3 und W4 nur je eine Stoffeinheit freigesetzt wird und diese erst zum Ende des Rückbauvorgangs 1 vollständig zur Verfügung steht.

Die berechneten kumulativen Ressourcennutzungen aller Stoffstromvorgänge werden aus Übersichtlichkeitsgründen in einer unten nachfolgenden Matrix zusammengefasst. Wie in Beispiel 3.1 beschrieben, setzt Rückbauvorgang 1 eine Stoffeinheit in L1, eine Stoffeinheit in L2 und drei Stoffeinheiten in L4 frei. Wenn für jede kumulative Ressource  $k = L1, \dots, L5$  dieser Wert und die Spaltensumme aus der Matrix addiert werden, ergibt sich null. Dies zeigt, dass die Flusserhaltung in der Instanz  $\mathcal{I}_{RCSP/M-gra}$  gewährleistet ist und zum Ende des Projekts alle Stoffe zur Senke geführt sind. Beispielsweise ist die Spaltensumme der ersten Spalte, welche zu L1 gehört,  $-1$ . Wird die Freisetzung einer Stoffeinheit durch Rückbauvorgang 1 hinzuaddiert, ergibt sich null.

Um die Ein- und Auslagerungen durch granulare Stoffstromvorgänge zu veranschaulichen, werden exemplarisch die granularen Stoffstromvorgänge  $5^o$  [P4],  $8^o$  [P4] und  $11^o$  [P4] betrachtet. Es handelt sich hierbei um diejenigen Stoffstromvorgänge, die von Verarbeitungsschritt P4 abgeleitet wurden. Jeder von ihnen lagert aus ihrer vorgeschalteten kumulativen Ressource L4 eine Stoffeinheit aus und lagert sie in ihre nachgeschaltete

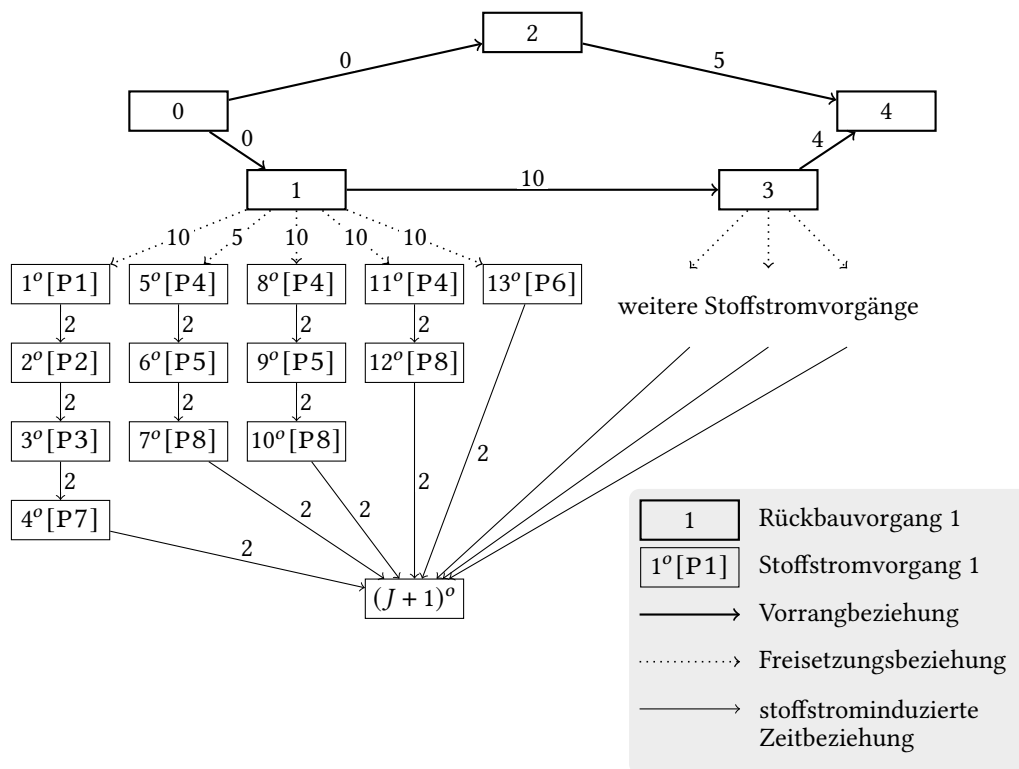
$$\left( r_{jk}^{y,start}, r_{jk}^{y,end} \right)_{j=1^o, \dots, 13^o; k=L1, \dots, L5} = \begin{pmatrix} -1, 1 & 0, 0 & 0, 0 & 0, 0 & 0, 0 \\ -1, 0 & 0, 0 & 0, 1 & 0, 0 & 0, 0 \\ 0, 0 & 0, 0 & -1, 0 & 0, 0 & 0, 1 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 & -1, 0 \\ 0, 0 & 0, 0 & 0, 0 & -1, 0 & 0, 1 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 & -1, 1 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 & -1, 0 \\ 0, 0 & 0, 0 & 0, 0 & -1, 0 & 0, 1 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 & -1, 1 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 & -1, 0 \\ 0, 0 & 0, 0 & 0, 0 & -1, 0 & 0, 1 \\ 0, 0 & 0, 0 & 0, 0 & 0, 0 & -1, 0 \\ 0, 0 & -1, 0 & 0, 0 & 0, 0 & 0, 0 \end{pmatrix}$$

kumulative Ressource L5 ein. Abbildung 4.2 zeigt die Veränderung der Ressourcenprofile dieser beiden kumulativen Ressourcen, wenn die genannten Stoffstromvorgänge in einem Schedule sequenziell und ohne Leerlaufzeiten eingeplant sind (vgl. Gantt-Diagramme im oberen Teil der Abbildung). □



**Abbildung 4.2:** Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.2.

Abbildung 4.3 zeigt exemplarisch den Vorgangsknotennetzplan  $N^{temp}$  zu einem Projekt mit den Rückbauvorgängen  $i = 1, \dots, 4$ . Davon setzen Rückbauvorgang 1 und Rückbauvorgang 3 Stoffe frei. Rückbauvorgang 1 und Stoffstromvorgänge  $Succ^o(1)$  entstammen dem vorangegangenen Beispiel 4.2. Die Kantengewichte entsprechen den Dauern der Rückbauvorgänge bzw. den zeitlichen Mindestabständen. Es ist deutlich zu erkennen, wie die stoffstrominduzierten Zeitbeziehungen Ketten aus den Stoffstromvorgängen formen.



**Abbildung 4.3:** Exemplarischer Vorgangsknotennetzplan einer Instanz  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$ ; Rückbauvorgang 1 und Stoffstromvorgänge  $\text{Succ}^0(1)$  entstammen dem Beispiel 4.2.

#### 4.2.2 Aggregiertes Transformationsschema

Der Grundgedanke des *aggregierten Transformationsschemas* ist es, die in einer Instanz  $\mathcal{I}_{\text{RMCPSP}}$  über Stoffstrompfade und Stofffreisetzungen beschriebenen Stoffströme solange gemeinsam zu modellieren, wie der Verlauf der Stoffströme miteinander vergleichbar ist. Eine solche Ausgestaltungsmöglichkeit des Transformationsschemas ist dadurch motiviert, dass die Stofffreisetzungen in einer realen Probleminstanz je nach gewählter Stoffeinheit große Werte annehmen können. Dafür müssen sehr viele granulare Stoffstromvorgänge erzeugt werden, die sich gleichen. An deren Stelle kann auch ein *aggregierter Stoffstromvorgang* treten, der die granularen Stoffstromvorgänge in sich vereint.

Das aggregierte Transformationsschema ist als Erweiterung des granularen Transformationsschemas konzipiert. Das bedeutet, die mit dem granularen Transformationsschema erzeugte Instanz  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$  bildet den Ausgangspunkt. Durch Aggregation der darin enthaltenen granularen Stoffstromvorgängen wird die Instanzgröße reduziert, was wiederum darauf abzielt, die Lösbarkeit einer Instanz zu verbessern. Die Lösbarkeit beschreibt, inwiefern für eine Instanz innerhalb praktikabler Rechenzeiten Lösungen hoher Qualität berechnet werden können. Bis wann eine Rechenzeit als praktikabel gilt und was unter einer hohen Lösungsqualität verstanden wird, hängt von verschiedenen Faktoren ab und ist lediglich qualitativ zu verstehen. Die Instanzgröße hat aufgrund der NP-Schwere des

Problems generell einen Einfluss auf die Lösbarkeit der Instanz (Coelho und Vanhoucke 2020, S. 2). Coelho und Vanhoucke (2020) zeigen für das (RCPSp) allerdings auch, dass relativ kleine Instanzen sehr schwer zu lösen sein können, wohingegen es einfach zu lösende große Instanzen gibt. Die Komplexitätsklasse gibt lediglich Auskunft über das Verhalten eines Lösungsverfahrens im ungünstigsten Fall und ermöglicht keine allgemeingültige Aussage zur Lösbarkeit jeder Problem Instanz (Coelho und Vanhoucke 2020, S. 2). Vor diesem Hintergrund gilt es zu beachten, dass die Reduktion der Instanzgröße gegebenenfalls nur zu einem gewissen Grad zur Verbesserung der Lösbarkeit beiträgt.

Darüber hinaus schränkt die Aggregation die Planungsflexibilität ein, was die Realitätstreue des Modells reduziert. Aggregierte Stoffstromvorgänge werden per Definition als Einheit geplant und erlauben daher keine Verarbeitungsunterbrechungen, wie sie mit granularen Vorgängen zwischen jeder einzelnen Stoffeinheit möglich sind. Bei der Ausgestaltung des aggregierten Transformationsschemas gilt es folglich, einen Kompromiss im Spannungsfeld zwischen der Instanzgröße als Maß für die Lösbarkeit und der Realitätstreue zu finden. In Abschnitt 4.2.4 werden theoretische Überlegungen zu diesem Spannungsfeld angestellt. In Kapitel 6 wird experimentell untersucht, wie sich die Instanzgröße und die Realitätstreue auf die Lösung des (RCPSp/M) auswirken.

Der nachfolgend beschriebenen Aggregation liegt die Annahme zugrunde, dass die Dauer der Stoffverarbeitung linear mit der Anzahl der verarbeiteten Stoffeinheiten skaliert, während die erneuerbaren Ressourcennutzungen konstant bleiben. Bildlich gesprochen werden bei der Aggregation Stoffstromvorgänge im Gantt-Diagramm aneinandergehängt und zu einer Einheit verschmolzen. Diese Art der Aggregation wird als *horizontale Aggregation* bezeichnet. Welche Voraussetzungen dafür vorliegen müssen und welche alternativen Ausgestaltungsmöglichkeiten es gibt, wird in Abschnitt 4.2.5 erörtert.

Eine mit dem aggregierten Transformationsschema erzeugte Instanz  $\mathcal{I}_{\text{RCPSp/M}}$  weist eine spezielle Struktur auf, weshalb für sie die Notation  $\mathcal{I}_{\text{RCPSp/M-agg}}$  verwendet wird. Das aggregierte Transformationsschema ist formal in Algorithmus 3 beschrieben und wird als `TransformationSchemeAgg` bezeichnet. Die Zeilenreferenzen in diesem Abschnitt beziehen sich auf Algorithmus 3. Zunächst wird das aggregierte Transformationsschema verbal erläutert. Am Ende dieses Abschnitts wird an die in Beispiel 3.1 beschriebene Instanz  $\mathcal{I}_{\text{RMCPSP}}$  angeknüpft und gezeigt, wie die Erzeugung der dazugehörigen Instanz  $\mathcal{I}_{\text{RCPSp/M-agg}}$  abläuft.

Als Erweiterung des granularen Transformationsschemas ist das aggregierte Transformationsschema so gestaltet, dass ein zulässiger Schedule zu einer Instanz  $\mathcal{I}_{\text{RCPSp/M-agg}}$  stets in einen zulässigen Schedule zur zugrundeliegenden Instanz  $\mathcal{I}_{\text{RCPSp/M-gra}}$  rücktransformiert werden kann. Diese Vorgehensweise stützt sich auf der Annahme, dass die granulare Instanz die reale Problemstellung am besten beschreibt und die aggregierte Instanz lediglich eine Abstrahierung davon ist. Die Rücktransformierbarkeit stellt sicher, dass jeder Schedule zur Instanz  $\mathcal{I}_{\text{RCPSp/M-agg}}$  ausführbar ist, d. h. keine Restriktionen des realen Problems verletzt. Sie ermöglicht auch eine bessere Vergleichbarkeit der beiden Transformationsschemas hinsichtlich des Spannungsfeldes zwischen Instanzgröße und Realitätstreue.

**Algorithmus 3:** TransformationSchemeAgg

---

**Eingabe:** Instanz  $\mathcal{I}_{\text{RMCPSP}}$ ; Dauern  $d_p$  und ern. Ressourcennutzungen  $r_{pk}^\alpha$  für alle  $p \in \mathcal{P}$  und  $k \in \mathcal{R}^\alpha$

- 1 berechne  $\mathcal{I}_{\text{RCPSP/M-gra}} := \text{TransformationSchemeGra}(\mathcal{I}_{\text{RMCPSP}}, \dots)$ ;
- 2 **solange** aggregierbare Teilmengen  $\hat{V}^o$  existieren **tue**
- 3     erzeuge einen neuen aggregierten Stoffstromvorgang  $\hat{j}$ ;
- 4     setze  $d_{\hat{j}} := \sum_{j \in \hat{V}^o} d_j$ ;
- 5     setze  $r_{\hat{j}k}^\alpha := r_{jk}^\alpha$  für ein beliebiges  $j \in \hat{V}^o$  und jedes  $k \in \mathcal{R}^\alpha$ ;
- 6     setze  $r_{\hat{j}k}^{\mathcal{Y}, \text{lin}} := \sum_{j \in \hat{V}^o} (r_{jk}^{\mathcal{Y}, \text{start}} + r_{jk}^{\mathcal{Y}, \text{end}})$  für jedes  $k \in \mathcal{R}^{\mathcal{Y}}$ ;
- 7     **wenn**  $j \in \hat{V}^o$  folgen auf einen Rückbauvorgang  $i \in V$  **dann**
- 8         sortiere  $j \in \hat{V}^o$  als  $(j_1, j_2, \dots, j_{|\hat{V}^o|})$  nach nichtfallendem  $d_{ij}^{\text{rel}}$ ;
- 9         berechne  $d_{ij}^{\text{rel}}$  gemäß Gleichung (4.17);
- 10     **sonst**  $/* j \in \hat{V}^o$  folgen auf einen Stoffstromvorgang  $j'' \in V^o *$
- 11         sortiere  $j \in \hat{V}^o$  als  $(j_1, j_2, \dots, j_{|\hat{V}^o|})$  nach nichtfallendem  $d_{j''j}^{\text{flow}}$ ;
- 12         berechne  $d_{j''j}^{\text{flow}}$  gemäß Gleichung (4.18);
- 13     **Ende**
- 14     **für**  $u = 1, \dots, |\hat{V}^o|$  **tue**
- 15         setze  $j'$  auf den direkten Nachfolger von  $j_u$  aus  $(j_1, j_2, \dots, j_{|\hat{V}^o|})$ ;
- 16         berechne  $d_{jj'}^{\text{flow}}$  gemäß Gleichung (4.19);
- 17     **Ende**
- 18     ersetze in den Anordnungsbeziehungen jedes  $j \in \hat{V}^o$  durch  $\hat{j}$ ;
- 19     lösche jeden granularen Stoffstromvorgang  $j \in \hat{V}^o$ ;
- 20 **Ende**

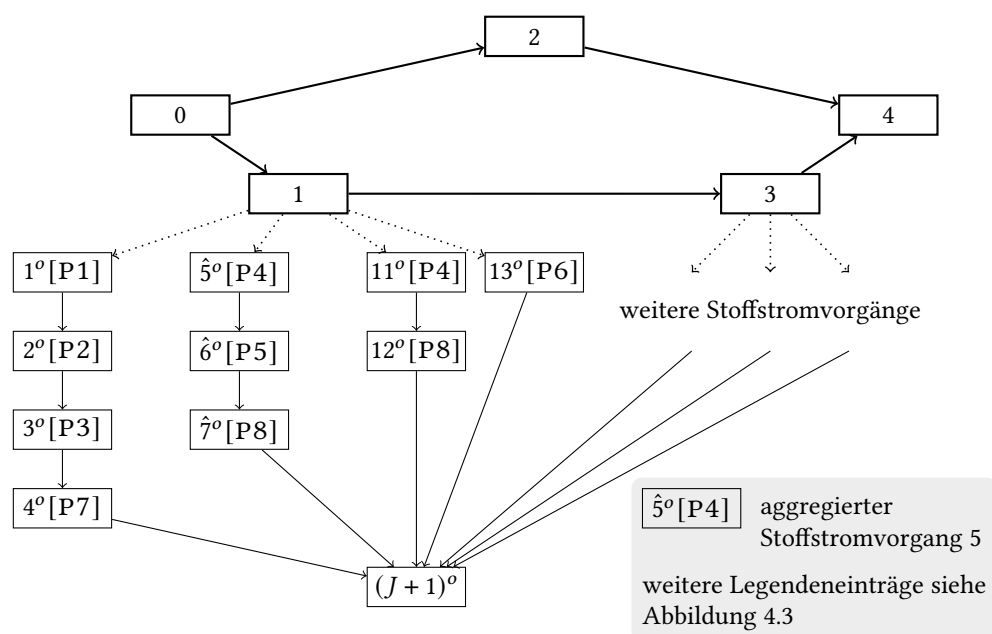
**Ausgabe:** Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$

---

TransformationSchemeAgg bekommt eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  als Eingabe übergeben. Diese Instanz wird zunächst mittels Ausführung von TransformationSchemeGra in eine Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  umgewandelt (vgl. Zeile 1). In  $\mathcal{I}_{\text{RCPSP/M-gra}}$  weisen alle von einem Verarbeitungsschritt  $p \in \mathcal{P}$  abgeleiteten granularen Stoffstromvorgänge dieselbe Dauer  $d_p \in \mathbb{Z}_{\geq 0}$  und dieselben erneuerbaren Ressourcennutzungen  $r_{pk}^\alpha \in \mathbb{Z}_{\geq 0}$  für alle  $k \in \mathcal{R}^\alpha$  auf. Bei  $d_p$  und  $r_{pk}^\alpha$  handelt es sich um Eingaben zum Transformationsschema. Im weiteren Verlauf von TransformationSchemeAgg wird die Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  umgebaut, sodass daraus eine Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$  entsteht.

Eine Menge von granularen Stoffstromvorgängen, die in einer Iteration des aggregierten Transformationsschemas durch einen aggregierten Stoffstromvorgang ersetzt werden, wird als *aggregierbare Teilmenge*  $\hat{V}^o \subseteq \{j \in V^o \mid j \text{ ist granulär}\}$  bezeichnet. In der Schleife in Zeile 2 werden solange aggregierbare Teilmengen gesucht, bis keine mehr existieren. Anhand der in Beispiel 4.2 erzeugten Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  wird nachfolgend hergeleitet, nach welcher Logik die aggregierbaren Teilmengen gebildet werden. Ein naheliegender

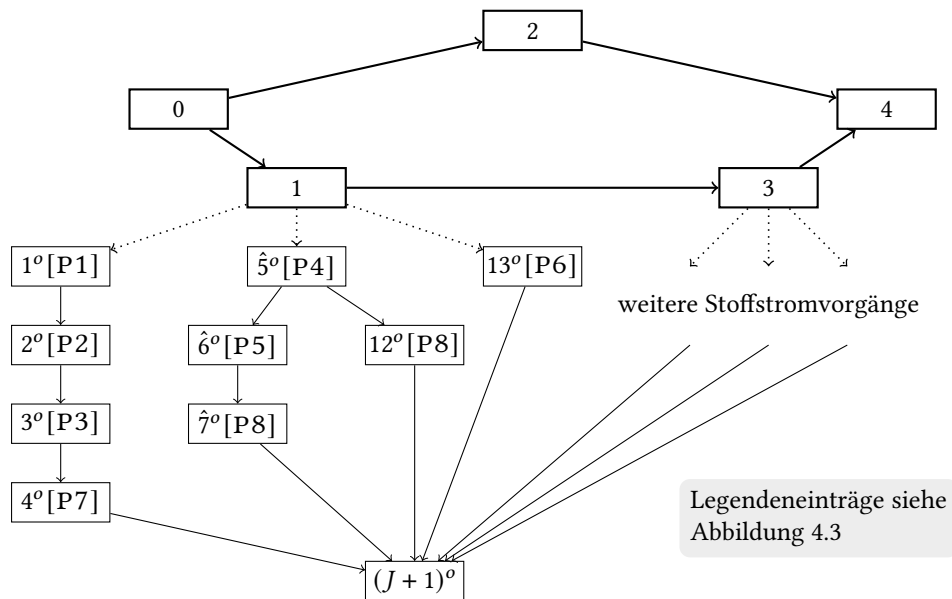
Ansatz ist es, für jeden Rückbauvorgang  $i \in V$ , jeden Stoffstrompfad  $w \in \mathcal{W}$  und jeden Verarbeitungsschritt  $p$  in  $w$  alle von  $p$  abgeleiteten granularen Stoffstromvorgänge zu aggregieren. Das ist möglich, da diese granularen Stoffstromvorgänge dieselben erneuerbaren Ressourcen nutzen. Außerdem handelt es sich um denselben Stoffstrom, d. h., die Stoffverarbeitung erfolgt nach demselben Muster, wodurch eine Zusammenfassung als sinnvoll erscheint. Bildlich gesprochen werden alle Stoffstromvorgänge, die für denselben Stoffstrompfad erzeugt wurden, auf jeder Ebene der stoffstrominduzierten Zeitstruktur verschmolzen. Dieser Ansatz ist in Abbildung 4.4 dargestellt und wird insbesondere im Vergleich zu Abbildung 4.3 nachvollziehbar. Da der Stoffstrompfad  $W_2$  von zwei Stoffeinheiten durchlaufen wird und aus drei Verarbeitungsschritten besteht, wurden die sechs granularen Stoffstromvorgänge  $5^0, \dots, 10^0$  erzeugt. Diese werden nun paarweise verschmolzen, sodass in Abbildung 4.4 an deren Stelle nur noch die drei aggregierten Stoffstromvorgänge  $\hat{5}^0 [P4]$ ,  $\hat{6}^0 [P5]$  und  $\hat{7}^0 [P8]$  übrigbleiben.



**Abbildung 4.4:** Exemplarischer Vorgangsknotennetzplan einer teilweise aggregierten Instanz  $\mathcal{I}_{RCSP/M-agg}$  (vgl. Abbildung 4.3).

Bei der Betrachtung von Abbildung 4.4 fällt auf, dass auch der granulare Stoffstromvorgang  $11^0$  von Verarbeitungsschritt P4 abgeleitet wurde. Er nutzt somit dieselben erneuerbaren Ressourcen wie  $\hat{5}^0$ . Außerdem folgt  $11^0$  wie  $\hat{5}^0$  direkt auf den Rückbauvorgang 1. Die mit  $11^0$  modellierte Stoffverarbeitung erfolgt somit nach demselben Muster wie die mit  $\hat{5}^0$  modellierte Stoffverarbeitung. Es erscheint daher ebenfalls sinnvoll, auch  $11^0$  mit  $\hat{5}^0$  zu verschmelzen, wie in Abbildung 4.5 dargestellt.

Nun kann argumentiert werden, dass sowohl der granulare Stoffstromvorgang  $12^0$  als auch der aggregierte Stoffstromvorgang  $\hat{7}^0$  von Verarbeitungsschritt P8 abgeleitet wurden und sie deshalb verschmolzen werden sollten. Eine solche Verschmelzung würde allerdings die Realitätstreue des Modells verfälschen, da sich die mit  $12^0$  modellierte Stoffverarbeitung von der mit  $\hat{7}^0$  modellierten Stoffverarbeitung unterscheidet. Am besten ist dies bei Betrachtung



**Abbildung 4.5:** Exemplarischer Vorgangsknotenetzplan einer vollständig aggregierten Instanz  $\mathcal{I}_{\text{RCPS}/M\text{-agg}}$  (vgl. Abbildung 4.4).

von Abbildung 4.5 zu erkennen, da sich beide Stoffstromvorgänge nicht auf derselben Ebene der stoffstrominduzierten Zeitstruktur befinden. Würde  $12^\circ$  mit  $7^\circ$  verschmolzen werden, müsste aufgrund der Anordnungsbeziehung ( $6^\circ, 7^\circ$ ) auf die Ausführung von  $6^\circ$  gewartet werden, bis die Verarbeitung der für  $12^\circ$  vorgesehenen Stoffeinheit begonnen werden kann. Allerdings durchläuft diese Stoffeinheit gar nicht den Verarbeitungsschritt P5, von welchem  $6^\circ$  abgeleitet wurde. Es würde daher modellbedingt ein zeitlich verzögerter Schedule berechnet. Um dies zu vermeiden, muss  $12^\circ$  auch in der Instanz  $\mathcal{I}_{\text{RCPS}/M\text{-agg}}$  als granularer Stoffstromvorgang und damit unabhängig einplanbar weitergeführt werden.

Die exemplarisch beschriebene Logik zur Bildung der aggregierbaren Teilmengen ist ein Vorschlag, der als sinnvoller Kompromiss im Spannungsfeld zwischen Instanzgröße und Realitätstreue erachtet wird. Formal ausgedrückt, muss eine aggregierbare Teilmenge  $\hat{V}^\circ$  für diese Logik folgende Bedingungen erfüllen:

- (1) Sie besteht aus mindestens zwei granularen Stoffstromvorgängen.
- (2) Für alle  $j, j' \in \hat{V}^\circ$  und alle  $k \in \mathcal{R}^\alpha$  gilt, dass  $r_{jk}^\alpha = r_{j'k}^\alpha$ . Nur wenn alle erneuerbaren Ressourcennutzungen gleich sind, ist eine horizontale Aggregation möglich. Andernfalls würde die erneuerbare Ressourcennutzung des aggregierten Stoffstromvorgangs über seine Ausführung variieren, was im (RCPS/M) nicht vorgesehen ist. Da in TransformationSchemeGra alle von einem Verarbeitungsschritt  $p$  abgeleiteten Stoffstromvorgänge dieselben erneuerbaren Ressourcennutzungen aufweisen, wird diese Bedingung wie folgt umformuliert: Für alle  $j \in \hat{V}^\circ$  gilt, dass sie vom selben Verarbeitungsschritt  $p \in \mathcal{P}$  abgeleitet wurden.

- (3) Für alle  $j, j' \in \hat{V}^o$  und alle  $k \in \mathcal{R}^y$  gilt, dass  $r_{jk}^{y,start} = r_{j'k}^{y,start}$  und  $r_{jk}^{y,end} = r_{j'k}^{y,end}$ . Nur wenn diese kumulativen Ressourcennutzungen gleich sind, ist eine Aggregation sinnvoll. Andernfalls würde die kumulative Ressourcennutzung des aggregierten Stoffstromvorgangs über seine Ausführung variieren und könnte nicht mehr über die vorhandenen Nutzungsarten abgebildet werden.
- (4) Für alle  $j \in \hat{V}^o$  gilt, dass sie denselben direkten Vorgänger haben (die im granularen Transformationsschema erzeugten Stoffstromvorgänge haben jeweils nur einen Vorgänger). Diese Bedingung stellt auch sicher, dass keine Stoffstromvorgänge aggregiert werden, die Stoffe verschiedener Rückbauvorgänge verarbeiten. Die Nachfolger sind irrelevant. Sie haben keinen Einfluss auf die frühestmöglichen Startzeitpunkte eines Stoffstromvorgangs und schränken daher die Realitätstreue nicht ein.

Für eine gefundene aggregierbare Teilmenge  $\hat{V}^o$  wird in Zeile 3 ein neuer aggregierter Stoffstromvorgang  $\hat{j}$  erzeugt. Dabei ist  $\hat{j}$  eine Hilfsvariable für den jeweils neu erzeugten aggregierten Stoffstromvorgang. Im Sinne der horizontalen Aggregation werden in Zeile 4 die Dauern der granularen Stoffstromvorgänge  $j \in \hat{V}^o$  aufsummiert und als Dauer von  $\hat{j}$  gesetzt. Die konstant bleibenden erneuerbaren Ressourcennutzungen werden in Zeile 5 von einem beliebigen granularen Stoffstromvorgang  $j \in \hat{V}^o$  an  $\hat{j}$  übergeben. Die kumulativen Ressourcennutzungen aller granularen Stoffstromvorgänge  $j \in \hat{V}^o$  werden in Zeile 6 aufsummiert und bilden die kumulative Ressourcennutzung von  $\hat{j}$ . Als Nutzungsart wird dafür die gleichmäßige Nutzung während der Ausführung von  $\hat{j}$  verwendet. Wie Beispiel 4.3 am Ende dieses Abschnitts zeigen wird, werden dadurch die einzelnen kumulativen Ressourcennutzungen der granularen Stoffstromvorgänge  $j \in \hat{V}^o$  in der aggregierten Instanz exakt nachgebildet. Dies stellt die Rücktransformierbarkeit sicher.

In den Zeilen 7 bis 18 wird die stoffstrominduzierte Zeitstruktur angepasst. Dies ist vergleichsweise aufwendig, weil wie beschrieben jede zulässige Lösung für eine Instanz  $\bar{\mathcal{I}}_{\text{RCPSM-agg}}$  in eine zulässige Lösung für die zugrundeliegende Instanz  $\bar{\mathcal{I}}_{\text{RCPSM-gra}}$  rücktransformiert werden können soll. Es ist daher sicherzustellen, dass die in der Instanz  $\bar{\mathcal{I}}_{\text{RCPSM-gra}}$  gegebenen zeitlichen Mindestabstände auch in der Instanz  $\bar{\mathcal{I}}_{\text{RCPSM-agg}}$  eingehalten werden. Zur Berechnung des zeitlichen Mindestabstands des aggregierten Stoffstromvorgangs  $\hat{j}$  zu seinem direkten Vorgänger sind folgende Schritte erforderlich: Gemäß Bedingung (4) für die Bildung aggregierbare Teilmenge  $\hat{V}^o$  haben alle granularen Stoffstromvorgänge  $j \in \hat{V}^o$  denselben direkten Vorgänger. Falls dies ein Rückbauvorgang  $i \in V$  ist (vgl. Zeile 7), werden die granularen Stoffstromvorgänge  $j \in \hat{V}^o$  als Folge  $(j_1, j_2, \dots, j_{|\hat{V}^o|})$  nach nichtfallenden zeitlichen Mindestabständen  $d_{ij}^{\text{rel}}$  sortiert (vgl. Zeile 8). Dann wird

$$d_{i\hat{j}}^{\text{rel}} := \min \left\{ \tau \in \mathbb{Z}_{\geq 0} \mid \tau + \frac{u-1}{|\hat{V}^o|} \cdot d_j \geq d_{iju}^{\text{rel}} \text{ für } u = 1, \dots, |\hat{V}^o| \right\} \quad (4.17)$$

gesetzt (vgl. Zeile 9). Mit dem Term  $\frac{u-1}{|\hat{V}^o|} \cdot d_j$  wird derjenige Zeitpunkt nach dem Start des aggregierten Stoffstromvorgangs  $\hat{j}$  berechnet, zu dem die Verarbeitung der  $u$ -ten Stoffeinheit startet. Dieser Zeitpunkt muss für alle Stoffeinheiten  $u = 1, \dots, |\hat{V}^o|$  gerade soweit um  $\tau$  Perioden nach hinten verschoben werden, bis er größer oder gleich  $d_{iju}^{\text{rel}}$  (=



der Zeitpunkt, zu dem der zugehörige granulare Stoffstromvorgang starten kann) ist. Die zuvor erfolgte Sortierung gibt vor, welche der in  $\hat{j}$  verarbeiteten Stoffeinheiten welchen granularen Stoffstromvorgängen  $j \in \hat{V}^o$  entstammen. Es ist naheliegend, dass Stoffeinheiten aus granularen Stoffstromvorgängen, die früher starten können, in  $\hat{j}$  auch früher verarbeitet werden. Daher wird nach nichtfallenden zeitlichen Mindestabständen  $d_{ij}^{rel}$  sortiert. Zum besseren Verständnis der Gleichung (4.17) wird auf das Beispiel 4.3 am Ende dieses Abschnitts verwiesen.

Falls der direkte Vorgänger der granularen Stoffstromvorgänge  $j \in \hat{V}^o$  ein Stoffstromvorgang  $j'' \in V^o$  ist (vgl. Zeile 10), werden die granularen Stoffstromvorgänge  $j \in \hat{V}^o$  als Folge  $(j_1, j_2, \dots, j_{|\hat{V}^o|})$  nach nichtfallenden zeitlichen Mindestabständen  $d_{j''j}^{flow}$  sortiert (vgl. Zeile 11). Dann wird analog zur vorangegangenen Gleichung (4.17)

$$d_{j''\hat{j}}^{rel} := \min \left\{ \tau \in \mathbb{Z}_{\geq 0} \mid \tau + \frac{u-1}{|\hat{V}^o|} \cdot d_j \geq d_{j''j_u}^{rel} \quad \text{für } u = 1, \dots, |\hat{V}^o| \right\} \quad (4.18)$$

gesetzt (vgl. Zeile 12).

Anschließend müssen die zeitlichen Mindestabstände zwischen  $\hat{j}$  und den direkten Nachfolgern aller granularen Stoffstromvorgänge  $j \in \hat{V}^o$  berechnet werden, sodass sie auf die Aggregation angepasst sind. Dazu werden in Zeile 14 mit  $u = 1, \dots, |\hat{V}^o|$  die Stoffstromvorgänge der sortierten Folge  $(j_1, j_2, \dots, j_{|\hat{V}^o|})$  durchlaufen. Die Hilfsvariable  $j'$  wird jeweils auf den direkten Nachfolger von  $j_u$  gesetzt (vgl. Zeile 15). Dann kann in Zeile 16 der zeitliche Mindestabstand

$$d_{\hat{j}j'}^{flow} := \frac{u}{|\hat{V}^o|} \cdot d_j \quad (4.19)$$

berechnet werden. Mit dem Term  $\frac{u}{|\hat{V}^o|} \cdot d_j$  wird derjenige Zeitpunkt nach dem Start des aggregierten Stoffstromvorgangs  $\hat{j}$  berechnet, zu dem die Verarbeitung der  $u$ -ten Stoffeinheit abgeschlossen ist. Zu diesem Zeitpunkt kann die Stoffeinheit an den Nachfolger  $j'$  übergeben werden, weshalb der zeitliche Mindestabstand entsprechend gesetzt wird. Zum besseren Verständnis der Gleichung (4.19) wird auf das Beispiel 4.3 am Ende dieses Absatzes verwiesen.

In Zeile 18 wird in allen Anordnungsbeziehungen  $(i, j) \in E^{rel}$ ,  $(j'', j) \in E^{flow}$  und  $(j, j') \in E^{flow}$ , für die  $j \in \hat{V}^o$  gilt,  $j$  durch  $\hat{j}$  ersetzt. Bildlich gesprochen werden in  $N^{temp}$  alle bisher mit den Knoten  $j \in \hat{V}^o$  inzidenten Kanten abgelöst und stattdessen mit dem Knoten des aggregierten Stoffstromvorgangs  $\hat{j}$  verbunden. Parallele Kanten werden entfernt.

Nachdem nun alle Problemparameter des aggregierten Stoffstromvorgangs  $\hat{j}$  festgelegt sind, werden in Zeile 19 die granularen Stoffstromvorgänge  $j \in \hat{V}^o$  gelöscht. Anschließend wird mit der Bestimmung der nächsten aggregierbaren Teilmenge fortgefahren.

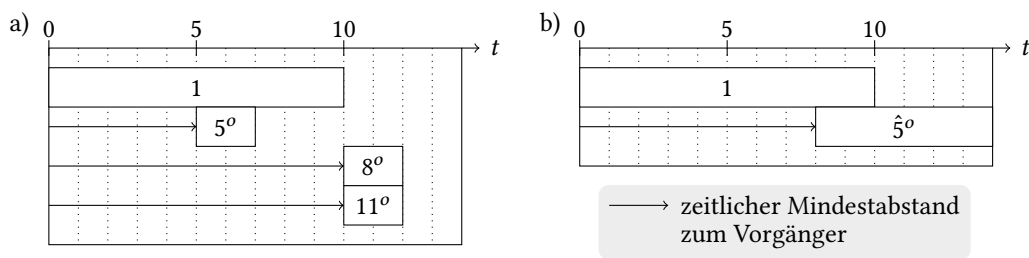
**Beispiel 4.3 (Fortsetzung der Beispiele 3.1 und 4.2):** Nachfolgend werden die Schritte des aggregierten Transformationsschemas anhand der in Beispiel 3.1 beschriebenen Instanz  $\mathcal{I}_{RMCPSP}$  erläutert. Diese Instanz wird zunächst wie in Beispiel 4.2 in die Instanz  $\mathcal{I}_{RCPSM/M-gra}$  transformiert.

In der ersten Iteration von TransformationSchemeAgg werden die drei granularen Stoffstromvorgänge  $5^o$ ,  $8^o$  und  $11^o$  als aggregierbare Teilmenge ausgewählt, d. h.  $\hat{V}^o = \{5^o, 8^o, 11^o\}$  (vgl. Abbildung 4.3). Für sie wird der aggregierte Stoffstromvorgang  $\hat{5}^o$  erzeugt. Seine Dauer wird auf  $d_{\hat{5}^o} := d_{5^o} + d_{8^o} + d_{11^o} = 2 + 2 + 2 = 6$  und seine erneuerbare Ressourcennutzung auf  $r_{\hat{5}^o, R1}^\alpha := r_{5^o, R1}^\alpha = 1$  gesetzt. Seine kumulativen Ressourcennutzungen werden auf  $r_{\hat{5}^o, L4}^{\gamma, lin} := r_{5^o, L4}^{\gamma, start} + r_{5^o, L4}^{\gamma, end} + r_{8^o, L4}^{\gamma, start} + r_{8^o, L4}^{\gamma, end} + r_{11^o, L4}^{\gamma, start} + r_{11^o, L4}^{\gamma, end} = -1 + 0 - 1 + 0 - 1 + 0 = -3$  und  $r_{\hat{5}^o, L5}^{\gamma, lin} := r_{5^o, L5}^{\gamma, start} + r_{5^o, L5}^{\gamma, end} + r_{8^o, L5}^{\gamma, start} + r_{8^o, L5}^{\gamma, end} + r_{11^o, L5}^{\gamma, start} + r_{11^o, L5}^{\gamma, end} = 0 + 1 + 0 + 1 + 0 + 1 = 3$  gesetzt.

In der Instanz  $\bar{I}_{RCPSP/M-gra}$  betragen die zeitlichen Mindestabstände  $d_{1,5^o}^{rel} = 5$ ,  $d_{1,8^o}^{rel} = 10$ ,  $d_{1,11^o}^{rel} = 10$ , wie in Beispiel 4.2 berechnet. Die aggregierbare Teilmenge  $\hat{V}^o = \{5^o, 8^o, 11^o\}$  wird nach nichtfallenden zeitlichen Mindestabständen als Folge  $(5^o, 8^o, 11^o)$  sortiert. Gemäß Gleichung (4.17) muss nun das kleinstmögliche  $\tau \in \mathbb{Z}_{\geq 0}$  gefunden werden, für das

$$\begin{aligned} \tau + (1 - 1)/3 \cdot 6 &\geq 5, \\ \tau + (2 - 1)/3 \cdot 6 &\geq 10 \text{ und} \\ \tau + (3 - 1)/3 \cdot 6 &\geq 10 \end{aligned}$$

erfüllt sind. Das trifft auf  $\tau = 8$  zu, d. h.  $d_{1, \hat{5}^o}^{rel} := 8$ . Abbildung 4.6 veranschaulicht dies. Sie zeigt ein Gantt-Diagramm für einen Schedule mit  $5^o$ ,  $8^o$ ,  $11^o$  (d. h. vor der Aggregation; vgl. Abbildung 4.6a) und ein Gantt-Diagramm für einen Schedule mit  $\hat{5}^o$  (d. h. nach der Aggregation, vgl. Abbildung 4.6b). Die Stoffstromvorgänge starten jeweils zu ihren frühestmöglichen Startzeitpunkten. Es ist ersichtlich, dass der Schedule mit  $\hat{5}^o$  zu einem Schedule mit  $5^o$ ,  $8^o$ ,  $11^o$  rücktransformiert werden kann. Bildlich gesprochen muss dazu in Abbildung 4.6b Stoffstromvorgang  $\hat{5}^o$  wieder in die drei granularen Stoffstromvorgänge  $5^o$ ,  $8^o$ ,  $11^o$  aufgespalten werden. Dann sind deren ursprünglichen zeitlichen Mindestabstände weiterhin korrekt eingehalten. Das wäre nicht der Fall, wenn  $d_{1, \hat{5}^o}^{rel} < 8$ .

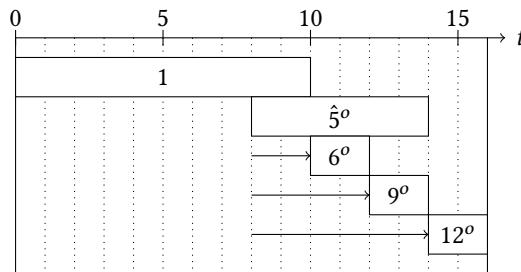


**Abbildung 4.6:** Gantt-Diagramme zu Beispiel 4.3: a) zeitliche Mindestabstände zwischen granularen Stoffstromvorgängen und ihrem Vorgänger; b) zeitlicher Mindestabstand zwischen aggregiertem Stoffstromvorgang und seinem Vorgänger.

Die direkten Nachfolger der als Folge ( $5^o$ ,  $8^o$ ,  $11^o$ ) sortierten granularen Stoffstromvorgänge sind  $6^o$ ,  $9^o$  und  $12^o$  (vgl. Abbildung 4.3). Ihre zeitlichen Mindestabstände zu  $\hat{5}^o$  werden gemäß Gleichung (4.19) auf

$$\begin{aligned} d_{\hat{5}^o, 6^o}^{flow} &:= 1/3 \cdot 6 = 2, \\ d_{\hat{5}^o, 9^o}^{flow} &:= 2/3 \cdot 6 = 4 \text{ und} \\ d_{\hat{5}^o, 12^o}^{flow} &:= 3/3 \cdot 6 = 6 \end{aligned}$$

gesetzt. Abbildung 4.7 veranschaulicht dies. Zum Zeitpunkt 10 ist die erste Stoffeinheit in  $\hat{5}^o$  verarbeitet, weshalb  $6^o$  starten kann. Zum Zeitpunkt 12 ist die zweite Stoffeinheit in  $\hat{5}^o$  verarbeitet, weshalb  $9^o$  starten kann. Und zum Zeitpunkt 14 ist die dritte Stoffeinheit in  $\hat{5}^o$  verarbeitet, weshalb  $12^o$  starten kann.



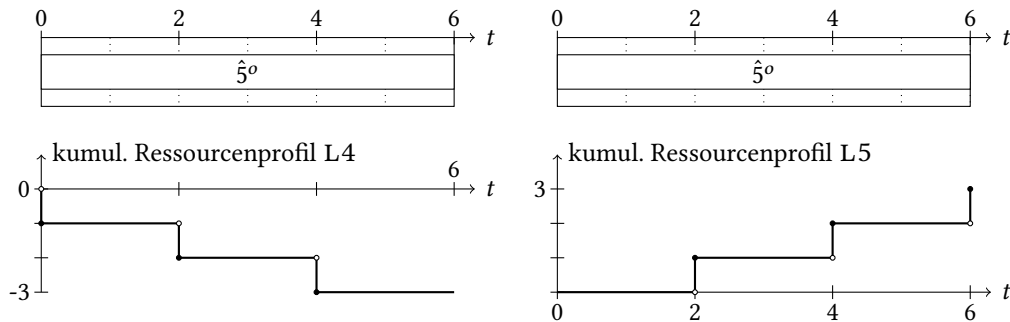
**Abbildung 4.7:** Gantt-Diagramm zu Beispiel 4.3: Zeitliche Mindestabstände zwischen einem aggregierten Stoffstromvorgang und seinen Nachfolgern.

In Zeile 18 werden die Anordnungsbeziehungen ( $1, 5^o$ ), ( $1, 8^o$ ), ( $1, 11^o$ ), ( $5^o, 6^o$ ), ( $8^o, 9^o$ ) und ( $11^o, 12^o$ ) zu ( $1, \hat{5}^o$ ), ( $\hat{5}^o, 6^o$ ), ( $\hat{5}^o, 9^o$ ) und ( $\hat{5}^o, 12^o$ ) umgewandelt. In Zeile 19 werden die granularen Stoffstromvorgänge  $5^o$ ,  $8^o$  und  $11^o$  gelöscht. In der zweiten Iteration wird  $\hat{V}^o := \{6^o, 9^o\}$  identifiziert und zu  $\hat{6}^o$  aggregiert. In der dritten Iteration wird  $\hat{V}^o := \{7^o, 10^o\}$  identifiziert und zu  $\hat{7}^o$  aggregiert. Auf weitere Details zu diesen Schritten wird verzichtet, da sie analog zu den beschriebenen ablaufen.

Um die Ein- und Auslagerungen durch aggregierte Stoffstromvorgänge zu veranschaulichen, wird exemplarisch der aggregierte Stoffstromvorgang  $\hat{5}^o$  betrachtet. Er aggregiert die granularen Stoffstromvorgänge  $5^o$ ,  $8^o$  und  $11^o$ , weshalb er gleichmäßig während seiner Ausführung drei Stoffeinheiten aus der kumulativen Ressource L4 auslagert und in die kumulative Ressource L5 einlagert. Abbildung 4.8 zeigt die Veränderung der Ressourcenprofile dieser beiden kumulativen Ressourcen, wenn  $\hat{5}^o$  zum Zeitpunkt 0 startet. Für  $t = 0$  gilt beispielsweise gemäß Gleichung (4.12)

$$\begin{aligned} r_{\hat{5}^o, L4}^Y(0, 0) &= r_{\hat{5}^o, L4}^{Y, start} + \left\lfloor (t + 1 - S_{\hat{5}^o}^o) r_{\hat{5}^o, L4}^{Y, lin} / d_{\hat{5}^o} \right\rfloor \\ &= 0 + \lfloor (0 + 1 - 0) \cdot (-3) / 6 \rfloor = -1 \text{ und} \\ r_{\hat{5}^o, L5}^Y(0, 0) &= r_{\hat{5}^o, L5}^{Y, start} + \left\lfloor (t - S_{\hat{5}^o}^o) r_{\hat{5}^o, L5}^{Y, lin} / d_{\hat{5}^o} \right\rfloor \\ &= 0 + \lfloor (0 - 0) \cdot 3 / 6 \rfloor = 0. \end{aligned}$$

Das heißt, zu diesem Zeitpunkt wurde eine Stoffeinheit aus L4 ausgelagert, aber noch keine Stoffeinheit in L5 eingelagert. Wird  $\hat{5}^o$  wieder in seine zugrundeliegenden granularen Stoffstromvorgänge  $5^o$ ,  $8^o$  und  $11^o$  aufgespalten, bleiben die kumulativen Ressourcenprofile in Abbildung 4.8 unverändert (vgl. Abbildung 4.2). Das zeigt, dass die gleichmäßige Nutzung während der Vorgangsausführung für aggregierte Stoffstromvorgänge geeignet ist, um die einzelnen kumulativen Ressourcennutzungen der granularen Stoffstromvorgänge exakt nachzubilden. □

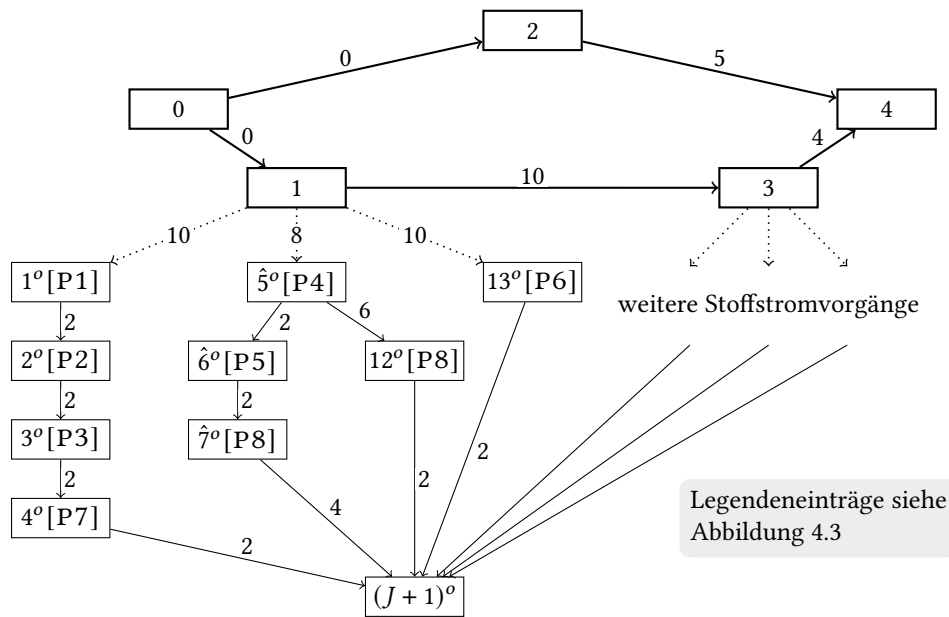


**Abbildung 4.8:** Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.3.

Abbildung 4.9 zeigt exemplarisch den Vorgangsknotennetzplan  $N^{temp}$  zu einem Projekt mit den Rückbauvorgängen  $i = 1, \dots, 4$ . Davon setzen Rückbauvorgang 1 und Rückbauvorgang 3 Stoffe frei. Rückbauvorgang 1 entspricht dem betrachteten Rückbauvorgang aus dem Beispiel 4.3. Die Kantengewichte entsprechen den Dauern der Rückbauvorgänge bzw. den zeitlichen Mindestabständen.

### 4.2.3 Strukturelle Eigenschaften von granularen und aggregierten Instanzen

Bei Betrachtung der Vorgangsknotennetzpläne in den vorangegangenen Abschnitten ist erkennbar, dass die durch Freisetzungsbeziehungen und stoffstrominduzierte Zeitbeziehungen hervorgerufene Struktur der Rückbau- und Stoffstromvorgänge einem besonderen Muster folgt. Zur Beschreibung dieser Struktur wird die Menge aller Rückbau- und Stoffstromvorgänge  $V \cup V^o$  in Teilmengen  $M_i := \{i\} \cup \overline{Succ^o}(i)$  aufgeteilt, wobei  $i \in V$  und  $\overline{Succ^o}(i) \subset V^o$ . Eine solche Teilmenge  $M_i$  wird als *Stoffstromstruktur* bezeichnet. Beispielsweise gilt in Abbildung 4.3,  $M_1 = \{1, 1^o, \dots, 13^o\}$  und in Abbildung 4.5,  $M_1 = \{1, 1^o, 2^o, 3^o, 4^o, \hat{5}^o, \hat{6}^o, \hat{7}^o, 12^o, 13^o\}$ . Der von einer Stoffstromstruktur  $M_i$  induzierte Teilgraph des Vorgangsknotennetzplans  $N^{temp}$  enthält alle Knoten aus  $M_i$  und die Kanten zwischen diesen Knoten. Falls das granulare oder aggregierte Transformationsschema verwendet wurde, ist dieser Teilgraph ein Out-Tree mit der Wurzel  $i$ , wie an den Abbildungen 4.3 und 4.5 zu erkennen ist. Außerdem gilt dann, dass alle Stoffstromstrukturen disjunkt sind. Für das granulare Transformationsschema lässt sich dies insofern weiter spezifizieren, als der von  $M_i$  induzierte Teilgraph ein Out-Tree ist, der nur an der Wurzel  $i$  verzweigt.



**Abbildung 4.9:** Exemplarischer Vorgangsknotennetzplan einer Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$ ; Rückbauvorgang 1 und Stoffstromvorgänge  $\overline{\text{Succ}}^o(1)$  entstammen dem Beispiel 4.3.

Out-Trees sind die grafische Darstellung einer divergierenden Struktur (vgl. Abschnitt 2.4.4). Deshalb werden die Stoffströme in den mit dem granularen und dem aggregierten Transformationsschema erzeugten Instanzen  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  und  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$  als *divergierende Stoffströme* bezeichnet. Für diese Arbeit wird definiert, dass die Stoffströme in einer Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  genau dann als divergierend gelten, falls die stoffstrominduzierte Zeitstruktur divergierend ist und die Flussershaltung gewährleistet ist. Formal muss gelten, dass für jeden Stoffstromvorgang  $j \in V^o \setminus \{(J+1)^o\}$  entweder genau ein Rückbauvorgang  $i$  existiert mit

1.  $(i, j) \in E^{\text{rel}}$ , d. h., Stoffstromvorgang  $j$  hat genau einen direkten Vorgänger  $i$ , und
2.  $f_{ik}^y(\cdot, \infty) + \sum_{j'' \in \overline{\text{Succ}}^o(i)} r_{j''k}^y(\cdot, \infty) = 0$  für jedes  $k \in \mathcal{R}^y$  mit  $f_{ik}^y(\cdot, \infty) > 0$ , d. h. jede von  $i$  eingelagerte Stoffeinheit wird durch seine transitiven Nachfolger wieder ausgelagert,

oder genau ein Stoffstromvorgang  $j' \neq j$  existiert mit

1.  $(j', j) \in E^{\text{low}}$ , d. h., Stoffstromvorgang  $j$  hat genau einen direkten Vorgänger  $j'$ , und
2.  $r_{j'k}^y(\cdot, \infty) + \sum_{j'' \in \overline{\text{Succ}}^o(j')} r_{j''k}^y(\cdot, \infty) = 0$  für jedes  $k \in \mathcal{R}^y$  mit  $r_{j'k}^y(\cdot, \infty) > 0$ , d. h., jede von  $j'$  eingelagerte Stoffeinheit wird durch seine transitiven Nachfolger wieder ausgelagert.

Die divergierenden Stoffströme werden bei der Entwicklung problemspezifischer Lösungsverfahren in Kapitel 5 eine zentrale Rolle spielen. Es ist zu beachten, dass das Attribut „divergierend“ die Stoffströme lediglich aus zeitlicher Perspektive beschreibt. Die zeitliche Perspektive ist nicht mit der räumlichen Perspektive zu verwechseln. Aus zeitlicher

Perspektive sind die Stoffströme in den Instanzen  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  und  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$  divergierend, während sie aus räumlicher Perspektive beliebig zusammen-, auseinander- oder in Zyklen fließen können. Zur grafischen Darstellung der zeitlichen Perspektive dienen Vorgangsknotennetzpläne (z. B. Abbildung 4.3), in denen die Kanten Vorrang- bzw. Anordnungsbeziehungen abbilden. Zur grafischen Darstellung der räumlichen Perspektive dienen Flussnetzwerke (z. B. Abbildung 3.5), in denen die Kanten physische Stoffflüsse abbilden. Der Unterschied zwischen beiden Darstellungsweisen ist auch daran zu erkennen, dass in Vorgangsknotennetzplänen keine Lager enthalten sind, wohingegen sie in Flussnetzwerken enthalten sein können.

Es ist zu beachten, dass die aus dem Produktionsmanagement bekannten Vergenztypen für Stoffströme nicht der zeitlichen Perspektive, sondern den zugrundeliegenden Produktionsstrukturen entstammen. Daher ist von der in Abschnitt 2.4.4 vorgenommenen Definition divergierender Stoffströme bei der hier betrachteten stoffstrombasierten Planung eines Rückbauprojekts Abstand zu nehmen.

#### 4.2.4 Auswirkungen der Aggregation auf die Realitätstreue

Zu Beginn des Abschnitts 4.2.2 wurde auf das Spannungsfeld zwischen Instanzgröße und Realitätstreue hingewiesen, in welchem sich die Ausgestaltungsmöglichkeiten des aggregierten Transformationsschemas bewegen. Beim Vergleich der in den Beispielen 4.2 und 4.3 erzeugten Instanzen ist erkennbar, dass die Zahl der Stoffstromvorgänge durch die Aggregation von 13 auf neun reduziert wurde. Diese Reduktion mag relativ geringfügig erscheinen. Bei großen Stofffreisetzungen kann mit der Aggregation allerdings eine deutliche Reduktion der Instanzgröße erzielt werden. Dadurch werden der benötigte Speicherplatz für eine Instanz, der benötigte Speicherplatz während der Laufzeit eines Lösungsverfahrens und die Anzahl der Iterationen des Lösungsverfahrens reduziert.

Unter Fortführung der Annahme, dass die granulare Instanz die reale Problemstellung am besten abbildet, hat die Aggregation eine nachteilige Auswirkung auf die Realitätstreue. Dies wurde damit begründet, dass aggregierte Vorgänge als nichtunterbrechbare Einheit eingeplant werden. Ob die verminderte Realitätstreue des Modells sich allerdings auf die Lösungsqualität auswirkt, hängt von der Instanzbeschaffenheit (d. h. dem Zusammenspiel aller Problemparameter einer Instanz) ab. Nachfolgend wird anhand verschiedener Beispiele demonstriert, welche Auswirkungen möglich sind.

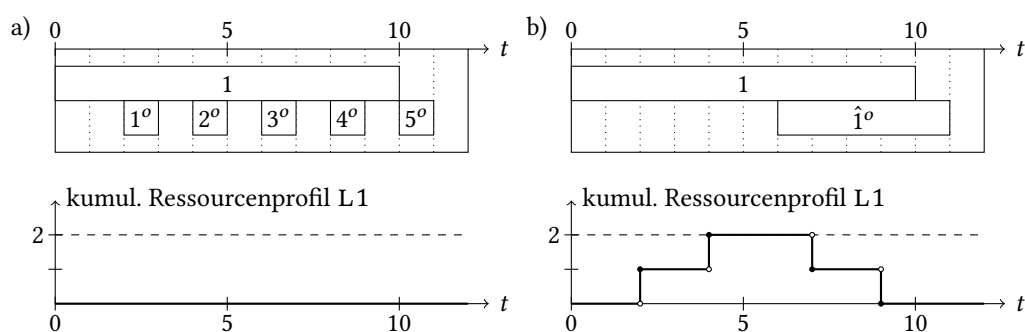
**Beispiel 4.4:** Eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  besteht aus zehn realen Rückbauvorgängen, die jeweils zehn Perioden dauern und jeweils gleichmäßig während ihrer Ausführung fünf Stoffeinheiten mit einem Stoffstrompfad  $W1 = (L1, P1, s)$  freisetzen. Es gilt somit  $d_i = 10$  und  $f_{i,W1}^{y,lin} = 5$  für  $i = 1, \dots, 10$ . Die Kapazität des Lagers  $L1$  beträgt  $R_{L1}^y = 2$  Stoffeinheiten. Die Verarbeitung einer Stoffeinheit in  $P1$  dauert eine Periode, d. h.  $d_{P1} = 1$ . Es existieren keine Vorrangbeziehungen und keine erneuerbaren Ressourcen.

Die mit TransformationSchemeGra erzeugte Instanz  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$  besteht aus den 50 granularen Stoffstromvorgängen  $j = 1^{\circ}, \dots, 50^{\circ}$ , wobei jeweils fünf davon direkt auf einen Rückbauvorgang  $i \in \{1, \dots, 10\}$  folgen. Jeder granuläre Stoffstromvorgang dauert eine Periode. Jeder Rückbauvorgang  $i = 1, \dots, 10$  setzt alle zwei Perioden eine Stoffeinheit frei, weshalb die zeitlichen Mindestabstände  $d_{ij}^{\text{rel}}$  der fünf auf  $i$  folgenden granularen Stoffstromvorgänge zwei, vier, sechs, acht und zehn Perioden betragen. Im optimalen Schedule zu  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$  kann jeder Stoffstromvorgang  $j$  zu seinem durch  $d_{ij}^{\text{rel}}$  vorgegebenen frühestmöglichen Startzeitpunkt starten und die freigesetzte Stoffeinheit direkt von seinem Vorgänger  $i$  übernehmen. Folglich bleiben die kumulativen Ressourcen über die gesamte Projektdauer leer und alle Rückbauvorgänge können zum Zeitpunkt null starten, d. h.  $S_i = 0$  für  $i = 1, \dots, 10$ . Der optimale Zielfunktionswert beträgt zehn Perioden (= Endzeitpunkt aller Rückbauvorgänge; Stoffstromvorgänge werden nicht berücksichtigt; vgl. Problemformulierung des (RMCPSP)).

Die mit TransformationSchemeAgg erzeugte Instanz  $\mathcal{I}_{\text{RCSP}/M\text{-agg}}$  besteht aus den 10 aggregierten Stoffstromvorgängen  $j = \hat{1}^{\circ}, \dots, \hat{10}^{\circ}$ , wobei jeweils einer davon direkt auf einen Rückbauvorgang  $i \in \{1, \dots, 10\}$  folgt. Jeder aggregierte Stoffstromvorgang dauert fünf Perioden. Gemäß Gleichung (4.17) betragen die zeitlichen Mindestabstände  $d_{ij}^{\text{rel}}$  zwischen jedem Rückbauvorgang  $i = 1, \dots, 10$  und seinem nachfolgenden Stoffstromvorgang  $j$  jeweils sechs Perioden. Nur so ist die rücktransformierte Lösung auch eine zulässige Lösung für die zugehörige granuläre Instanz. Im Zeitintervall vom Start jedes Rückbauvorgangs bis zum Start seines nachfolgenden aggregierten Stoffstromvorgangs müssen die freigesetzten Stoffeinheiten in der kumulativen Ressource L1 gepuffert werden. Die von einem Rückbauvorgang  $i$  stammende gepufferte Stoffmenge erreicht jeweils im Intervall  $\tau \in [4, 7)$  nach dem Start von  $i$  ihr Maximum von zwei Stoffeinheiten. Ein weiterer Rückbauvorgang  $i'$  kann daher frühestens fünf Perioden nach Rückbauvorgang  $i$  starten, weil die erste Stoffeinheit von  $i'$  zwei Perioden nach seinem Start eingelagert wird und damit sieben Perioden nach  $i$ . Dadurch fällt die Einlagerung mit dem Ende des Intervalls  $[4, 7)$  zusammen, d. h., die Einlagerung der Stoffeinheiten von  $i'$  und die Auslagerungen von  $i$  heben sich auf. Im optimalen Schedule zu  $\mathcal{I}_{\text{RCSP}/M\text{-agg}}$  starten folglich alle Rückbauvorgänge um fünf Perioden gegeneinander versetzt. Der optimale Zielfunktionswert beträgt  $5 \cdot 9 + 10$  (Dauer des letzten Rückbauvorgangs) = 55 Perioden und ist damit um 450 % höher als der optimale Zielfunktionswert zur granularen Instanz.

Abbildung 4.10 zeigt einen Ausschnitt aus den Gantt-Diagrammen der granularen und der aggregierten Instanz. Der Ausschnitt umfasst Rückbauvorgang 1 und seine nachfolgenden Stoffstromvorgänge. Unterhalb der Gantt-Diagramme ist die durch diese Vorgänge verursachte Veränderung des kumulativen Ressourcenprofils von L1 dargestellt. Es ist ersichtlich, dass in der granularen Instanz keine Einlagerungen erfolgen (vgl. Abbildung 4.10a). Dagegen wird in der aggregierten Instanz die kumulative Ressourcenkapazität im Intervall  $t \in [4, 7)$  voll ausgeschöpft, weshalb der nächste Rückbauvorgang 2 frühestens zum Zeitpunkt fünf starten kann (vgl. Abbildung 4.10b).  $\square$

**Beispiel 4.5:** Wie in Beispiel 4.4 besteht eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  aus zehn realen Rückbauvorgängen, die jeweils zehn Perioden dauern und jeweils gleichmäßig während ihrer



**Abbildung 4.10:** Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.4: a) granulare Instanz; b) aggregierte Instanz.

Ausführung fünf Stoffeinheiten mit einem Stoffstrompfad  $W1 = (L1, P1, s)$  freisetzen. Die Kapazität des Lagers L1 beträgt  $R_{L1}^y = 2$  Stoffeinheiten. Im Gegensatz zu Beispiel 4.4 dauert die Verarbeitung einer Stoffeinheit in P1 nun zwei Perioden, d. h.  $d_{P1} = 2$ . Es existieren keine Vorrangbeziehungen und keine erneuerbaren Ressourcen.

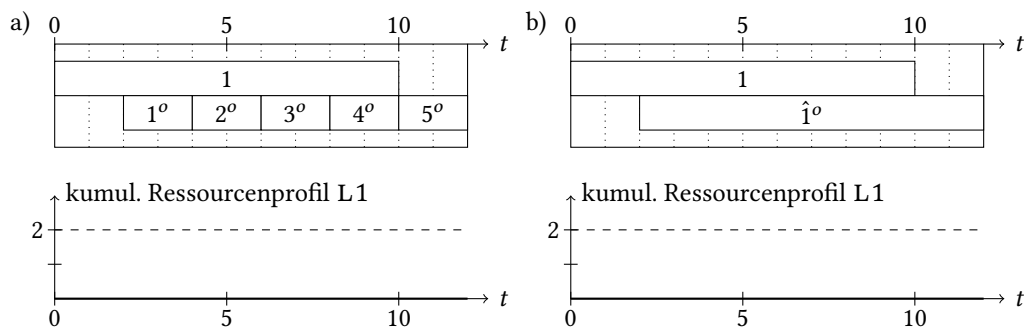
Die mit TransformationSchemeGra erzeugte Instanz  $\mathcal{I}_{RCPSM/M-gra}$  unterscheidet sich von Beispiel 4.4 lediglich dadurch, dass die 50 granularen Stoffstromvorgänge nun jeweils zwei Perioden dauern. Der optimale Zielfunktionswert beträgt weiterhin zehn Perioden, da das Lager L1 durchgängig leer bleibt.

Die mit TransformationSchemeAgg erzeugte Instanz  $\mathcal{I}_{RCPSM/M-agg}$  unterscheidet sich von Beispiel 4.4 dadurch, dass die zehn aggregierten Stoffstromvorgänge nun jeweils zehn Perioden dauern. Gemäß Gleichung (4.17) betragen die zeitlichen Mindestabstände  $d_{ij}^{rel}$  zwischen jedem Rückbauvorgang  $i = 1, \dots, 10$  und seinem nachfolgenden Stoffstromvorgang  $j$  jeweils zwei Perioden. Aufgrund der übereinstimmenden Dauern von Rückbau- und nachfolgenden Stoffstromvorgängen heben sich die Ein- und Auslagerungen gegenseitig auf. Es müssen keine Stoffe mehr in L1 gepuffert werden, weshalb  $S_i = 0$  für alle Rückbauvorgänge  $i = 1, \dots, 10$  gilt. Der optimale Zielfunktionswert beträgt wie in  $\mathcal{I}_{RCPSM/M-gra}$  zehn Perioden.

Abbildung 4.11 zeigt hierzu den Ausschnitt aus den Gantt-Diagrammen und die Veränderungen der zugehörigen kumulativen Ressourcenprofile, die in beiden Fällen bei null bleiben.  $\square$

Die Gegenüberstellung der Beispiele 4.4 und 4.5 verdeutlicht, dass das Verhältnis zwischen der Dauer des einlagernden Vorgangs und der Dauer seiner auslagernden Nachfolger einen Einfluss auf die Lösungsqualität haben kann. In Beispiel 4.4 werden Stoffe schneller freigesetzt, als sie verarbeitet werden. Wenn das Lager leer gehalten werden soll, muss eine direkte Übergabe von Stoffeinheiten vom freisetzenden Rückbauvorgang an die auslagernden Stoffstromvorgänge modelliert werden können. Dies ist nur mit granularen Stoffstromvorgängen möglich, zwischen denen es Leerlaufzeiten geben kann. Mit aggregierten Stoffstromvorgängen entstehen hingegen künstliche Pufferlagerbestände. In Beispiel 4.5 erfolgen Stofffreisetzung und -verarbeitung synchron. Es ist in diesem Fall





**Abbildung 4.11:** Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.5: a) granulare Instanz; b) aggregierte Instanz.

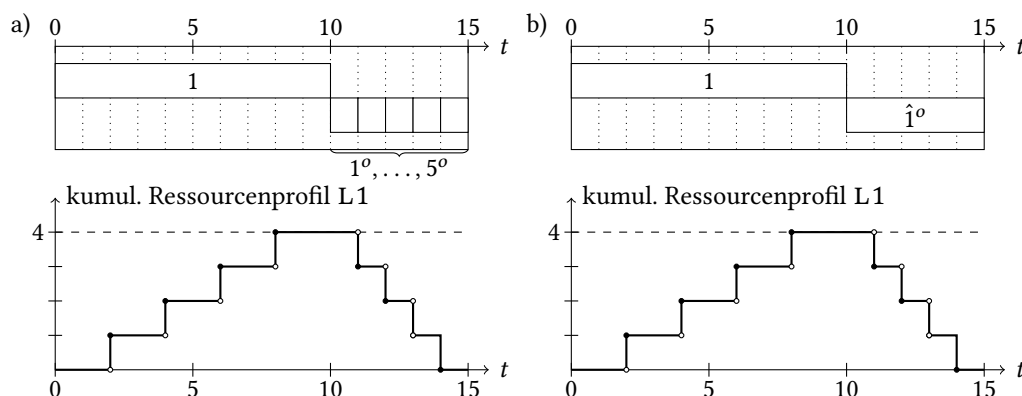
auch mit aggregierten Stoffstromvorgängen möglich, die direkte Übergabe von Stoffeinheiten zu modellieren und damit die Lager leerzuhalten. Besonders hervorzuheben ist, dass in Beispiel 4.4 die Stoffverarbeitung zwar schneller erfolgt als in Beispiel 4.5, aber der Zielfunktionswert zur aggregierten Instanz größer ist. Dies ist eine direkte Folge der verminderten Realitätstreue der aggregierten Instanz und stellt eine Schwäche des aggregierten Transformationsschemas dar.

Dennoch kann das Verhältnis zwischen der Dauer des einlagernden Vorgangs und der Dauer seiner auslagernden Nachfolger nicht als allgemeingültiger Indikator für die Auswirkungen einer Aggregation auf die Lösungsqualität herangezogen werden, wie nachfolgendes Beispiel zeigt.

**Beispiel 4.6:** Die in Beispiel 4.4 beschriebene Instanz  $\mathcal{I}_{\text{RCMCPSP}}$  besteht nun zusätzlich aus einer erneuerbaren Ressource R1 mit Kapazität  $R_{\text{R1}}^\alpha = 1$ . Alle Rückbauvorgänge und Verarbeitungsschritt P1 benötigen eine Einheit dieser erneuerbaren Ressource, d. h.  $r_{i,\text{R1}}^\alpha = 1$  für  $i = 1, \dots, 10$  und  $r_{\text{P1},\text{R1}}^\alpha = 1$ . Die Kapazität des Lagers L1 beträgt nun  $R_{\text{L1}}^\gamma = 4$ . Andere Problemparameter bleiben gegenüber Beispiel 4.4 unverändert.

Die mit TransformationSchemeGra und TransformationSchemeAgg erzeugten Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  und  $\mathcal{I}_{\text{RCPSP/M-agg}}$  sind in ihrer zeitlichen Struktur gleich wie in Beispiel 4.4. Den Stoffstromvorgängen wurde die Ressourcennutzung von Verarbeitungsschritt P1 übergeben, d. h.  $r_{j,\text{R1}}^\alpha = 1$  für alle granularen Stoffstromvorgänge  $j = 1^\circ, \dots, 50^\circ$  bzw. alle aggregierten Stoffstromvorgänge  $j = \hat{1}^\circ, \dots, \hat{10}^\circ$ . Die erneuerbaren Ressourcenrestriktionen bedeuten für den Schedule, dass keine Vorgänge parallel ausgeführt werden können. Folglich muss sowohl bei der granularen als auch bei der aggregierten Instanz auf das Ende eines Rückbauvorgangs gewartet werden, bis die Stoffverarbeitung beginnen kann. Im optimalen Schedule des (RCPSP/M) zur granularen und zur aggregierten Instanz starten alle Rückbauvorgänge um 15 Perioden (= Dauer eines Rückbauvorgangs + Dauer seiner nachfolgenden Stoffstromvorgänge) gegeneinander versetzt. Der optimale Zielfunktionswert beträgt  $15 \cdot 9 + 10$  (Dauer des letzten Rückbauvorgangs) = 145 Perioden.

Wie Abbildung 4.12 zeigt, müssen für jeden Rückbauvorgang vier Stoffeinheiten in L1 gepuffert werden. Mit einer Kapazität  $R_{R1}^Y < 4$  wäre sowohl die granulare als auch die aggregierte Instanz unlösbar. □



**Abbildung 4.12:** Gantt-Diagramme und Veränderungen der kumulativen Ressourcenprofile zu Beispiel 4.6: a) granulare Instanz; b) aggregierte Instanz.

Die zusätzliche Einführung einer erneuerbaren Ressource in Beispiel 4.6 hat dazu geführt, dass für die granulare und die aggregierte Instanz ein Schedule mit derselben Qualität berechnet werden konnte. Aufgrund der erneuerbaren Ressourcenrestriktionen sind Schedules, in denen Stoffe direkt vom Rückbauvorgang zum Stoffstromvorgang übergeben werden, ohnehin unzulässig. Die erneuerbare Ressource hat demnach die Auswirkungen des Verhältnisses zwischen der Dauer des einlagernden Vorgangs und der Dauer seiner auslagernden Nachfolger obsolet gemacht.

In Bezug auf Beispiel 4.4 (vgl. Abbildung 4.10) könnte auch argumentiert werden, das aggregierte Transformationsschema folgendermaßen umzugestalten: Es hätten auch aggregierte Stoffstromvorgänge der Dauer zehn erzeugt werden können. Damit wäre es möglich, die direkte Übergabe von Stoffeinheiten zwischen Rückbauvorgang und Stoffstromvorgang zu modellieren. Für die in Beispiel 4.4 beschriebene Entscheidungssituation hätte diese Umgestaltung keine negativen Konsequenzen auf die Realitätstreue. Bei Vorhandensein einer erneuerbaren Ressource, wie in Beispiel 4.6, würde dies jedoch das reale Problem verfälschen. Dann würde die erneuerbare Ressource über zehn Perioden, statt über die realen fünf Perioden, welche die Stoffverarbeitung eigentlich benötigt, genutzt werden. Sobald Rückbau- und Stoffstromvorgänge nicht mehr parallel ausgeführt werden, kann die um fünf Perioden überhöhte Dauer außerdem zu künstlichen Verzögerungen im Schedule führen.

In den bisherigen Beispielen war lediglich ein einzelnes Lager vorhanden und die Verarbeitung aller Stoffe hat gleich lange gedauert. Wenn mehrere Lager vorhanden sind oder Stoffverarbeitungen unterschiedlich lange dauern, kann es vorteilhaft sein, die Verarbeitung bestimmter Stoffe zu unterbrechen, um die Verarbeitung anderer Stoffe vorzuziehen. Ein solcher Fall wird in Beispiel 4.7 skizziert.

**Beispiel 4.7:** Eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  besteht aus den beiden realen Rückbauvorgängen 1 und 2, die jeweils vier Perioden dauern. Es existiert eine Vorrangbeziehung (1, 2). Es gilt  $f_{1,W1}^{Y,start} = 3$  mit  $W1 = (L1, P1, s)$  und  $f_{2,W2}^{Y,start} = 3$  mit  $W2 = (L2, P1, s)$ . Die Kapazitäten der beiden Lager betragen  $R_{L1}^Y = R_{L2}^Y = 2$ . Die Verarbeitung einer Stoffeinheit in P1 dauert eine Periode und benötigt eine Einheit einer erneuerbaren Ressource R1, d. h.  $d_{P1} = 1$  und  $r_{P1,R1}^\alpha = 1$ . Die Kapazität von R1 beträgt  $R_{R1}^\alpha = 1$ . Die Rückbauvorgänge benötigen keine erneuerbaren Ressourcen.

Sowohl für granulare als auch für aggregierte Stoffstromvorgänge gilt ein zeitlicher Mindestabstand von null, da die Stoffe jeweils zum Start eines Rückbauvorgangs freigesetzt werden. Aufgrund der erneuerbaren Ressourcenrestriktionen können Stoffstromvorgänge nicht parallel ausgeführt werden. Außerdem muss aufgrund der kumulativen Ressourcenrestriktionen immer mindestens ein Stoffstromvorgang zeitgleich mit seinem vorangehenden Rückbauvorgang starten. Nur durch die damit erfolgte direkte Übergabe einer der drei freigesetzten Stoffeinheiten können  $R_{L1}^Y = R_{L2}^Y = 2$  eingehalten werden.

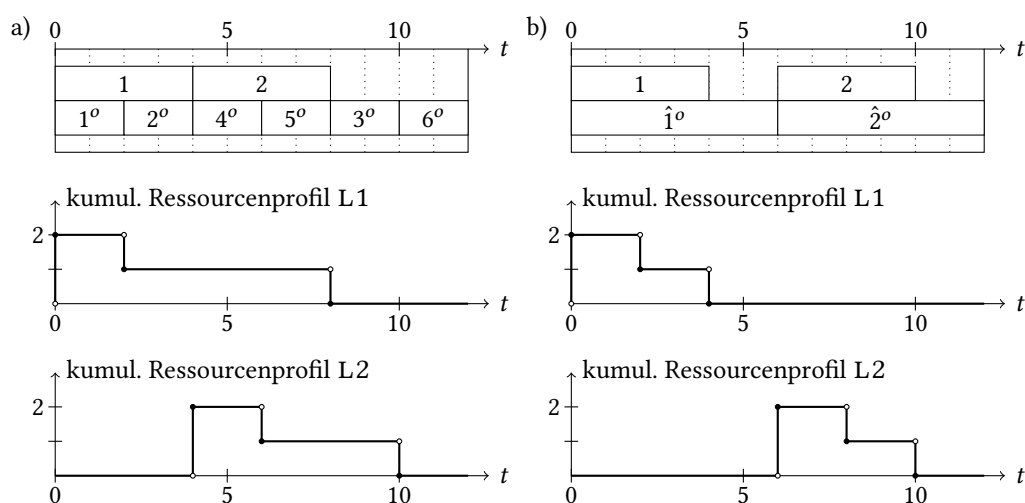
In der mit TransformationSchemeGra erzeugten Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  folgen auf Rückbauvorgang 1 drei granulare Stoffstromvorgänge  $1^\circ, 2^\circ, 3^\circ$  und auf Rückbauvorgang 2 drei granulare Stoffstromvorgänge  $4^\circ, 5^\circ, 6^\circ$ . Jeder dieser granularen Stoffstromvorgänge dauert zwei Perioden. Ein optimaler Schedule ist  $S_1 = 0, S_{1^\circ} = 0, S_{2^\circ} = 2, S_{3^\circ} = 8, S_2 = 4, S_{4^\circ} = 4, S_{5^\circ} = 6$  und  $S_{6^\circ} = 10$ , da die beiden Rückbauvorgänge zu ihren frühestmöglichen Startzeitpunkten starten. Der optimale Zielfunktionswert beträgt acht Perioden.

In der mit TransformationSchemeAgg erzeugten Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$  folgt auf Rückbauvorgang 1 ein aggregierter Stoffstromvorgang  $\hat{1}^\circ$  und auf Rückbauvorgang 2 ein aggregierter Stoffstromvorgang  $\hat{2}^\circ$ . Jeder dieser aggregierten Stoffstromvorgänge dauert sechs Perioden. Der optimale Schedule ist  $S_1 = S_{\hat{1}^\circ} = 0$  und  $S_2 = S_{\hat{2}^\circ} = 6$ . Es ist nicht möglich, Rückbauvorgang 2 früher zu starten, da sein nachfolgender Stoffstromvorgang zeitgleich mit ihm starten muss. Dies ist aufgrund der erneuerbaren Ressourcenrestriktionen frühestens zum Zeitpunkt sechs möglich. Der optimale Zielfunktionswert beträgt zehn Perioden.

Abbildung 4.13 zeigt die Gantt-Diagramme der granularen und der aggregierten Instanz. Unterhalb der Gantt-Diagramme sind die zugehörigen kumulativen Ressourcenprofile von L1 und L2 dargestellt.  $\square$

In der granularen Instanz aus Beispiel 4.7 wurde die Verarbeitung der Stoffe aus Rückbauvorgang 1 zum Zeitpunkt 4 unterbrochen, um Stoffe aus Rückbauvorgang 2 zu verarbeiten und damit den Start von Rückbauvorgang 2 vorzuziehen. Dies ist in der aggregierten Instanz nicht möglich. Obwohl keine Leerlaufzeiten zwischen den Stoffstromvorgängen existieren und augenscheinlich ein Verschmelzen der granularen Stoffstromvorgänge im Gantt-Diagramm möglich wäre, wird die Lösungsqualität durch die fehlende Unterbrechbarkeit infolge einer Aggregation gemindert.

Zusammenfassend lässt sich festhalten, dass die möglichen Auswirkungen der verminderten Realitätstreue auf die Lösungsqualität komplex sind. Sie können nicht auf analytische Weise beschrieben werden, sondern hängen von der individuellen Instanzbeschaffenheit



**Abbildung 4.13:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 4.7: a) granulare Instanz; b) aggregierte Instanz.

ab. Deshalb ist ein experimenteller Vergleich von granulemem und aggregiertem Transformationsschema, wie er in Kapitel 6 erfolgt, erforderlich.

#### 4.2.5 Weitere Transformationsschemas

Wie bereits mehrfach angemerkt, sind TransformationSchemeGra und TransformationSchemeAgg nur zwei Vorschläge für konkrete Ausgestaltungsmöglichkeiten der Transformation einer Instanz  $\mathcal{I}_{\text{RMCPSP}}$  in eine Instanz  $\mathcal{I}_{\text{RCPSM/M}}$ . Sie können auch als Extremfälle mit maximal granularen bzw. maximal aggregierten Stoffstromvorgängen betrachtet werden. Dazwischen sind beliebige weitere Abstufungen denkbar. Beispielsweise kann eine Obergrenze für die maximale Anzahl an aggregierbaren granularen Stoffstromvorgängen vorgegeben werden, die verhindert, dass aggregierte Stoffstromvorgänge zu groß werden. Damit könnte das Spannungsfeld zwischen Instanzgröße und Realitätstreue weiter ausgelotet werden.

Auch Annahmen, die zur Formulierung der Transformationsschemas getroffen wurden, können aufgehoben werden. Wenn es beispielsweise aus technischen oder organisatorischen Gründen nicht möglich wäre, die Verarbeitung zwischen jeder einzelnen Stoffeinheit zu unterbrechen, kann das granulare Transformationsschema entsprechend angepasst oder sogar von der Anwendung ausgeschlossen werden. Wenn Verarbeitungsschritte je nach Stoffart unterschiedliche Dauern aufweisen, können die Transformationsschemas stoffartspezifisch ausgestaltet werden. Es können auch nichtlineare Effekte bei der Verarbeitungsdauer berücksichtigt werden, wie beispielsweise Lernkurven, die die Dauern späterer Stoffstromvorgänge verkürzen.

Auf die getroffene Annahme, dass die Dauer der Stoffverarbeitung linear mit der verarbeiteten Stoffmenge skaliert, wird nachfolgend ausführlicher eingegangen. Damit soll einerseits

die Plausibilität dieser Annahme für die in dieser Arbeit betrachtete Problemstellung begründet werden und andererseits eine alternative Ausgestaltungsmöglichkeit für ein aggregiertes Transformationsschema aufgezeigt werden.

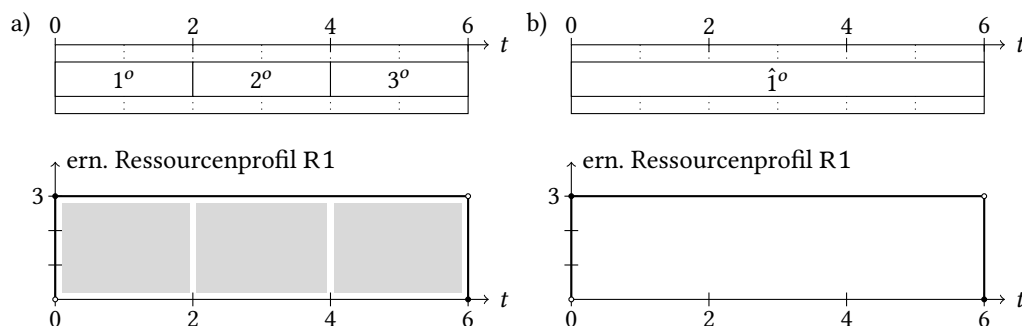
Mit der linearen Skalierbarkeit der Verarbeitungsdauer wurde in Abschnitt 4.2.2 die horizontale Aggregation begründet, d. h. ein Verschmelzen von granularen Stoffstromvorgängen im Gantt-Diagramm, indem die Dauer aufsummiert wird, während die Ressourcennutzungen beibehalten werden. Die Annahme der linearen Skalierbarkeit basiert auf der strategischen Betrachtungsweise, mit der der Stoffstrombereich geplant wird. Hierbei wird die Verarbeitung der Stoffeinheiten im Gegensatz zur operativen Maschinenbelegungsplanung nicht mit exakter zeitlicher Auflösung betrachtet. Stattdessen wird der Stoffstrombereich so modelliert, dass mindestens eine Verarbeitungskapazität (z. B. Maschinen oder Personal) unter Volllast arbeitet und somit ein höherer Durchsatz nur durch eine höhere Verarbeitungsdauer erzielt werden kann. Konkret bedeutet dies, dass die Problemparameter für das Transformationsschema folgendermaßen zu gestalten sind: Die Dauer  $d_p$  der Verarbeitung einer Stoffeinheit in jedem Verarbeitungsschritt  $p \in \mathcal{P}$  wird modellexogen so bestimmt, dass sie der Verarbeitungsdauer unter Volllast entspricht. Entsprechend muss für mindestens eine erneuerbare Ressource  $k \in \mathcal{R}^\alpha$  die erneuerbare Ressourcennutzung  $r_{pk}^\alpha$  der Ressourcenkapazität  $R_k^\alpha$  gleichen. In einem Schedule ist es dann nicht möglich, Stoffstromvorgänge, die von  $p$  abgeleitet wurden, parallel auszuführen. Die Dauer der Stoffverarbeitung skaliert daher linear mit der verarbeiteten Stoffmenge, sodass eine horizontale Aggregation sinnvoll ist. Beispiel 4.8 verdeutlicht diesen Zusammenhang.

**Beispiel 4.8:** In einem kerntechnischen Rückbauprojekt gibt es den Verarbeitungsschritt  $P1 = \text{„Sandstrahlen“}$ . Zum Sandstrahlen einer Stoffeinheit werden eine Sandstrahlstation und eine Fachkraft benötigt. Sandstrahlstationen und Fachkräfte werden als erneuerbare Ressource  $R1$  bzw.  $R2$  modelliert. Es stehen insgesamt drei Sandstrahlstationen und zehn Fachkräfte zur Verfügung, d.h.  $R_{R1}^\alpha = 3$  und  $R_{R2}^\alpha = 10$ . Das Sandstrahlen einer Stoffeinheit an einer Sandstrahlstation durch eine Fachkraft dauert sechs Perioden.

Zur Verarbeitung einer Stoffeinheit unter Volllast (d. h. unter Verwendung aller verfügbaren Sandstrahlstationen und Fachkräfte), müsste sie auf die drei Sandstrahlstationen aufgeteilt werden, welche den Engpass für das Sandstrahlen darstellen. Damit wäre ihre rechnerische Verarbeitungsdauer zwei Perioden. Fachkräfte stellen keinen Engpass dar. Die Problemparameter für die Transformation sind folglich  $d_{P1} := 2$ ,  $r_{P1,R1}^\alpha := 3$  und  $r_{P1,R2}^\alpha := 3$ . Granulare Stoffstromvorgänge können dann, wie in TransformationSchemeAgg beschrieben, horizontal aggregiert werden, da ihre parallele Ausführung aufgrund von  $R_{R1}^\alpha = 3$  ohnehin nicht möglich ist.

Abbildung 4.14 zeigt dazu ein exemplarisches Gantt-Diagramm einer granularen und einer horizontal aggregierten Instanz, in welcher drei Stoffeinheiten sandgestrahlt werden. Die drei von  $P1$  abgeleiteten granularen Stoffstromvorgänge  $1^o, 2^o, 3^o$  dauern jeweils zwei Perioden und nutzen jeweils drei Sandstrahlstationen (= erneuerbare Ressource  $R1$ ). Die Dauer des horizontal aggregierten Stoffstromvorgangs  $\hat{1}^o$  entspricht der Summe der granularen Dauern. Seine Ressourcennutzung gleicht den granularen Ressourcennutzungen.

Das erneuerbare Ressourcenprofil für die Sandstrahlstationen ist unterhalb der Gantt-Diagramme dargestellt, wobei die grauen Flächen in Abbildung 4.14a die Beiträge der einzelnen granularen Stoffstromvorgänge kenntlich machen. □



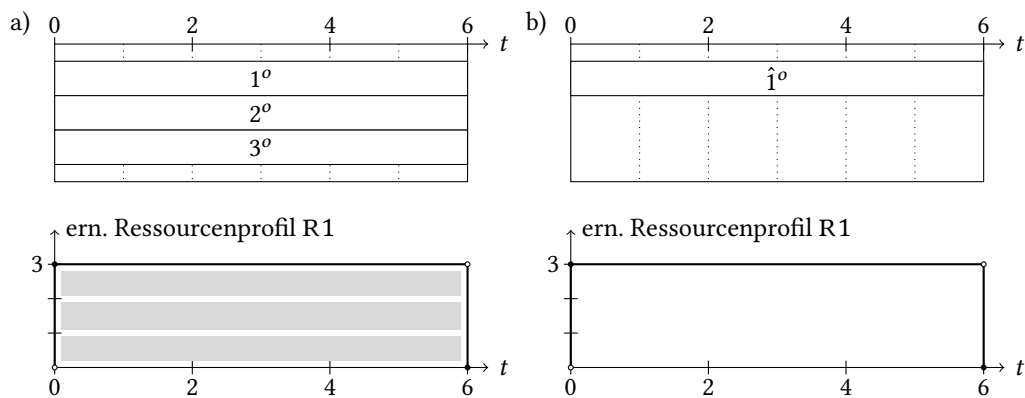
**Abbildung 4.14:** Gantt-Diagramme und erneuerbare Ressourcenprofile zu Beispiel 4.8: a) granulare Instanz; b) horizontal aggregierte Instanz.

Alternativ zur beschriebenen Modellierung könnte das aggregierte Transformationsschema derart umgestaltet werden, dass eine *vertikale Aggregation* stattfindet. Darunter wird ein Verschmelzen von granularen Stoffstromvorgängen im Gantt-Diagramm verstanden, indem Ressourcennutzungen aufsummiert werden, während die Dauer beibehalten wird. Beispiel 4.9 verdeutlicht den Grundgedanken der vertikalen Aggregation.

**Beispiel 4.9 (Fortsetzung des Beispiels 4.8):** Das Sandstrahlen einer Stoffeinheit aus Beispiel 4.8 wird nun nicht unter Volllast, sondern realitätsgetreu jeweils auf einer Sandstrahlstation mit einer Fachkraft modelliert. Die Problemparameter für die Transformation werden daher auf  $d_{P1} := 6$ ,  $r_{P1,R1}^\alpha := 1$  und  $r_{P1,R2}^\alpha := 1$  gesetzt. TransformationSchemeAgg würde die reale Problemstellung verfälschen, da durch die horizontale Aggregation 18 Perioden für das Sandstrahlen benötigt würden. Deshalb muss TransformationSchemeAgg so abgeändert werden, dass je maximal drei granulare Stoffstromvorgänge vertikal aggregiert werden, d. h. drei Perioden als Dauer beibehalten wird und die erneuerbaren Ressourcennutzungen der granularen Stoffstromvorgänge aufsummiert werden.

Abbildung 4.15 zeigt dazu ein exemplarisches Gantt-Diagramm einer granularen und einer vertikal aggregierten Instanz, in welcher drei Stoffeinheiten sandgestrahlt werden. Die drei von P1 abgeleiteten granularen Stoffstromvorgänge  $1^\circ$ ,  $2^\circ$ ,  $3^\circ$  dauern jeweils sechs Perioden und nutzen jeweils eine Sandstrahlstation (= erneuerbare Ressource R1). Die Dauer des vertikal aggregierten Stoffstromvorgangs  $\hat{1}^\circ$  gleicht den granularen Dauern. Seine Ressourcennutzung entspricht der Summe der granularen Ressourcennutzungen. Das erneuerbare Ressourcenprofil für die Sandstrahlstationen ist unterhalb der Gantt-Diagramme dargestellt, wobei die grauen Flächen in Abbildung 4.15a die Beiträge der einzelnen granularen Stoffstromvorgänge kenntlich machen. □

Beim Vergleich der Abbildungen 4.14 und 4.15 ist zu erkennen, dass die mit der horizontalen und der vertikalen Aggregation erzeugten Instanzen übereinstimmen. Auch wenn



**Abbildung 4.15:** Gantt-Diagramme und erneuerbare Ressourcenprofile zu Beispiel 4.9: a) granulare Instanz; b) vertikal aggregierte Instanz.

das Sandstrahlen einer einzelnen, unteilbaren Stoffeinheit unter Volllast mit einer rechnerischen Dauer von zwei Perioden nicht realitätsgetreu ist, wird die Realitätstreue des Modells bei drei sandgestrahlten Stoffeinheiten wiederhergestellt. Für große Stoffmengen und eng beschränkte Verarbeitungskapazitäten nähern sich Instanzen der horizontalen und der vertikalen Aggregation hinsichtlich ihrer Realitätstreue an. Daher beschränkt sich diese Arbeit im weiteren Verlauf auf die horizontale Aggregation, welche in TransformationSchemeAgg umgesetzt ist. Falls die Verarbeitungskapazitäten nicht so eng binden, wie in den Beispielen 4.8 und 4.9 im Fall der Sandstrahlstationen, sollte jedoch auch die vertikale Aggregation in Betracht gezogen werden. Dann können vertikal aggregierte Stoffstromvorgänge die erneuerbaren Ressourcennutzungen realitätsgetruer abbilden, da diese Nutzungen nicht wie bei horizontal aggregierten Stoffstromvorgängen immer bis zur Kapazitätsgrenze hochskaliert wurden.

Abschließend sei erneut darauf hingewiesen, dass sämtliche im Zusammenhang mit der Ausgestaltung der Transformationsschemas getroffenen Annahmen und Entscheidungen die Allgemeinheit des Schedulingproblems (RCPSP/M) nicht einschränken. Es handelt sich lediglich um Annahmen zur Struktur der Probleminstanzen. Insbesondere die Annahme, dass das granulare Transformationsschema die reale Problemstellung am besten beschreibt, muss nicht notwendigerweise zutreffen, um die Anwendbarkeit der vorgestellten Konzepte zu gewährleisten.

### 4.3 Problemcharakteristika

Dieser Abschnitt dient dazu, das Verständnis des (RCPSP/M) zu schärfen und die vorgenommene Abgrenzung von der Literatur in Abschnitt 3.5 zu vertiefen. Er schließt inhaltlich an Abschnitt 4.1 an und befasst sich mit allgemeinen Charakteristika des (RCPSP/M), die unabhängig von der Anwendung eines Transformationsschemas zu beobachten sind. Dennoch wurde es als sinnvoll erachtet, die Vorstellung der Transformationsschemas vor

diesem Abschnitt anzusiedeln, da insbesondere die Abbildungen und Beispiele in Abschnitt 4.2 eine hilfreiche Grundlage für die nachfolgenden Darlegungen bilden.

### 4.3.1 Notwendigkeit der stoffstrominduzierten Zeitstruktur

Beim Vergleich mit existierenden Ansätzen in der Literatur in Abschnitt 3.5.1 wurde die Notwendigkeit der stoffstrominduzierten Zeitstruktur damit begründet, dass in Rückbauprojekten inhomogene Stoffe vorliegen. Inhomogene Stoffe unterscheiden sich hinsichtlich der Verarbeitungsschritte und Lager, die sie durchlaufen. Deshalb werden die Stoffstromvorgänge modellexogen mittels Freisetzungsbeziehungen und stoffstrominduzierten Zeitbeziehungen in eine feste zeitliche Struktur eingebettet. Im Vergleich dazu werden bei produktionswirtschaftlichen Problemstellungen häufig homogene Stoffströme betrachtet, die nach demselben Muster verarbeitet werden. In einer solchen Anwendungsumgebung wird keine stoffstrominduzierte Zeitstruktur benötigt, sondern die zeitliche Anordnung der Stoffstromvorgänge kann modellendogen über die Einhaltung der Höchst- und Mindestbestände der Lager sichergestellt werden (vgl. Schwindt und Trautmann 2000, S. 506). Im nachfolgenden Beispiel wird dieser Unterschied veranschaulicht.

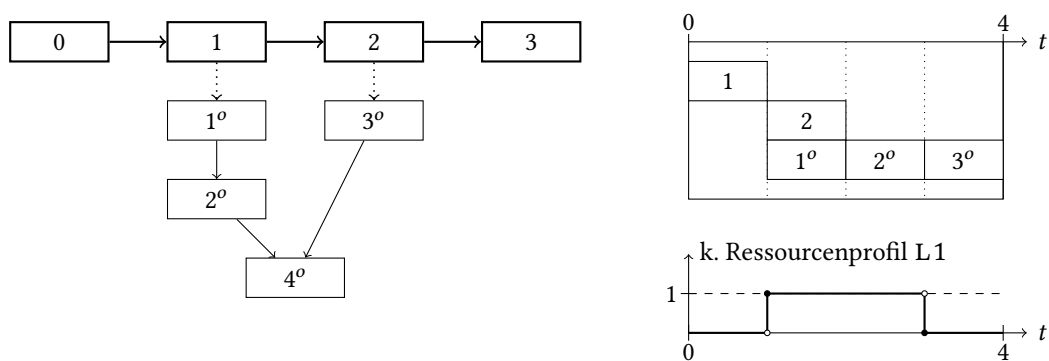
**Beispiel 4.10:** In einem Rückbauprojekt wird zuerst eine Wand 1 und danach eine Wand 2 abgebaut. Der Bauschutt von Wand 1 muss noch nachbehandelt werden, bevor er abtransportiert wird. Der Bauschutt von Wand 2 kann sofort abtransportiert werden. Der verfügbare Platz reicht nur für den Bauschutt einer Wand aus, d. h., es muss zuerst ein Abtransport des Bauschutts von Wand 1 erfolgen, bevor Wand 2 vollständig abgebaut werden kann. Der Rückbau, die Nachbehandlung und der Abtransport dauern jeweils eine Periode. Aufgrund beschränkter Verarbeitungskapazitäten können Nachbehandlung und die Abtransporte nicht parallel erfolgen.

Zur Modellierung dieser Problemstellung werden die beiden Rückbauvorgänge  $1 =$  „Abbau von Wand 1“ und  $2 =$  „Abbau von Wand 2“ erzeugt und mit einer Vorrangbeziehung verknüpft. Auf Rückbauvorgang 1 folgt Stoffstromvorgang  $1^o =$  „Nachbehandlung“ und auf Stoffstromvorgang  $1^o$  folgt Stoffstromvorgang  $2^o =$  „Abtransport“. Auf Rückbauvorgang 2 folgt Stoffstromvorgang  $3^o =$  „Abtransport“. Alle Rückbau- und Stoffstromvorgänge dauern eine Periode. Der Bauschutt jeder Wand wird als eine Stoffeinheit betrachtet und der verfügbare Platz wird als eine kumulative Ressource  $L1$  mit Kapazität  $R_{L1}^Y = 1$  modelliert. Somit setzen die beiden Rückbauvorgänge je eine Stoffeinheit in  $L1$  frei. Die beiden Stoffstromvorgänge  $2^o$  und  $3^o$  lagern je eine Stoffeinheit aus, die nach ihrer Ausführung die Senke erreicht. Stoffstromvorgang  $1^o$  verursacht keine Ein- oder Auslagerungen, d. h., er ist neutral. Es wird davon ausgegangen, dass Einlagerungen zum Vorgangsende und Auslagerungen zum Vorgangstart erfolgen. Da die stoffstrominduzierte Zeitstruktur in Einklang mit diesen kumulativen Ressourcennutzungen stehen muss, werden alle zeitlichen Mindestabstände auf eine Periode gesetzt. Jeder Nachfolger kann also zeitgleich mit dem Ende seines Vorgängers starten und die Stoffeinheit von ihm übernehmen. Die Stoffstromvorgänge  $1^o$ ,  $2^o$  und  $3^o$  nutzen je eine Einheit einer erneuerbaren Ressource  $R1$ ,



deren Kapazität auf  $R_{R1}^{\alpha} = 1$  beschränkt ist. Damit ist sicherstellt, dass sie nicht parallel ausgeführt werden.

Abbildung 4.16 zeigt den entsprechenden Vorgangsknotennetzplan. Rückbauvorgänge 0 und 3 sowie Stoffstromvorgang  $4^{\circ}$  sind fiktiv. Ebenfalls in Abbildung 4.16 ist das Gantt-Diagramm und das kumulative Ressourcenprofil von L1 für einen optimalen Schedule zu dieser Instanz gezeigt. Stoffstromvorgang  $1^{\circ}$  kann aufgrund der stoffstrominduzierten Zeitstruktur frühestens zum Zeitpunkt eins starten und muss abgeschlossen sein, bevor mit Stoffstromvorgang  $2^{\circ}$  die kumulative Ressource entleert werden kann. Somit kann Rückbauvorgang 2 auch erst zum Zeitpunkt eins starten, da es ansonsten zu einer Überlastung von L1 käme. Der optimale Zielfunktionswert beträgt zwei Perioden.

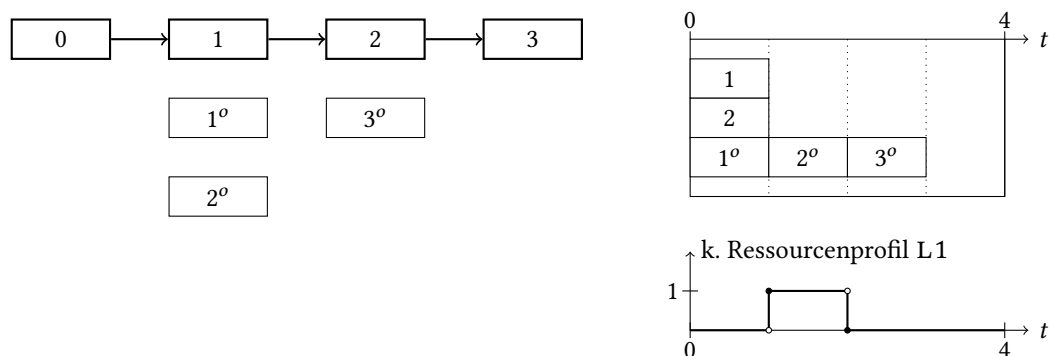


**Abbildung 4.16:** Vorgangsknotennetzplan, Gantt-Diagramm und kumulatives Ressourcenprofil zu Beispiel 4.10 mit stoffstrominduzierter Zeitstruktur.

Nun wird dieselbe Instanz ohne die stoffstrominduzierte Zeitstruktur betrachtet. Dazu zeigt Abbildung 4.17 den Vorgangsknotennetzplan sowie das Gantt-Diagramm und das kumulative Ressourcenprofil für einen optimalen Schedule. Die Stoffstromvorgänge  $1^{\circ}$ ,  $2^{\circ}$  und  $3^{\circ}$  sind nicht mehr an die Rückbauvorgänge angebunden und können zu beliebigen Zeitpunkten starten, sofern die übrigen Restriktionen eingehalten werden. Auf fiktive Start- und Endvorgänge für die Stoffstromvorgänge wird aus Darstellungsgründen im Vorgangsknotennetzplan verzichtet. Da die stoffstrominduzierte Zeitstruktur implizit für die Einhaltung eines Mindestbestands von null sorgt und nun nicht mehr vorhanden ist, muss dies mit zusätzlichen kumulativen Ressourcenrestriktionen  $f_{L1}^Y(S, t) + r_{L1}^Y(S^{\circ}, t) \geq 0$  ( $t \geq 0$ ) sichergestellt werden. Ansonsten könnte beispielsweise Stoffstromvorgang  $2^{\circ}$  zum Zeitpunkt null starten, wodurch das kumulative Ressourcenprofil auf  $-1$  fallen würde.

In dem im Gantt-Diagramm der Abbildung 4.17 gezeigten optimalen Schedule startet Stoffstromvorgang  $1^{\circ}$  bereits zum Zeitpunkt null, weshalb Stoffstromvorgang  $2^{\circ}$  zum Zeitpunkt eins starten kann. Dadurch ist zum Zeitpunkt eins genügend Kapazität vorhanden, sodass auch Rückbauvorgang 2 bereits zum Zeitpunkt null starten kann. Der optimale Zielfunktionswert beträgt folglich eine Periode.  $\square$

Der Vergleich der beiden Schedules in den Abbildungen 4.16 und 4.17 zeigt, dass die reale Problemstellung bei inhomogenen Stoffen (Wand 1 muss nachbehandelt werden und



**Abbildung 4.17:** Vorgangsknotennetzplan, Gantt-Diagramm und kumulatives Ressourcenprofil zu Beispiel 4.10 ohne stoffstrominduzierte Zeitstruktur.

Wand 2 nicht) nicht mehr realitätsgetreu modelliert wird, wenn die stoffstrominduzierte Zeitstruktur weggelassen wird. Der neutrale Stoffstromvorgang  $1^o$  startet in Abbildung 4.17 schon, obwohl seine Stoffe noch gar nicht freigesetzt wurden. Der Zielfunktionswert entspricht nicht der realen Projektdauer.

Beispiel 4.10 belegt folglich, dass es für die stoffstrombasierte Planung eines Rückbauprojekts ungeeignet wäre, die zeitliche Anordnung der Stoffstromvorgänge wie Schwindt und Trautmann (2000) lediglich über kumulative Ressourcenrestriktionen zu steuern.

### 4.3.2 Kritischer Pfad

Beim Vergleich mit der Schedulingliteratur in Abschnitt 3.5.2 wurde die organisatorische Einordnung der Stoffverarbeitung als gewichtigster Unterschied zwischen existierenden Ansätzen und dem (RMCPSP) herausgestellt. Während bei Schwindt und Trautmann (2000) oder anderen bekannten Arbeiten zu stoffstrombasierter Projekt- bzw. Produktionsplanung die Stoffverarbeitung das primäre Untersuchungsobjekt ist, hat sie in dieser Arbeit lediglich eine sekundäre Stellung. Im (RCPSP/M) macht sich diese Stellung am kritischen Pfad bemerkbar.

Nach Zimmermann u. a. (2006, S. 84) handelt sich sich beim kritischen Pfad um den längsten Pfad vom fiktiven Start- zum fiktiven Endknoten in einem Vorgangsknotennetzplan  $N$  bzw.  $N^{temp}$ . In den aus der Literatur bekannten Schedulingproblemen, wie dem (RCPSP), sind der fiktive Start- und Endknoten einfach zu identifizieren, da der Vorgangsknotennetzplan die eindeutige Quelle 0 und die eindeutige Senke  $I + 1$  hat. Im (RCPSP/M) ist der fiktive Startknoten ebenfalls einfach zu identifizieren, da der Vorgangsknotennetzplan  $N^{temp}$  die eindeutige Quelle 0 hat. Es gibt jedoch zwei fiktive Endknoten: Der fiktive End-Rückbauvorgang  $I + 1$  und der fiktive End-Stoffstromvorgang  $(J + 1)^o$  (vgl. z. B. Abbildung 4.3). Um zu klären, welcher dieser beiden Endknoten für die Bestimmung des kritischen Pfades ausschlaggebend ist, muss zunächst geklärt werden, zu welchem Zweck der kritische Pfad bestimmt wird. Die Länge des kritischen Pfades beschreibt den frühestmöglichen Endzeitpunkt des Projekts. Alle Vorgänge, die auf dem kritischen Pfad liegen, sind kriti-

sche Vorgänge, deren Verzögerung zu einer Verzögerung des Projekts führt. Besonders relevant ist die Kenntnis des kritischen Pfades und der kritischen Vorgänge, wenn als Zielsetzung die Minimierung der Projektdauer verfolgt wird. Dann beschreibt die Länge des kritischen Pfades eine untere Schranke für den Zielfunktionswert. Jede Verzögerung eines kritischen Vorgangs verringert den Grad der Zielerreichung. Diesen Zweck soll der kritische Pfad auch im (RCPSP/M) erfüllen. Und da der Zielfunktionswert im (RCPSP/M) dem Startzeitpunkt des fiktiven End-Rückbauvorgangs  $I + 1$  entspricht, wird sein Knoten  $I + 1$  in  $N^{temp}$  als ausschlaggebend für die Bestimmung des kritischen Pfades betrachtet. Zusammengefasst wird der kritische Pfad im (RCPSP/M) als der längste Pfad von Knoten 0 zu Knoten  $I + 1$  in  $N^{temp}$  definiert.

Bei Betrachtung eines Vorgangsknotennetzplans zu einer Instanz  $\bar{I}_{RCPSP/M}$ , wie beispielsweise in Abbildung 4.3, ist zu erkennen, dass der kritische Pfad nur Rückbauvorgänge umfassen kann. Dies folgt daraus, dass mit den Freisetzungsbeziehungen nur Zeitstrukturen von den Rückbauvorgängen in Richtung der Stoffstromvorgänge modelliert werden können, jedoch nicht umgekehrt. Eine solche Anordnung wurde damit begründet, dass der Stoffstrombereich dem Rückbaubereich natürlicherweise nachgeordnet ist. Folgerichtig können Stoffstromvorgänge auch nicht kritisch sein, d. h. bei einer Verzögerung direkt eine Erhöhung des Zielfunktionswertes verursachen. Allerdings können die Stoffstromvorgänge indirekt zu einer Erhöhung des Zielfunktionswertes beitragen, falls der Stoffstrombereich überlastet ist. Dies wird im (RCPSP/M) durch die erneuerbaren und kumulativen Ressourcenrestriktionen bewirkt. Die Charakterisierung der Stoffstromvorgänge als *nicht kritisch* bedeutet lediglich, dass ihre Verzögerung bei Relaxierung aller Ressourcenrestriktionen zu keiner Erhöhung des Zielfunktionswertes führen würde.

In einer optimalen Lösung des (RCPSP/M) ist es aufgrund des kritischen Pfades unerheblich, wie lange die Stoffverarbeitung noch über das Ende eines Rückbauprojekts hinaus dauert. In erster Linie müssen die Rückbauvorgänge so schnell wie möglich abgeschlossen werden. Nur wenn ein Rückkopplungseffekt auftritt, unterliegen auch Stoffstromvorgänge dem Optimierungskalkül, da sie freie Lagerfläche für weitere Rückbauvorgänge schaffen. Diese Besonderheit hebt das (RCPSP/M) sowohl strukturell als auch methodisch von existierenden Ansätzen ab. In Kapitel 5 werden problemspezifische Lösungsverfahren vorgestellt, die sich diese Eigenschaft zunutze machen.

Inwiefern die Auskopplung der Stoffstromvorgänge aus der Zielsetzung des Schedulingproblems der realen Problemstellung gerecht wird, hängt von der Stellung des Stoffstrombereichs innerhalb der Organisation ab. Wie bereits erläutert, erfüllt der Rückbau in der Regel die klassischen Merkmale eines Projekts, wohingegen die Stoffverarbeitung überwiegend produktionswirtschaftliche Merkmale aufweist. Deshalb wurde für diese Arbeit der hierarchische Modellierungsansatz gewählt, in dem primär das Rückbauprojekt und sekundär die Stoffverarbeitung geplant werden soll. Falls die Stoffverarbeitung sogar organisatorisch vom Rückbauprojekt entkoppelt ist, wie im kerntechnischen Rückbau mit Reststoffbearbeitungszentren (vgl. Abschnitt 3.2.3) möglich, erscheint der hierarchische Ansatz in besonderer Weise sinnvoll. Dennoch sind die in dieser Arbeit vorgestellten Modelle und Methoden auch auf die in der Literatur üblichen Probleme übertragbar, in wel-

chen alle Vorgänge (Rückbau- und Stoffstromvorgänge) zur Projektdauer gezählt werden. In Kapitel 6 wird dies anhand von Rechenexperimenten gezeigt.

## 4.4 Modellierungsaspekte

Dieser Abschnitt kann als Leitfaden bei der Modellierung einer realen Problemstellung als Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  verstanden werden. Er geht auf verschiedene Aspekte ein, die dabei beachtet werden müssen – unabhängig davon, ob eines der vorgeschlagenen Transformationsschemas verwendet, das Transformationsschema anderweitig ausgestaltet oder die reale Problemstellung direkt als Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  modelliert wird.

### 4.4.1 Ein- und Auslagerungen in bzw. aus demselben Lager

In praktischen Problemstellungen ist es für Verarbeitungsschritte durchaus denkbar, dass sie die zu verarbeitenden Stoffe aus demselben Lager beziehen, in das sie sie nach der Verarbeitung ganz oder teilweise wieder einlagern. Im (RMCPSP) ist das immer dann der Fall, wenn in einem Stoffstrompfad  $w \in \mathcal{W}$  das vor- und nachgeschaltete Lager eines Verarbeitungsschritts identisch sind. Auf die Stoffstromvorgänge übertragen bedeutet dies, dass Stoffeinheiten aus derselben kumulativen Ressource ausgelagert werden, in die sie ganz oder teilweise wieder eingelagert werden. In derartigen Situationen ist bei Verwendung gleichmäßiger kumulativer Ressourcennutzungen während der Vorgangsausführung ein besonderer Modellierungsaspekt zu beachten, der nachfolgend erläutert wird.

Zunächst wird der Fall betrachtet, in dem ein Stoffstromvorgang  $j \in V^o$  seine verarbeiteten Stoffeinheiten aus einer kumulativen Ressource  $k \in \mathcal{R}^Y$  auslagert und sie vollständig wieder in die identische kumulative Ressource  $k$  einlagert. Plausibel ist beispielsweise eine Auslagerung zum Start und eine Einlagerung zum Ende. Da in  $r_{jk}^Y(\cdot, \infty)$  über sämtliche Nutzungsarten aufsummiert wird, wird sich  $r_{jk}^Y(\cdot, \infty) = 0$  ergeben, d. h.,  $j$  ist bezüglich  $k$  neutral. Stoffstromvorgang  $j$  verringert das kumulative Ressourcenprofil nur während seiner Ausführung, nicht jedoch über sein Ende hinaus. Da die eingelagerte Menge mit dem Problemparameter  $r_{jk}^{Y,start}$  und die ausgelagerte Menge mit dem Problemparameter  $r_{jk}^{Y,end}$  angegeben wird, lässt sich aus der Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  reproduzieren, wie viele Stoffeinheiten  $j$  verarbeitet und auf welche kumulative Ressource er zugreift. Falls der Arbeitsfortschritt des Stoffstromvorgangs  $j$  konstant ist und die Stoffeinheiten synchron zu ihrer Verarbeitung ein- und ausgelagert werden, ist die Modellierung mittels gleichmäßiger Nutzung während die Vorgangsausführung sinnvoll. Hierfür wurde der Problemparameter  $r_{jk}^{Y,lin}$  definiert. Da es sich bei  $r_{jk}^{Y,lin}$  um einen Skalar handelt, kann nicht zwischen der realen Ein- und Auslagerung unterschieden werden. Stattdessen beträgt  $r_{jk}^{Y,lin} = 0$ , d. h. entspricht dem Differenzbetrag der realen Ein- und Auslagerung. Stoffstromvorgang  $j$  hat somit keine Auswirkungen auf das kumulative Ressourcenprofil. Es lässt sich aufgrund von  $r_{jk}^{Y,lin} = 0$

aus der Instanz  $\mathcal{I}_{\text{RCSP}/M}$  nicht mehr reproduzieren, wie viele Stoffeinheiten  $j$  verarbeitet und auf welche kumulative Ressource er zugreift. Diese Information lässt sich lediglich reproduzieren, wenn eine Flusserhaltung vorliegt und der gesamte Stoffstrom reproduziert wird, der durch  $j$  fließt. Beispiel 4.11 veranschaulicht die beschriebene Situation.

**Beispiel 4.11:** Eine Instanz  $\mathcal{I}_{\text{RCSP}/M}$  enthält einen Rückbauvorgang 1, der fünf Stoffeinheiten zu seinem Ende in eine kumulative Ressource L1 freisetzt, d. h.  $f_{1,L1}^{y,end} = 5$ . Darauf folgt ein Stoffstromvorgang  $1^o$ , der die fünf Stoffeinheiten aus L1 auslagert, verarbeitet und wieder in L1 einlagert. Auf  $1^o$  folgt ein weiterer Stoffstromvorgang  $2^o$ , der die fünf Stoffeinheiten zu seinem Start aus L1 auslagert, d. h.  $r_{2^o,L1}^{y,start} = -5$ . Sie erreichen danach die Senke.

Falls Stoffstromvorgang  $1^o$  die fünf Stoffeinheiten zu seinem Start auslagert und zu seinem Ende einlagert, gilt für seine kumulativen Ressourcennutzungen  $r_{1^o,L1}^{y,start} = -5$ ,  $r_{1^o,L1}^{y,lin} = 0$  und  $r_{1^o,L1}^{y,end} = 5$ . Es handelt sich bei  $1^o$  bezüglich L1 um einen neutralen Stoffstromvorgang, da  $r_{1^o,L1}^y(\cdot, \infty) = 0$ .

Falls Stoffstromvorgang  $1^o$  die fünf Stoffeinheiten gleichmäßig während seiner Ausführung ein- und auslagert, gilt für seine kumulativen Ressourcennutzungen  $r_{1^o,L1}^{y,start} = 0$ ,  $r_{1^o,L1}^{y,lin} = 0$  und  $r_{1^o,L1}^{y,end} = 0$ . Es handelt sich bei  $1^o$  bezüglich L1 um einen neutralen Stoffstromvorgang, da  $r_{1^o,L1}^y(\cdot, \infty) = 0$ . Die Anzahl der in  $1^o$  verarbeiteten Stoffeinheiten und das von  $1^o$  verwendete Lager lassen sich lediglich reproduzieren, indem der gesamte Stoffstrom ausgehend von Rückbauvorgang 1 und bis hin zu Stoffstromvorgang  $2^o$  reproduziert wird. Es sei angemerkt, dass Stoffstromvorgang  $1^o$  die kumulative Ressource L1 im Modell aufgrund von  $r_{1^o,L1}^{y,lin} = 0$  nicht nutzt. In der Praxis würde sich der Bestand von L1 während seiner Ausführung genaugenommen jedoch um diejenigen Stoffeinheiten reduzieren, die sich jeweils in Verarbeitung befinden. Hierbei handelt es sich um eine Modellierungsungenauigkeit, von der angenommen wird, dass sie bei großen Stoffmengen zu keiner Einschränkung der Realitätstreue führt. Andernfalls könnte die Reduktion mittels geeigneter kumulativer Ressourcennutzungen zum Start und Ende approximiert werden.  $\square$

Im Folgenden wird der Fall betrachtet, dass ein Stoffstromvorgang  $j \in V^o$  seine verarbeiteten Stoffeinheiten aus einer kumulativen Ressource  $k \in \mathcal{R}^y$  auslagert und teilweise wieder in die identische kumulative Ressource  $k$  einlagert. Ein solcher Stoffstromvorgang gilt hinsichtlich  $k$  nicht als neutral, da  $r_{jk}^y(\cdot, \infty) < 0$ . Bei einer Auslagerung zum Start und Einlagerung zum Ende lassen sich wie bereits oben alle Informationen aus der Instanz  $\mathcal{I}_{\text{RCSP}/M}$  reproduzieren. Bei gleichmäßiger Ein- und Auslagerung während der Vorgangsausführung entspricht  $r_{jk}^{y,lin}$  dem Differenzbetrag der realen Ein- und Auslagerung, welcher im betrachteten Fall echt negativ ist. Daher lässt sich aus der Instanz  $\mathcal{I}_{\text{RCSP}/M}$  zwar reproduzieren, dass  $j$  auf  $k$  zugreift. Die Anzahl der verarbeiteten Stoffeinheiten lässt sich jedoch lediglich reproduzieren, wenn eine Flusserhaltung vorliegt und der gesamte Stoffstrom reproduziert wird, der durch  $j$  fließt. Beispiel 4.12 veranschaulicht die beschriebene Situation.

**Beispiel 4.12:** Eine Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  enthält einen Rückbauvorgang 1, der fünf Stoffeinheiten zu seinem Ende in eine kumulative Ressource L1 freisetzt, d. h.  $f_{1,L1}^{y,end} = 5$ . Darauf folgt ein Stoffstromvorgang  $1^o$ , der die fünf Stoffeinheiten aus L1 auslagert, verarbeitet und davon drei Stoffeinheiten wieder in L1 einlagert. Die übrigen beiden Stoffeinheiten erreichen die Senke. Auf  $1^o$  folgt ein weiterer Stoffstromvorgang  $2^o$ , der die drei in L1 verbliebenen Stoffeinheiten zu seinem Start auslagert, d. h.  $r_{2^o,L1}^{y,start} = -3$ . Sie erreichen danach die Senke.

Falls Stoffstromvorgang  $1^o$  die fünf Stoffeinheiten zu seinem Start auslagert und die drei Stoffeinheiten zu seinem Ende einlagert, gilt für die kumulativen Ressourcennutzungen  $r_{1^o,L1}^{y,start} = -5$ ,  $r_{1^o,L1}^{y,lin} = 0$  und  $r_{1^o,L1}^{y,end} = 3$ .

Falls Stoffstromvorgang  $1^o$  die fünf bzw. drei Stoffeinheiten gleichmäßig während seiner Ausführung ein- und auslagert, gilt für die kumulativen Ressourcennutzungen  $r_{1^o,L1}^{y,start} = 0$ ,  $r_{1^o,L1}^{y,lin} = -2$  ( $= -5 + 3$ ) und  $r_{1^o,L1}^{y,end} = 0$ . Die Anzahl der in  $1^o$  verarbeiteten Stoffeinheiten lässt sich lediglich reproduzieren, indem der gesamte Stoffstrom ausgehend von Rückbauvorgang 1 und bis hin zu Stoffstromvorgang  $2^o$  reproduziert wird.  $\square$

Modelltheoretisch bringt die beschriebene Diskrepanz zwischen dem Wert des Problemparameters  $r_{jk}^{y,lin}$  und den realen Ein- und Auslagerungen keine Einschränkungen mit sich. Für ein Lösungsverfahren zum (RCPSP/M) ist ausschließlich die Information über die tatsächliche Veränderung (d. h. die Summe aus Ein- und Auslagerungen zu den jeweiligen Zeitpunkten) des kumulativen Ressourcenprofils relevant, welche mit dem skalaren  $r_{jk}^{y,lin}$  angegeben wird.

#### 4.4.2 No-Wait-Stoffverarbeitung

Bei der Vorstellung des Stoffstrommodells wurden die Stoffstrompfade ohne Beschränkung der Allgemeinheit als abwechselnde Folge von Verarbeitungsschritten und Lagern definiert. Falls Stoffe in der Realität ohne Unterbrechung von einem Verarbeitungsschritt zum nachfolgenden Verarbeitungsschritt übergeben werden sollen, kann dies mit einem Dummy-Lager der Kapazität null modelliert werden. In diesem Zusammenhang wird von einer *No-Wait-Stoffverarbeitung* gesprochen. Beispielsweise ist es im kerntechnischen Rückbau denkbar, dass Stoffe aufgrund vorliegender Radioaktivität unterbrechungsfrei weiterverarbeitet werden müssen (vgl. Hübner u. a. 2021, S. 273).

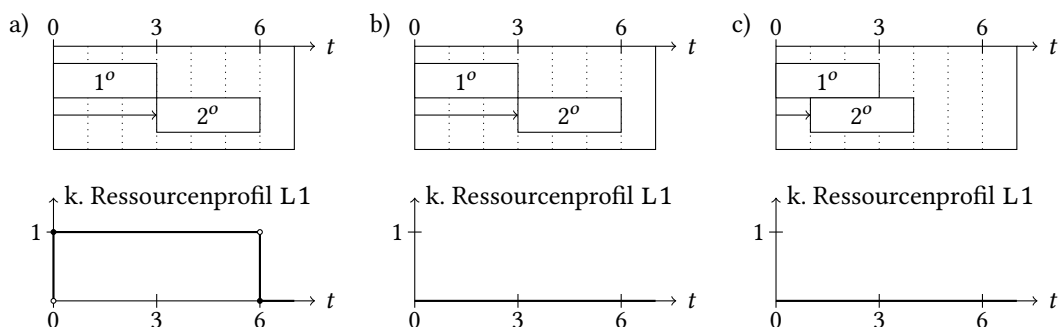
Formal wird die No-Wait-Stoffverarbeitung wie folgt modelliert: Wie bereits in Abschnitt 3.4.1 definiert, gilt für jeden Stoffstrompfad  $w \in W$ ,  $w := (k_1(w), p_1(w), k_2(w), p_2(w), \dots, s)$ . Eine No-Wait-Stoffverarbeitung liegt vor, wenn für jede Stoffeinheit ein Verarbeitungsschritt  $p_x(w)$  ohne Unterbrechung auf den vorangehenden Verarbeitungsschritt  $p_{x-1}(w)$  folgen muss. Dann ist das Lager  $k_x(w)$  als Dummy-Lager der Kapazität  $R_{k_x(w)}^y := 0$  zu modellieren. Auf das (RCPSP/M) übertragen bedeutet dies, dass die von  $p_{x-1}(w)$  abgeleiteten Stoffstromvorgänge Stoffe in eine kumulative Ressource der Kapazität null einlagern

und die von  $p_x(w)$  abgeleiteten Stoffstromvorgänge die Stoffe aus dieser kumulativen Ressource auslagern.

Um die Kapazität des Dummy-Lagers in einer Lösung des (RCPSP/M) nicht zu überschreiten, müssen Startzeitpunkte für Stoffstromvorgänge gefunden werden, sodass die von  $p_{x-1}(w)$  abgeleiteten Stoffstromvorgänge ihre Stoffe direkt an die von  $p_x(w)$  abgeleiteten Stoffstromvorgänge übergeben und das kumulative Ressourcenprofil konstant bei null bleibt. In anderen Worten, Ein- und Auslagerungen müssen jeweils gleichzeitig stattfinden. Diese Anforderung schränkt den Lösungsraum erheblich ein und birgt die Gefahr, dass das (RCPSP/M) unlösbar wird. Zur Sicherstellung der Lösbarkeit müssen die stoffstrominduzierte Zeitstruktur, die kumulativen Ressourcennutzungen und die Dauern der beteiligten Stoffstromvorgänge korrekt aufeinander abgestimmt sein. Beispiel 4.13 verdeutlicht, was darunter verstanden wird.

**Beispiel 4.13:** Eine Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  enthält zwei Stoffstromvorgänge  $1^o$  und  $2^o$ , die eine No-Wait-Verarbeitung von drei Stoffeinheiten abbilden. Es existiert eine stoffstrominduzierte Zeitbeziehung  $(1^o, 2^o) \in E^{\text{flow}}$ , d. h., Stoffstromvorgang  $2^o$  folgt auf Stoffstromvorgang  $1^o$ . Zwischen beide Stoffstromvorgänge wird eine kumulative Ressource L1 mit  $R_{L1}^Y := 0$  geschaltet, welche ein Dummy-Lager modelliert. Stoffstromvorgang  $1^o$  lagert drei Stoffeinheiten in L1 ein und Stoffstromvorgang  $2^o$  lagert drei Stoffeinheiten aus L1 aus. Es gilt  $d_{1^o} = d_{2^o} := 3$ .

Zunächst wird angenommen, dass die Einlagerung zum Vorgangsstart und die Auslagerung zum Vorgangsende erfolgt. Der zeitliche Mindestabstand wird auf  $d_{1^o, 2^o}^{\text{flow}} = 3$  gesetzt, d. h., Stoffstromvorgang  $2^o$  kann erst nach dem Ende von Stoffstromvorgang  $1^o$  starten. Diese Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  ist unzulässig, weil  $1^o$  und  $2^o$  nicht eingeplant werden können, sodass eine direkte Übergabe der Stoffe erfolgt. Das Zusammenspiel von zeitlichem Mindestabstand und Nutzungsarten erzwingt, dass in jeder zulässigen Lösung mindestens eine Stoffeinheit in L1 eingelagert werden muss. Da aber  $R_{L1}^Y := 0$ , ist das (RCPSP/M) unlösbar. Abbildung 4.18a veranschaulicht diese Situation.



**Abbildung 4.18:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 4.13: a) Einlagerung zum Start und Auslagerung zum Ende; b) Einlagerung zum Ende und Auslagerung zum Start; c) gleichmäßige Ein- und Auslagerung während der Ausführung.

Damit die Ein- und Auslagerung gleichzeitig stattfinden, könnte als zeitlicher Mindestabstand  $d_{1^o,2^o}^{flow} = -3$  gewählt werden. Dies wäre jedoch nicht realitätsgetreu, da der auslagernde Stoffstromvorgang dann vor dem einlagernden Stoffstromvorgang ausgeführt werden könnte, was der realen Problemstellung widerspricht. Zur korrekten Modellierung der No-Wait-Stoffverarbeitung müssen daher andere Nutzungsarten gewählt werden: Wenn die Einlagerung zum Vorgangsende und die Auslagerung zum Vorgangsstart erfolgt, kann mit einem zeitlichen Mindestabstand  $d_{1^o,2^o}^{flow} = 3$  eine direkte Übergabe der Stoffe erfolgen (vgl. Abbildung 4.18b).

Falls die Ein- und Auslagerung gleichmäßig während der Vorgangsausführung erfolgen, ist ebenfalls auf den korrekten zeitlichen Mindestabstand zu achten. Aufgrund der +1 in Gleichung (4.12) muss der zeitliche Mindestabstand genauso groß sein, wie die Dauer zur Verarbeitung einer Stoffeinheit, d. h. in diesem Beispiel  $d_{1^o,2^o}^{flow} = 1$ . Dann heben sich Ein- und Auslagerungen auf, sodass eine direkte Übergabe der Stoffe erfolgen kann (vgl. Abbildung 4.18c).  $\square$

Eine generelle Voraussetzung zur Modellierung der No-Wait-Stoffverarbeitung bei gleichmäßiger Nutzung während der Ausführung ist, dass sich die Stoffverarbeitungen synchronisieren lassen. Das bedeutet, dass die Dauer der beteiligten Stoffstromvorgänge gleich sein muss. Wenn die Dauern unterschiedlich sind, müssen zwangsläufig Stoffe eingelagert werden, da Stoffstromvorgänge per Definition nicht unterbrechbar sind. Zum besseren Verständnis wird auch auf Abbildung 4.10b verwiesen, wo sich die Stoffverarbeitungen nicht synchronisieren lassen und die Lagerauslastung folglich immer mindestens zwei Stoffeinheiten betragen wird. Im Gegensatz dazu ist in Abbildung 4.11b die Synchronisierung möglich.

Zusammenfassend lässt sich festhalten, dass die No-Wait-Stoffverarbeitung am besten modelliert werden kann, wenn die Einlagerung zum Vorgangsende und die Auslagerung zum Vorgangsstart erfolgt. Dann können die zeitlichen Mindestabstände jeweils auf die Dauer des vorangehenden Stoffstromvorgangs gesetzt werden und eine direkte Übergabe ist möglich. Die Einlagerung zum Vorgangsstart und die Auslagerung zum Vorgangsende sind für die No-Wait-Stoffverarbeitung als Nutzungsarten ungeeignet, da sie während der Vorgangsausführung Lagerplatz blockieren und eine Instanz mit Dummy-Lagern der Kapazität null unzulässig machen. Die Modellierung einer No-Wait-Stoffverarbeitung mit gleichmäßigen Ein- und Auslagerungen während der Vorgangsausführung ist zwar möglich, jedoch müssen sich die Stoffverarbeitungen synchronisieren lassen und der zeitliche Mindestabstand zwischen den beteiligten Stoffstromvorgängen muss genau der Verarbeitungsdauer einer Stoffeinheit entsprechen.

Abschließend soll noch darauf hingewiesen werden, dass die No-Wait-Stoffverarbeitung nicht dadurch modelliert werden kann, dass je zwei aufeinanderfolgende Stoffstromvorgänge zu einem einzelnen Stoffstromvorgang verschmolzen werden. Da Stoffstromvorgänge unterschiedliche erneuerbare Ressourcennutzungen haben können, würde die erneuerbare Ressourcennutzung des verschmolzenen Stoffstromvorgangs während seiner Ausführung variieren (vgl. Hübner u. a. 2021, S. 274). Das Konzept der erneuerbaren Ressourcen sieht



jedoch nur konstante erneuerbare Ressourcennutzungen eines Vorgangs vor, weshalb bei einer No-Wait-Stoffverarbeitung getrennte Stoffstromvorgänge benötigt werden.

#### 4.4.3 Mindestlagerdauern und Transportzeiten

In Rückbauprojekten lassen sich nur wenige praktische Beispiele finden, in denen Mindestlagerdauern plausibel erscheinen. Im kerntechnischen Rückbau wäre es vorstellbar, eine erforderliche Abklinglagerung bestimmter Stoffe als Mindestlagerdauer zu modellieren. Grundsätzlich ist jedoch eher davon auszugehen, dass Stoffe möglichst schnell einem Entsorgungsziel zugeführt werden sollen, welches sich außerhalb der Systemgrenzen des Schedulingmodells befindet. Ungeachtet der praktischen Relevanz lassen sich Mindestlagerdauern sehr einfach im (RCPSP/M) berücksichtigen, indem die ohnehin für die Freisetzungsbeziehungen und die stoffstrominduzierten Zeitbeziehungen definierten zeitlichen Mindestabstände  $d_{ij}^{rel}$  bzw.  $d_{jj'}^{flow}$  entsprechend erhöht werden.

Ähnlich kann verfahren werden, falls Transportzeiten berücksichtigt werden sollen. Ein Aufschlag der Transportzeiten auf die zeitlichen Mindestabstände ist sinnvoll, solange der Transport keine entscheidungsrelevanten Ressourcen benötigt. Andernfalls kann der Transport als zusätzlicher Verarbeitungsschritt im (RMCPSP) bzw. über zusätzliche Stoffstromvorgänge im (RCPSP/M) modelliert werden. Dadurch lassen sich Nutzungen erneuerbarer Ressourcen, wie beispielsweise eines Transportfahrzeugs, abbilden.

#### 4.4.4 Festlegung der Stoffeinheiten

In Modellabgrenzung (3) wurde erläutert, dass die Geometrie von Objekten in dieser Arbeit nicht berücksichtigt wird, sondern die Lagerbestände mittels abstrakter Stoffeinheiten beschrieben werden. Es wurde definiert, dass jede Stoffeinheit gleich viel Lagervolumen bzw. Lagerfläche beansprucht. Die Festlegung der Stoffeinheiten erfolgt im Rahmen der Modellierung einer Problemstellung und ist daher eine modellexogene Entscheidung. Wie Beispiel 4.14 verdeutlicht, ist bei der Modellierung folgender Grundsatz zu beachten: Stoffströme können mit verschiedenen Stoffeinheiten modelliert werden, solange sie sich nicht vermischen, d. h. durch mindestens ein gemeinsames Lager fließen. Ansonsten muss eine gemeinsame Stoffeinheit festgelegt werden, beispielsweise indem geeignete Umrechnungsfaktoren verwendet werden. Je nach gewünschtem Detaillierungsgrad der Stoffstromplanung kann die gemeinsame Stoffeinheit den tatsächlich benötigten Lagerplatz mehr oder weniger stark approximieren.

**Beispiel 4.14:** Eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  besteht aus den zwei realen Rückbauvorgängen 1 und 2. Rückbauvorgang 1 setzt gleichmäßig über seine Ausführung fünf Kubikmeter Betonschutt frei. Dieser Betonschutt wird in Lager L1 gepuffert, bevor er abtransportiert wird. Rückbauvorgang 2 setzt gleichmäßig über seine Ausführung zehn Container mit Stahlschrott frei. Dieser Stahlschrott wird in Lager L2 gebracht, wo er dekontaminiert und

anschließend freigemessen wird. Nach der Freimessung ist der Verbleib des Stahlschrotts nicht mehr entscheidungsrelevant.

Der Abtransport von Betonschutt wird als Verarbeitungsschritt P1 modelliert, die Dekontamination von Stahlschrott als Verarbeitungsschritt P2 und die Freimessung als Verarbeitungsschritt P3. Somit durchläuft der Betonschutt den Stoffstrompfad  $W1 = (L1, P1, s)$  und der Stahlschrott den Stoffstrompfad  $W2 = (L2, P2, L2, P3, s)$ . Dadurch, dass sich die Ströme von Betonschutt und Stahlschrott nicht vermischen, können sie mit verschiedenen Stoffeinheiten beschrieben werden. Das heißt, ein Kubikmeter kann als eine Stoffeinheit für den Betonschutt definiert werden und ein Container kann als eine Stoffeinheit für den Stahlschrott definiert werden. Die Stofffreisetzungen betragen dann  $f_{1,W1}^{y,lin} = 5$  und  $f_{2,W2}^{y,lin} = 10$ . Aus modelltheoretischer Sicht handelt es sich um zwei getrennte Stoffstrombereiche. Die Stoffeinheiten gleichen sich lediglich innerhalb ihres Stoffstrombereichs: Jeder Kubikmeter Betonschutt nimmt gleich viel Lagerkapazität in Anspruch und jeder Container Stahlschrott nimmt gleich viel Lagerkapazität in Anspruch. Die Lagerkapazität  $R_{L1}^Y$  wird daher in Kubikmetern beschrieben und die Lagerkapazität  $R_{L2}^Y$  in Containern.

Im Gegensatz zur gerade skizzierten Problemstellung wird nun davon ausgegangen, dass sowohl Betonschutt als auch Stahlschrott in einem gemeinsamen Lager L1 gepuffert werden (mit beliebiger räumlicher Verteilung innerhalb des Lagers). Der Betonschutt durchläuft demnach weiterhin den Stoffstrompfad W1. Der Stahlschrott durchläuft nun den Stoffstrompfad  $W2' = (L1, P2, L1, P3, s)$ . Die Ströme von Betonschutt und Stahlschrott vermischen sich jetzt in Lager L1, weshalb eine gemeinsame Stoffeinheit festgelegt werden muss. Diese Anforderung ist auch praktisch plausibel: Weil die Stoffe innerhalb des Lagers L1 beliebig verteilt sein können, ist eine Aussage zur Lagerauslastung nur mit einer gemeinsamen Einheit zur Beschreibung des Lagerbestands möglich. Gemäß Modellabgrenzung (3) soll sich diese Einheit vorrangig am Volumen bzw. der Fläche orientieren und nicht an der Geometrie, welche Gegenstand logistischer Planungsprobleme ist. Daher bietet es sich an, die Container mit Stahlschrott in Kubikmeter umzurechnen. Beispielsweise könnte ein Container Stahlschrott einen halben Kubikmeter Lagervolumen benötigen. Dann gilt  $f_{1,W1}^{y,lin} = 5$  und  $f_{2,W2'}^{y,lin} = 5$ .  $\square$

## 4.5 Modellerweiterungen

In diesem Abschnitt werden verschiedene Erweiterungen des (RCPSP/M) diskutiert, welche dazu dienen, die Anwendbarkeit des Modells auszudehnen. Die Darlegung der Erweiterungen erfolgt rein theoretisch, d. h. ohne Evidenz für die praktische Notwendigkeit in Rückbauprojekten. Es wird daher nicht ins Detail gegangen, sondern lediglich gezeigt, welche zusätzlichen Möglichkeiten das (RCPSP/M) bietet. Abschnitt 5.7 skizziert aufbauend auf diesem Abschnitt Anpassungen am Lösungsverfahren, die mit den nachfolgend vorgestellten Modellerweiterungen erforderlich werden.

### 4.5.1 Zeitliche Höchstabstände und Höchstlagerdauern

Die Freisetzungsbeziehungen und die stoffstrominduzierten Zeitbeziehungen wurden für das (RCPSP/M) so definiert, dass lediglich zeitliche Mindestabstände  $d_{ij}^{rel} \in \mathbb{Z}_{\geq 0}$  bzw.  $d_{jj'}^{flow} \in \mathbb{Z}_{\geq 0}$  möglich sind. Dadurch ist der Vorgangsknotenetzplan topologisch sortierbar, was wiederum zur Folge hat, dass sich Generierungsschemas einfacher anwenden lassen (vgl. Abschnitt 2.3.1).

Um auch zeitliche Höchstabstände einzubeziehen, ist eine Erweiterung des Modells auf das in Abschnitt 2.2.3 vorgestellte Konzept der Zeitbeziehungen erforderlich. Dazu werden die Vorrangrestriktionen im (RCPSP/M) durch Zeitrestriktionen (2.5) ersetzt. Außerdem werden für alle Freisetzungsbeziehungen  $(i, j) \in E^{rel}$  die zeitlichen Mindestabstände  $d_{ij}^{rel} \in \mathbb{Z}_{\geq 0}$  durch Zeitabstände  $\delta_{ij}^{rel} \in \mathbb{Z}$  ersetzt. Umgekehrte Freisetzungsbeziehungen werden ebenfalls benötigt, d. h.  $E^{rel} \subseteq V \times V^o \cup V^o \times V$ . Analog dazu werden auch für alle stoffstrominduzierten Zeitbeziehungen  $(j, j') \in E^{flow}$  die zeitlichen Mindestabstände  $d_{jj'}^{flow} \in \mathbb{Z}_{\geq 0}$  durch Zeitabstände  $\delta_{jj'}^{flow} \in \mathbb{Z}$  ersetzt. Ein zeitlicher Höchstabstand kann nun durch eine umgekehrte Anordnungsbeziehung und einen negativen Zeitabstand modelliert werden (vgl. Abschnitt 2.2.3).

In Abschnitt 4.4.3 wurde erläutert, dass Mindestlagerdauern über entsprechende zeitliche Mindestabstände zwischen Stoffstromvorgängen ausgedrückt werden können. Mit der Erweiterung um zeitliche Höchstabstände lassen sich analog dazu Höchstlagerdauern ausdrücken. Wenn beispielsweise die Stoffe, welche von einem Stoffstromvorgang  $1^o$  an einen Stoffstromvorgang  $2^o$  übergeben werden, maximal zehn Perioden zwischen dem Ende von  $1^o$  und dem Start von  $2^o$  gelagert werden dürfen, ist eine stoffstrominduzierte Zeitbeziehung  $(2^o, 1^o)$  mit Zeitabstand  $\delta_{2^o, 1^o}^{flow} := -(d_{1^o} + 10)$  einzuführen.

Mindest- und Höchstlagerdauern können mittels zeitlichen Mindest- und Höchstabständen ausgedrückt werden, falls inhomogene Stoffe vorliegen. Nur dann ist auch eine stoffstrominduzierte Zeitstruktur vorhanden, für welche die zeitlichen Mindest- und Höchstabstände definiert sind. Für homogene Stoffströme beschreiben Neumann, Schwindt und Trautmann (2005, S. 501f) eine Möglichkeit zur Modellierung von Mindest- und Höchstlagerdauern, die sie als „quarantine time“ bzw. „shelf life time“ bezeichnen. Dazu muss das Modell jedoch auch um die Möglichkeit erweitert werden, Mindestbestände für die kumulativen Ressourcen anzugeben (siehe Abschnitt 4.5.3).

### 4.5.2 Zeitlich variable Ressourcenkapazitäten

Gemäß Modelleinschränkung (5) und in Einklang mit den Konzepten in der Schedulingliteratur wurde sowohl für die erneuerbaren als auch für die kumulativen Ressourcen im (RCPSP/M) davon ausgegangen, dass ihre Kapazitäten  $R_k^\alpha$  bzw.  $R_k^\gamma$  über den Zeithorizont konstant sind. In der Praxis ist es jedoch möglich, dass diese Kapazitäten zeitlich variieren. Beispielsweise kann die Zahl der verfügbaren Arbeitskräfte im Laufe eines Rückbauprojekts aufgrund geplanter Abgänge absinken. Oder es kann nach Erreichen eines

bestimmten Projektfortschritts eine neue Lagerfläche geschaffen werden, weil zusätzlicher Platz zur Verfügung steht.

Zunächst werden *modellexogene zeitliche Variabilitäten* der Ressourcenkapazitäten betrachtet. Das bedeutet, der Verlauf der Kapazität über den Zeithorizont ist ein Problemparameter und keine entscheidungsabhängige Variable. Projektverantwortliche müssen also modellexogen bestimmen, zu welchem Zeitpunkt beispielsweise Arbeitskräfte aus der Organisation ausscheiden, oder zu welchem Zeitpunkt eine neue Lagerfläche geschaffen wird.

Eine Möglichkeit zur Modellierung modellexogener zeitlicher Variabilitäten geht auf Bartusch u. a. (1988) zurück. Ressourcenkapazitäten werden durch Einführung von Dummy-Vorgängen gezielt blockiert, sodass der zeitliche Kapazitätsverlauf nachgebildet wird. Dazu müssen die Dummy-Vorgänge zeitlich fixiert werden, was durch Einführung eines gleich großen zeitlichen Mindest- und Höchstabstands zum fiktiven Startvorgang möglich ist. Es sei angemerkt, dass das (RCPSP/M) hierzu wie im vorangegangenen Abschnitt 4.5.1 beschrieben um das Konzept der zeitlichen Höchstabstände erweitert werden muss. Beispiel 4.15 verdeutlicht die Modellierung modellexogener zeitlicher Variabilitäten für kumulative Ressourcen.

**Beispiel 4.15:** Eine Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  besteht aus zwei Stoffstromvorgängen  $1^o$  und  $2^o$ , die die Verarbeitung von drei Stoffeinheiten abbilden. Es existiert eine stoffstrominduzierte Zeitbeziehung  $(1^o, 2^o) \in E^{\text{flow}}$ , d. h., Stoffstromvorgang  $2^o$  folgt auf Stoffstromvorgang  $1^o$ . Zwischen beide Stoffstromvorgänge wird die kumulative Ressource L1 geschaltet, die erst nach Ende der zweiten Periode mit einer Kapazität von drei Stoffeinheiten zur Verfügung steht. Es gilt  $d_{1^o} = d_{2^o} := 1$  und der zeitliche Mindestabstand beträgt  $d_{1^o, 2^o}^{\text{flow}} = 1$  ( $= \delta_{1^o, 2^o}^{\text{flow}}$ ). Die Einlagerung erfolgt zum Vorgangsstart und die Auslagerung zum Vorgangsende, d. h.  $r_{1^o, \text{L1}}^{y, \text{start}} = 3$  und  $r_{2^o, \text{L1}}^{y, \text{end}} = -3$ .

Da der Kapazitätsverlauf von L1 bekannt ist, liegt eine modellexogene zeitliche Variabilität vor. Um diese zu modellieren, wird  $R_{\text{L1}}^y := 3$ , d. h. auf die über den Zeithorizont maximal verfügbare Lagerkapazität gesetzt. Diese Lagerkapazität muss während der ersten beiden Perioden blockiert werden. Dazu wird ein Dummy-Stoffstromvorgang  $3^o$  mit  $d_{3^o} := 2$  eingeführt. Er lagert zu seinem Start drei Einheiten in L1 ein und zu seinem Ende drei Einheiten aus L1 aus, d. h.  $r_{3^o, \text{L1}}^{y, \text{start}} := 3$  und  $r_{3^o, \text{L1}}^{y, \text{end}} := -3$ . Außerdem wird er auf den Zeitpunkt null fixiert. Hierzu dienen die Anordnungsbeziehungen  $(0, 3^o), (3^o, 0) \in E^{\text{rel}}$  mit den Zeitabständen  $\delta_{0, 3^o}^{\text{rel}} = 0$  und  $\delta_{3^o, 0}^{\text{rel}} = 0$ .

Das Zusammenspiel des zeitlichen Mindestabstands  $d_{1^o, 2^o}^{\text{flow}} = 1$  und der Nutzungsarten der kumulativen Ressourcen erzwingt, dass in jeder zulässigen Lösung zur beschriebenen Instanz drei Stoffeinheiten in Lager L1 eingelagert werden müssen. Aufgrund der blockierten Lagerkapazität kann Stoffstromvorgang  $1^o$  daher frühestens zum Zeitpunkt drei und Stoffstromvorgang  $2^o$  frühestens zum Zeitpunkt vier starten.  $\square$

Dummy-Vorgänge können auch fixiert werden, indem das (RCPSP/M) um zusätzliche Restriktionen erweitert wird, die ihre Startzeitpunkte auf einen bestimmten Wert festlegen. Für die Stoffstromvorgänge in Beispiel 4.13 müssten die beiden Gleichungen  $S_{3^o} = 0$  und  $S_{4^o} = 2$  zu den Restriktionen hinzugefügt werden.

Intuitiver als eine Modellierung mittels Dummy-Vorgängen ist eine Erweiterung des (RCPSP/M), in der die Ressourcenkapazitäten um eine zeitliche Dimension erweitert werden. Formal werden dazu  $R_k^\alpha$  bzw.  $R_k^\gamma$  durch  $R_{kt}^\alpha$  und  $R_{kt}^\gamma$  für alle  $k \in \mathcal{R}^\alpha$  bzw.  $k \in \mathcal{R}^\gamma$  und alle  $t \geq 0$  ersetzt. Damit können Projektverantwortliche direkt für jeden Zeitpunkt angeben, welche Ressourcenkapazität zur Verfügung steht.

Bei *modellendogenen zeitlichen Variabilitäten* der Ressourcenkapazitäten sind  $R_k^\alpha$  und  $R_k^\gamma$  entscheidungsabhängig. Formal müsste die Notation  $R_k^\alpha(S, S^o)$  bzw.  $R_k^\gamma(S, S^o)$  verwendet werden. Dies würde jedoch weitere Restriktionen im (RCPSP/M) nach sich ziehen, die diese Abhängigkeiten beschreiben. Sinnvoller ist es, die Ressourcenkapazitäten wie bisher konstant zu halten und die Variabilitäten über Vorgänge abzubilden. Im Vergleich zu Beispiel 4.15 sind diese Vorgänge jedoch nicht fixiert, wie Beispiel 4.16 zeigt.

**Beispiel 4.16:** Wie in Beispiel 4.15 besteht eine Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  aus zwei Stoffstromvorgängen  $1^o$  und  $2^o$ , die die Verarbeitung von drei Stoffeinheiten abbilden. Es existiert eine stoffstrominduzierte Zeitbeziehung  $(1^o, 2^o) \in \text{Flow}$ , d. h. Stoffstromvorgang  $2^o$  folgt auf Stoffstromvorgang  $1^o$ . Zwischen beide Stoffstromvorgänge wird die kumulative Ressource L1 geschaltet. Es gilt  $d_{1^o} = d_{2^o} := 1$  und der zeitliche Mindestabstand beträgt  $d_{1^o, 2^o}^{\text{flow}} = 1$ . Die Einlagerung erfolgt zum Vorgangsstart und die Auslagerung zum Vorgangsende, d. h.  $r_{1^o, L1}^{\gamma, \text{start}} = 3$  und  $r_{2^o, L1}^{\gamma, \text{end}} = -3$ . L1 steht erst mit einer Kapazität von drei Stoffeinheiten zur Verfügung, wenn die dafür benötigte Fläche geräumt wurde. Diese Räumung dauert vier Perioden.

Da der Zeitpunkt der Räumung nicht bekannt ist, liegt eine modellendogene zeitliche Variabilität vor. Um diese zu modellieren, wird  $R_{L1}^\gamma := 3$ , d. h. auf die über den Zeithorizont maximal verfügbare Lagerkapazität gesetzt. Da L1 zum Projektstart nicht zur Verfügung steht, wird ein Dummy-Stoffstromvorgang  $3^o$  mit  $d_{3^o} = 0$  und  $r_{3^o, L1}^{\gamma, \text{start}} = 3$  eingeführt und auf den Zeitpunkt null fixiert. Das Räumen der für L1 benötigten Fläche wird mit einem zusätzlichen Stoffstromvorgang  $4^o$  der Dauer  $d_{4^o} = 4$  modelliert. Dieser Stoffstromvorgang ist kein Dummy und kann somit auch erneuerbare Ressourcen, wie beispielsweise Arbeitskräfte, die zum Räumen benötigt werden, nutzen. Nach Abschluss von  $4^o$  steht L1 zur Verfügung, weshalb  $r_{4^o, L1}^{\gamma, \text{end}} := -3$  gesetzt wird. Der Startzeitpunkt  $S_{4^o}$  ist entscheidungsabhängig, womit die tatsächlich verfügbare Kapazität von L1 modellendogen bestimmt wird.

Falls  $4^o$  direkt zum Zeitpunkt null startet, kann ab Zeitpunkt vier in L1 eingelagert werden. Stoffstromvorgang  $1^o$  kann daher frühestens zum Zeitpunkt vier und Stoffstromvorgang  $2^o$  frühestens zum Zeitpunkt fünf starten.  $\square$

Dadurch, dass für jeden Stoffstromvorgang individuell festgelegt werden kann, zu welchem Zeitpunkt während seiner Ausführung er wie viele Einheiten einer kumulativen Ressource nutzt, lassen sich beliebige modellendogene zeitliche Variabilitäten modellieren. Die kumulativen Ressourcennutzungen sind hierbei nicht als Ein- und Auslagerungen zu verstehen, sondern gemäß der ursprünglichen Definition in Abschnitt 2.2.6 als Ressourcenbeanspruchungen und -freigaben. Eine Beanspruchung reduziert den verfügbaren Lagerplatz und eine Freigabe erhöht den verfügbaren Lagerplatz. Für erneuerbare Ressourcen sind die Zeitpunkte der Beanspruchung und Freigabe per Definition festgelegt. Kein Vorgang kann nach seinem Ende noch eine Auswirkung auf eine erneuerbare Ressource haben. Aufgrund dieser modelltheoretischen Natur der erneuerbaren Ressourcen müssen sie bei Vorhandensein modellendogener zeitlicher Variabilitäten ihrer Kapazität mit kumulativen Ressourcen nachgebildet werden (vgl. Abschnitt 2.2.6).

### 4.5.3 Homogene Stoffströme und Mindestbestände

Falls homogene Stoffströme vorliegen, kann die stoffstrominduzierte Zeitstruktur weggelassen werden (vgl. Abschnitt 4.3.1). Dadurch wird der Lösungsraum des Schedulingproblems nicht unnötig eingeschränkt, sondern die volle Flexibilität bleibt erhalten, wie Schwindt und Trautmann (2000, S. 506) argumentieren. Die stoffstrominduzierte Zeitstruktur wird jedoch im (RCPSM/M) benötigt, um für kumulative Ressourcen sicherzustellen, dass ein Mindestbestand von null eingehalten wird. Fällt sie weg, müssen negative Bestände durch zusätzliche kumulative Ressourcenrestriktionen verhindert werden.

Eine Möglichkeit hierzu ist die Erweiterung des (RCPSM/M) um das bereits in Abschnitt 2.2.6 vorgestellte Konzept der Mindestbestände für kumulative Ressourcen. Mindestbestände sind Problemparameter und werden als  $\underline{R}_k^Y$  für alle  $k \in \mathcal{R}^Y$  bezeichnet. Ihre Einhaltung wird im (RCPSM/M) durch Erweiterung um die Restriktionen

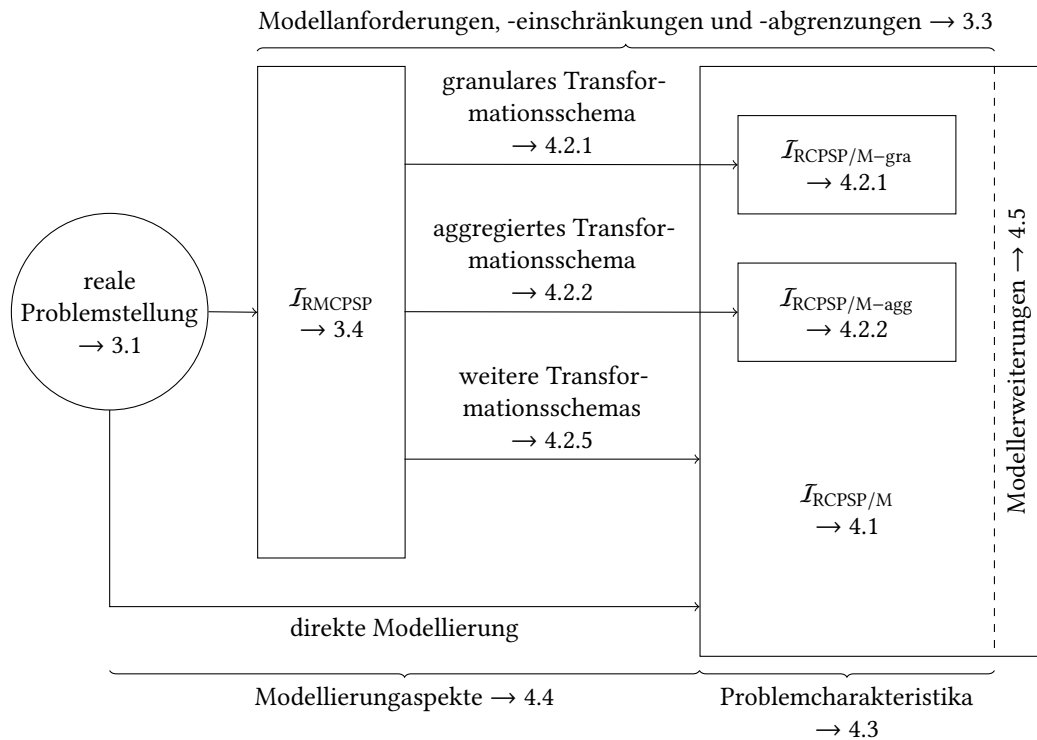
$$f_k^Y(S, t) + r_k^Y(S^o, t) \geq \underline{R}_k^Y \quad (\forall k \in \mathcal{R}^Y, t \geq 0) \quad (4.20)$$

sichergestellt. Ein Mindestbestand von  $\underline{R}_k^Y = 0$  drückt aus, dass keine negativen Bestände zugelassen sind.

Nach einer Idee von Carlier und Moukrim (2015, S. 185) ist die explizite Ergänzung der Restriktionen (4.20) nicht notwendig, um Mindestbestände zu berücksichtigen. Wenn die Restriktionen (4.20) mit  $-1$  multipliziert werden, sind sie schließlich identisch zu den bereits im (RCPSM/M) vorhandenen Restriktionen der Höchstbestände (4.15). Daher kann folgende Transformation der Problemparameter durchgeführt werden: Jede kumulative Ressource  $k$  mit Mindestbestand  $\underline{R}_k^Y$  und Höchstbestand  $R_k^Y$  (= Lagerkapazität) wird in zwei kumulative Ressourcen  $k_1$  und  $k_2$  mit  $R_{k_1}^Y := R_k^Y$  und  $R_{k_2}^Y := -\underline{R}_k^Y$  aufgespalten. Die Nutzungen dieser kumulativen Ressourcen durch Stoffstromvorgänge werden auf  $(r_{jk_1}^{Y,start}, r_{jk_1}^{Y,lin}, r_{jk_1}^{Y,end}) := (r_{jk}^{Y,start}, r_{jk}^{Y,lin}, r_{jk}^{Y,end})$  und  $(r_{jk_2}^{Y,start}, r_{jk_2}^{Y,lin}, r_{jk_2}^{Y,end}) := (-r_{jk}^{Y,start}, -r_{jk}^{Y,lin}, -r_{jk}^{Y,end})$  gesetzt. Damit ist es ausreichend, wie im ursprünglichen (RCPSM/M) lediglich Restriktionen der Höchstbestände zu berücksichtigen.

## 4.6 Überblick und Zusammenfassung

Abbildung 4.19 zeigt einen Überblick über die Inhalte des vorangegangenen Kapitels 3 und dieses Kapitels. Pfeile stellen dabei Modellierungsprozesse und Rechtecke stellen Probleminstanzen dar. Der Rechtspfeil  $\rightarrow$  symbolisiert einen Verweis auf den jeweiligen Abschnitt, in dem entsprechende Inhalte behandelt werden.



**Abbildung 4.19:** Überblick über die Inhalte der Kapitel 3 und 4.

Zusammengefasst wurde in diesem Kapitel mit dem (RCPSP/M) eine Konkretisierung der im (RMCPSP) beschriebenen Problemstellung vorgestellt, welche die Stoffstromplanung mittels Stoffstromvorgängen in die Rückbauplanung integriert. Eine reale Problemstellung kann entweder direkt als (RCPSP/M) oder auch als (RMCPSP) mit nachfolgender Anwendung eines Transformationsschemas modelliert werden. Es wurde ein granulares und ein aggregiertes Transformationsschema vorgestellt, die sich in der Art der erzeugten Stoffstromvorgänge unterscheiden. Darüber hinaus gibt es weitere Ausgestaltungsmöglichkeiten für Transformationsschemas. Die Modellierung einer realen Problemstellung mittels (RMCPSP) und nachfolgender Anwendung eines Transformationsschemas bietet den Vorteil, dass sich Projektverantwortliche mit weniger Modellierungsdetails beschäftigen müssen, sondern die Zusammenstellung der Problemparameter zu einem gewissen Grad automatisiert abläuft. Dies erscheint insbesondere bei großen Projekten in einem frühen Planungsstadium sinnvoll. Des Weiteren können die Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  und  $\mathcal{I}_{\text{RCPSP/M-agg}}$  aufgrund ihrer speziellen Struktur auch effizienter gelöst werden, wie nachfolgendes Kapitel zeigen wird. Das (RMCPSP) als konzeptionelle Problemformulierung mit

hohem Abstraktionsgrad dient außerdem dem besseren Verständnis des Zusammenhangs zwischen primär betrachtetem Rückbaubereich und sekundär betrachtetem Stoffstrombereich. Zur Anwendung eines Lösungsverfahrens eignet sich jedoch ausschließlich das konkretisierte (RCPSP/M), bei welchem nachfolgendes Kapitel ansetzt.

Das (RCPSP/M) und das granulare Transformationsschema werden auch in Gehring, Volk u. a. (2022) ausführlich vorgestellt. Für die darin beschriebenen Rechenexperimente wurde auch das aggregierte Transformationsschema eingesetzt. Details zum aggregierten Transformationsschema sowie zu den Problemcharakteristika, den Modellierungsaspekten und den Modellerweiterungen sind in dem Beitrag jedoch nicht enthalten.



## 5 Lösungsverfahren

Dieses Kapitel befasst sich mit Lösungsverfahren für das im vorangegangenen Kapitel 4 eingeführte (RCPSP/M). Zunächst wird dazu in Abschnitt 5.1 erläutert, welche Herausforderungen bei der Lösung von Schedulingproblemen mit kumulativen Ressourcen bestehen. Abschnitt 5.2 befasst sich mit der Analyse existierender Ansätze in der Literatur. In den darauffolgenden Abschnitten 5.3, 5.4 und 5.5 werden konkrete Lösungsverfahren für granulare, aggregierte und beliebige Instanzen des (RCPSP/M) vorgestellt. Kern der jeweiligen Lösungsverfahren sind neu entwickelte Generierungsschemas, die mit den kumulativen Ressourcenrestriktionen umgehen können. In Abschnitt 5.6 werden die Lösungsverfahren miteinander verglichen. Abschnitt 5.7 skizziert Erweiterungen der Lösungsverfahren. In den Abschnitten 5.8 und 5.9 werden die Constraintprogrammierung und die lineare Programmierung als Alternativen zu den neu entwickelten Lösungsverfahren behandelt. Das Kapitel schließt mit einem Überblick in Abschnitt 5.10.

Die in den Abschnitten 5.3, 5.4 und 5.5 entwickelten Lösungsverfahren sind strenggenommen keine vollwertigen Lösungsverfahren in dem Sinne, dass sie den Lösungsraum des Problems absuchen. Jedes Generierungsschema berechnet basierend auf seinen Eingabedaten lediglich einen Schedule oder schlägt fehl. Ein vollwertiges Lösungsverfahren ergibt sich durch Einbettung eines Generierungsschemas in eine Metaheuristik, die die Strategie zum Absuchen des Lösungsraums vorgibt. Während des Durchlaufs der Metaheuristik wird das Generierungsschema mehrfach mit veränderten Eingabedaten aufgerufen. Da die Metaheuristik jedoch von der konkreten Problemstellung unabhängig ist, wird in diesem Kapitel noch nicht darauf eingegangen. Zwei exemplarische Metaheuristiken, in die die nachfolgend behandelten Generierungsschemas eingebettet werden können, werden in Kapitel 6 behandelt.

### 5.1 Grundlagen zur Lösbarkeit von Schedulingproblemen mit kumulativen Ressourcen

Wie bereits in Abschnitt 2.2.6 dargelegt, führt das Vorhandensein kumulativer Ressourcen in einem Schedulingproblem dazu, dass bereits das Entscheidungsproblem zur Existenz einer zulässigen Lösung NP-vollständig ist (Neumann, Schwindt und Zimmermann 2003, S. 130). Vereinfacht ausgedrückt ist die Erzeugung einer zulässigen Lösung im Allgemeinen „komplex“, sobald kumulative Ressourcen vorhanden sind und „einfach“, sofern nur erneuerbare Ressourcen vorhanden sind. Nachfolgend wird qualitativ dargelegt, woraus dieser strukturelle Unterschied resultiert.

Für das (RCPSP), welches nur erneuerbare Ressourcen und Vorrangbeziehungen enthält, kann mit den in Abschnitt 2.3.1 vorgestellten Generierungsschemas in Polynomialzeit ein zulässiger Schedule bestimmt werden. Der Grund dafür liegt in der modelltheoretischen Natur der erneuerbaren Ressourcen: Für jeden Vorgang ist per Definition festgelegt, dass er zu seinem Start eine bestimmte Anzahl an erneuerbaren Ressourceneinheiten nutzt und genau diese Anzahl zu seinem Ende wieder freigibt. Es existieren keine vorgangsübergreifenden Ressourcennutzungen, d. h., kein Vorgang gibt erneuerbare Ressourceneinheiten frei, die ein anderer Vorgang zuvor beansprucht hat. Somit können in jeder Iteration eines Generierungsschemas die Nutzungen eines Vorgangs von den Kapazitäten der erneuerbaren Ressourcen subtrahiert werden – mit der Gewissheit, dass sich dieser Zustand in weiteren Iterationen nicht verändern wird. Folglich lässt sich das Generierungsschema so ausgestalten, dass in jeder Iteration ein zulässiger partieller Schedule vorliegt. Und dadurch, dass die Anzahl der Iterationen polynomiell mit der Problemgröße wächst, wird immer ein zulässiger Schedule in Polynomialzeit gefunden (oder die Unzulässigkeit der Instanz nachgewiesen).

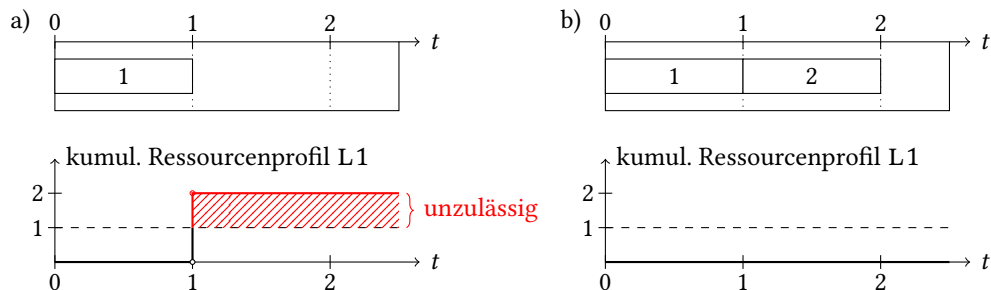
Das (RCPSP/max) resultiert aus der Erweiterung der Vorrangbeziehungen auf das generalisierte Konzept der Zeitbeziehungen. Hinsichtlich der erneuerbaren Ressourcen gilt weiterhin, dass in jeder Iteration eines Generierungsschemas ein zulässiger partieller Schedule vorliegt. Hinsichtlich der Zeitbeziehungen gilt dies dann allerdings nicht mehr. Bei dem in der Literatur üblichen seriellen Generierungsschema zu diesem Problem kann ein zeitlicher Höchstabstand verletzt werden (vgl. Abschnitt 2.3.1). Mittels Ausplanungsschritten wird in seinem solchen Fall versucht, die Zulässigkeit wiederherzustellen. Ob eine Wiederherstellung erfolgreich ist, kann aufgrund möglicher Verletzungen der erneuerbaren Ressourcenrestriktionen nicht garantiert werden. Deshalb kann das serielle Generierungsschema fehlschlagen. Die Erzeugung einer zulässigen Lösung ist nicht mehr generell in Polynomialzeit möglich. Es sei angemerkt, dass ein Generierungsschema für das (RCPSP/max) auch umgekehrt ausgestaltet werden könnte: Es lässt sich problemlos ein Schedule zusammenstellen, der hinsichtlich der Zeitbeziehungen zulässig ist. Dann können erneuerbare Ressourcenkapazitäten überschritten sein, wodurch Ausplanungsschritte erforderlich werden. Zusammengefasst begründet das Zusammenspiel aus Zeit- und erneuerbaren Ressourcenrestriktionen die NP-Vollständigkeit des Entscheidungsproblems zur Existenz einer zulässigen Lösung des (RCPSP/max).

Die kumulativen Ressourcen im (RCPSP/c) erschweren die Erzeugung einer zulässigen Lösung auf ähnliche Weise. Bei kumulativen Ressourcen sind vorgangsübergreifende Ressourcennutzungen möglich, d. h., ein Vorgang kann kumulative Ressourceneinheiten freigeben, die ein anderer Vorgang zuvor beansprucht hat. In einem Generierungsschema mit iterativer Einplanung der Vorgänge führt dies zwangsläufig dazu, dass auch unzulässige partielle Schedules vorliegen können. Die Unzulässigkeiten lassen sich durch die Einplanung weiterer Vorgänge, ggf. in Kombination mit Ausplanungsschritten, auflösen. Dies kann jedoch nicht garantiert werden, weshalb ein Generierungsschema fehlschlagen kann und die Erzeugung einer zulässigen Lösung nicht mehr generell in Polynomialzeit möglich ist. Beispiel 5.1 verdeutlicht anhand zweier Vorgänge und einer kumulativen Ressource, weshalb bei der Generierung eines Schedules Unzulässigkeiten auftreten können.

**Beispiel 5.1:** Eine Instanz  $\mathcal{I}_{\text{RCPSP}/c}$  besteht aus zwei Vorgängen 1 und 2 der Dauer  $d_1 = d_2 = 1$ , sowie einer kumulativen Ressource L1. Vorgang 1 nutzt  $r_{1,L1}^{y,\text{end}} = 2$  Einheiten von L1. Vorgang 2 nutzt  $r_{2,L1}^{y,\text{start}} = -2$  Einheiten von L1. Wird L1 als Lager interpretiert, lagert Vorgang 1 zwei Stoffeinheiten zu seinem Ende ein und Vorgang 2 diese beiden Stoffeinheiten zu seinem Start wieder aus. Es gilt ein Mindestbestand von  $\underline{R}_{L1}^y = 0$  und ein Höchstbestand von  $\overline{R}_{L1}^y = 1$ .

Um eine Lösung zu  $\mathcal{I}_{\text{RCPSP}/c}$  zu generieren, kann entweder zuerst Vorgang 1 und danach Vorgang 2 eingeplant werden oder umgekehrt. Wird zuerst Vorgang 1 eingeplant, wird der Höchstbestand von L1 zum Zeitpunkt eins überschritten (vgl. Abbildung 5.1a). Durch Einplanung von Vorgang 2 zum Zeitpunkt eins (d. h.  $S_2 := 1$ ) wird die Zulässigkeit wiederhergestellt (vgl. Abbildung 5.1b). Bei umgekehrter Einplanungsreihenfolge wird der Mindestbestand zunächst unterschritten und anschließend die Zulässigkeit wiederhergestellt. In jedem Fall muss in einer zulässigen Lösung der Endzeitpunkt von Vorgang 1 mit dem Startzeitpunkt von Vorgang 2 zusammenfallen.

Angenommen, es existiert eine weitere kumulative Ressource L2 mit denselben Höchst- und Mindestbeständen, d. h.  $\underline{R}_{L2}^y = 0$  und  $\overline{R}_{L2}^y = 1$ . Hiervon beansprucht Vorgang 2 zwei Einheiten zu seinem Ende und Vorgang 1 gibt zwei Einheiten zu seinem Start frei. In einer für L2 zulässigen Lösung muss das Ende von Vorgang 2 mit dem Start von Vorgang 1 zusammenfallen, was jedoch eine Unzulässigkeit für L1 nach sich ziehen würde. Folglich ist die Instanz in diesem Fall unlösbar.  $\square$



**Abbildung 5.1:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.1: a) nach Einplanung von Vorgang 1; b) nach Einplanung von Vorgang 2.

Der in Beispiel 5.1 beschriebene Fall der Unlösbarkeit aufgrund sich entgegengesetzter kumulativer Ressourcennutzungen ist einfach zu erkennen. Für größere Instanzen mit vielen Vorgängen, kumulativen Ressourcen und zeitlichen Abhängigkeiten ist allerdings kein effizienter Algorithmus bekannt, der Unzulässigkeiten für jede Instanz  $\mathcal{I}_{\text{RCPSP}/c}$  erkennt. Dies deckt sich mit der von Neumann, Schwindt und Zimmermann (2003, S. 130) nachgewiesenen NP-Vollständigkeit des Entscheidungsproblems zur Existenz einer zulässigen Lösung des (RCPSP/c). Die NP-Vollständigkeit eines Entscheidungsproblems zur Existenz einer zulässigen Lösung impliziert die NP-Schwere des zugehörigen Optimierungsproblems (Garey und Johnson 1979). Lediglich einige Spezialfälle mit nur einer kumulativen

Ressource und bestimmter Belegung der Problemparameter lassen sich in Polynomialzeit optimal lösen (Briskorn, Choi u. a. 2010).

Ein Generierungsschema für kumulative Ressourcen könnte auch so ausgestaltet werden, dass in einer Iteration mehrere Vorgänge simultan eingeplant werden. Damit würden Unzulässigkeiten zeitgleich zu ihrer Entstehung wieder aufgehoben. Dazu muss jedoch im Voraus bestimmt werden, welche Vorgänge simultan eingeplant werden. Für Beispiel 5.1 müsste im Voraus bestimmt werden, dass Vorgang 1 und 2 auf eine Weise simultan eingeplant werden müssen, sodass das Ende von Vorgang 1 mit dem Anfang von Vorgang 2 zusammenfällt. Die Komplexität bei der Erzeugung einer zulässigen Lösung für das (RCPSP/c) würde sich nun in diesen vorgeschalteten Rechenschritt verlagern und wäre nicht aufgehoben.

Zusammenfassend kann festgehalten werden, dass kumulative Ressourcen eine deutlich größere Herausforderung bei der Entwicklung eines Lösungsverfahrens darstellen als erneuerbare Ressourcen.

## 5.2 Existierende Ansätze in der Literatur

Dieser Abschnitt fokussiert sich auf die Analyse existierender Lösungsverfahren für Schedulingprobleme mit kumulativen Ressourcen. Die Ergebnisse dieser Analyse bilden die Grundlage für die Entwicklung eines Lösungsverfahrens für das (RCPSP/M) im weiteren Verlauf dieses Kapitels. Abschnitt 5.2.1 dient der Übersicht über relevante Literatur, die sich mit Lösungsverfahren zum (RCPSP/c) oder zu verwandten Problemstellungen befasst. In den Abschnitten 5.2.2 und 5.2.3 werden zwei existierende Lösungsverfahren detaillierter erläutert. Abschnitt 5.2.4 enthält eine Schlussfolgerung, inwiefern sich die existierenden Ansätze in der Literatur auf die Lösung des (RCPSP/M) übertragen lassen.

Für das Stoffstrom- und das Schedulingmodell wurde bereits in Abschnitt 3.5 eine Abgrenzung von existierenden Ansätzen in der Literatur vorgenommen.

### 5.2.1 Übersicht

Wie in Abschnitt 2.2.6 erläutert, werden kumulativen Ressourcen alternativ auch als Lagerressourcen (Carlier und Moukrim 2015) oder Reservoirs (Laborie 2003; Sourd und Rogerie 2005) bezeichnet. Das Schedulingproblem mit kumulativen Ressourcen wird in manchen Literaturbeiträgen auch als Schedulingproblem mit Ressourcen, die konsumiert und produziert werden, bezeichnet (Carlier, Moukrim und Xu 2009; Carlier und Moukrim 2015). Strenggenommen handelt es sich bei Letzterem nach der Definition von Carlier, Moukrim und Xu (2009) um einen Spezialfall des (RCPSP/c), in welchem lediglich ein Mindestbestand, aber kein Höchstbestand berücksichtigt wird. In Abschnitt 4.5.3 wurde beschrieben, dass sich Mindest- und Höchstbestände ineinander transformieren lassen. Diese Transformation erlaubt es, auch im Schedulingproblem mit Ressourcen, die konsu-

miert und produziert werden können, beliebige Höchst- und Mindestbestände abzubilden. Die Problemformulierung von Carlier, Moukrim und Xu (2009) unterscheidet sich allerdings auch dadurch vom (RCPSP/c), dass die Nutzungsarten auf den Start und das Ende eines Vorgangs eingeschränkt sind. Grund dafür ist, dass die Autoren statt der in dieser Arbeit verwendeten vorgangsbasierten Formulierung eine ereignisbasierte Formulierung verwenden. Und da Ereignisse augenblicklicher Art sind, können auch nur augenblickliche Nutzungsarten modelliert werden. Abgesehen davon sind die ereignis- und die vorgangsbasierte Modellierung ineinander transformierbar.

Wesentliche Beiträge zur Formulierung und Lösung des (RCPSP/c) wurden Anfang der 2000er Jahre von Neumann, Schwindt und Trautmann geleistet. Ihre für diese Arbeit relevanten Forschungsergebnisse sind in Schwindt und Trautmann (2000), Neumann und Schwindt (2002) und Neumann, Schwindt und Trautmann (2005) veröffentlicht. Auf die darin verwendeten Schedulingmodelle wurde bereits in Abschnitt 3.5.2 eingegangen. Hinsichtlich der Lösungsverfahren ähneln sich alle drei Artikel. In Neumann und Schwindt (2002) wird ein Branch-and-Bound-Verfahren zur exakten Lösung des (RCPSP/c) vorgestellt, sofern lediglich eine Nutzung zum Vorgangsstart oder -ende erfolgt. Auch in Schwindt und Trautmann (2000) wird dieses Branch-and-Bound-Verfahren verwendet, wobei im Vergleich zu Neumann und Schwindt (2002) mehr die Anwendung in der Prozessindustrie und weniger die theoretischen Erkenntnisse im Vordergrund stehen. Neumann und Schwindt (2002) kann somit als die ausführliche Erläuterung der in Schwindt und Trautmann (2000) skizzierten Idee eines Lösungsverfahrens für ein Schedulingproblem unter kumulativen Ressourcenrestriktionen betrachtet werden. In Neumann, Schwindt und Trautmann (2005) wird das in den beiden vorangegangenen Beiträgen betrachtete Modell und Lösungsverfahren um die kumulative Ressourcennutzung während der Vorgangsausführung erweitert. Die Autoren sprechen in diesem Zusammenhang von „kontinuierlichen kumulativen Ressourcen“. Außerdem wird eine Möglichkeit zur Beschneidung des Suchbaums vorgestellt, mit welchem ein exaktes Branch-and-Bound-Verfahren zu einem heuristischen Branch-and-Bound-Verfahren wird. Da das Verfahren von Neumann, Schwindt und Trautmann (2005) die Verfahren aus den früheren Beiträgen generalisiert, wird es in Abschnitt 5.2.2 dieser Arbeit detaillierter analysiert.

Schwindt, Fink u. a. (2007) betrachten in ihrem Konferenzbeitrag dieselbe Problemstellung wie Schwindt und Trautmann (2000) und Neumann und Schwindt (2002), verwenden jedoch eine andere Klasse von Lösungsverfahren. Statt mit einem Branch-and-Bound-Verfahren, welches zur exakten Lösung dient, befassen sie sich mit einem prioritätsregelbasierten Verfahren. Ein solcher Ansatz ist naheliegend, da prioritätsregelbasierte Verfahren zu den bekanntesten Heuristiken für das (RCPSP) zählen und daher bereits ausgiebig erforscht wurden. In Abschnitt 5.2.3 dieser Arbeit wird das Verfahren von Schwindt, Fink u. a. (2007) erläutert. Das von Carlier, Moukrim und Xu (2009) betrachtete Problem und ihr vorgeschlagenes Lösungsverfahren ist in Teilen mit Schwindt, Fink u. a. (2007) vergleichbar. Deshalb wird in Abschnitt 5.2.3 ebenfalls kurz darauf eingegangen. Darüber hinaus liefern Carlier, Moukrim und Xu (2009) einige theoretische Erkenntnisse zur Lösbarkeit von Schedulingproblemen mit kumulativen Ressourcen.

In etwa parallel zu Neumann und Schwindt (2002) hat sich Laborie (2003) ebenfalls mit der exakten Lösung von Schedulingproblemen unter kumulativen Ressourcenrestriktionen befasst. Für beide Beiträge gilt die Einschränkung, dass nur Nutzungen zum Vorgangsstart oder -ende möglich sind. Ansonsten unterscheiden sich die gewählten Lösungsansätze deutlich voneinander. Laborie (2003) fokussiert sich auf Techniken der sogenannten *Constraint Propagation*. Bei der Constraint Propagation werden die Werte, welche Entscheidungsvariablen annehmen können, schrittweise eingeschränkt, indem ihre Unzulässigkeit nachgewiesen wird (Sourd und Rogerie 2005, S. 511). Für Schedulingprobleme bedeutet dies zunächst, dass die mit den Start- und Endzeitpunkten der Vorgänge verbundenen Ereignisse basierend auf einem Enumerationsansatz sortiert werden. Über verschiedene Alternativen bei der Sortierung wird verzweigt. Für jeden Knoten im dadurch entstehenden Suchbaum werden die vorhandenen Informationen ausgenutzt, um eine Constraint Propagation durchzuführen. Wenn beispielsweise der spätestmögliche Startzeitpunkt eines Vorgangs  $i$  echt kleiner als sein frühestmöglicher Endzeitpunkt ist, wird sich Vorgang  $i$  mit Sicherheit im Zeitraum zwischen diesen beiden Zeitpunkten in Ausführung befinden. Wenn dann für einen anderen Vorgang  $i'$  in diesem Zeitraum nicht mehr genügend Ressourcen zur Verfügung stehen, kann nachgewiesen werden, dass  $j$  außerhalb dieses Zeitraums ausgeführt werden muss. Dadurch wird der Lösungsraum beschnitten. Die auf dieser exemplarischen Argumentation basierende Technik der Constraint Propagation wird als Timetabling bezeichnet. Laborie (2003) schlägt als Ergänzung zum Timetabling zwei neuartige Techniken vor, welche er „Energy Precedence Constraint“ und „Balance Constraint“ nennt. Die Energy Precedence Constraint ist nur für erneuerbare Ressourcen geeignet (im Beitrag als „discrete resources“ bezeichnet). Die Balance Constraint ist hingegen für kumulative Ressourcen geeignet (im Beitrag als „reservoirs“ bezeichnet). Grundlage für die Balance Constraint bildet die Berechnung unterer und oberer Schranken für das kumulative Ressourcenprofil vor und nach einer Ein- oder Auslagerung durch einen Vorgang. Falls diese Schranken auf eine mögliche Überschreitung einer kumulativen Ressourcenkapazität schließen lassen, werden mittels Constraint Propagation die Ausführungszeiträume bestimmter Vorgänge eingeschränkt, um die Überschreitung zu verhindern. Laborie (2003, S. 171f) gelingt es mit Hilfe der Balance Constraint, zwölf Instanzen optimal zu lösen bzw. ihre Unzulässigkeit zu beweisen, die Neumann und Schwindt (2002) nicht lösen konnten.

Das Modell von Laborie (2003) wird von Sourd und Rogerie (2005) um die gleichmäßige Nutzung während der Vorgangsausführung erweitert. Analog zu Laborie (2003) befassen sich auch Sourd und Rogerie (2005) mit Techniken der Constraint Propagation. Sie verwenden jedoch nur das Timetabling. Das darauf basierende exakte Lösungsverfahren testen sie nur mit sehr kleinen Instanzen, die zwischen fünf und zehn Vorgänge enthalten.

Neben der Lösung mittels Branch-and-Bound oder Constraint Propagation, stellt die (gemischt-)ganzzahlige lineare Programmierung generell ein weiterer Weg zur exakten Lösung von Optimierungsproblemen dar (Kolisch 1996b, S. 321). Koné u. a. (2013) wählen diesen Weg zur Lösung des (RCPSP/c). Auch Kyriakidis u. a. (2012) formulieren ein Schedulingproblem mit kumulativen Ressourcen als gemischt-ganzzahliges lineares Programm (MILP) (engl. mixed-integer linear program). Sie verwenden den Begriff der kumulativen

Ressource nicht explizit, sprechen aber von „activity consumption and production coefficients“ (Kyriakidis u. a. 2012, S. 372), woraus sich die modelltheoretische Übereinstimmung mit kumulativen Ressourcen erschließen lässt. MILPs bieten den Vorteil der Flexibilität für Erweiterungen. Außerdem stehen kommerzielle Solver zu deren Lösung zur Verfügung. Bei einem *Solver* handelt es sich um ein Programm, das vorimplementierte Lösungsverfahren sowie Schnittstellen zum Zugriff darauf bereitstellt. Laut Koné u. a. (2013, S. 26) handelt es sich bei der Formulierung als MILP und der Lösung mittels eines kommerziellen Solvers oft um die einzige Möglichkeit für industrielle Anwender, ihre in der Praxis auftretenden Optimierungsprobleme zu lösen. Allerdings sind MILPs hinsichtlich ihrer Effizienz im Vergleich zu den oben genannten problemspezifischen Lösungsverfahren nicht konkurrenzfähig, wie Koné u. a. (2013, S. 26) feststellen.

Ein mit der linearen Programmierung vergleichbarer Lösungsweg, der sich für Schedulingprobleme etabliert hat, ist die Constraintprogrammierung. Für Details zum dahinterstehenden Paradigma wird auf Abschnitt 5.8 verwiesen. Zur Lösung eines Constraintprogramms ist geeigneter Solver erforderlich. Trotz der begrifflichen Ähnlichkeit ist zu beachten, dass die Constraintprogrammierung nicht mit der Constraint Propagation verwechselt werden darf: Während unter der Constraintprogrammierung eine spezielle Art der Problemformulierung verstanden wird, handelt es sich bei der Constraint Propagation um einen Bestandteil eines Lösungsverfahrens. Bofill u. a. (2022) vergleichen die Leistungsfähigkeit von linearer Programmierung und Constraintprogrammierung an einem aus der Industrie motivierten Problem. Dieses Problem besteht aus der Bestimmung eines Schedules für eine Anlage zur Probenanalyse im Labor. Aufgrund begrenzter Abstellflächen innerhalb dieser Anlage liegen kumulative Ressourcenrestriktionen vor, weshalb Bofill u. a. (2022) ihr Problem als (RCPSP/c) modellieren. Sie schränken sich dabei jedoch auf Nutzungen zum Vorgangsstart und -ende ein. Die untersuchten Instanzen weisen weitere Einschränkungen auf: So existieren ausschließlich kumulative Ressourcennutzungen in Höhe von  $\{-1, 0, 1\}$  sowie erneuerbare Ressourcennutzungen in Höhe von  $\{0, 1\}$  (Bofill u. a. 2022, S. 8). Alle erneuerbaren Ressourcenkapazitäten betragen eins. Es sei angemerkt, dass Bofill u. a. (2022) auch einen dritten Lösungsansatz untersuchen, welcher aus der Formulierung des Problems mit Satisfiability Modulo Theories besteht. Da dieser Ansatz schlechter als die Constraintprogrammierung abschneidet, wird er in der vorliegenden Arbeit nicht weiter verfolgt.

Wie an den oben zitierten Beiträgen erkennbar ist, hat die Forschungsaktivität zum Schedulingproblem mit kumulativen Ressourcen in den vergangenen Jahren nachgelassen. Stattdessen hat sich der Fokus auf eine verwandte, spezialisierte Problemstellung aus der Maschinenbelegungsplanung verschoben. Als Beginn einer Reihe von Publikationen hierzu kann Briskorn, Choi u. a. (2010) betrachtet werden. Sie stellen ein neuartiges Ein-Maschinen-Schedulingproblem vor, in welchem der Bestand eines zentralen Lagers berücksichtigt wird. Ein Ein-Maschinen-Schedulingproblem ist eine Problemklasse aus der Maschinenbelegungsplanung, die sich mit der zeitlichen Planung einer Menge von Arbeitsgängen (engl. jobs) befasst, die allesamt sequenziell auf derselben Maschine bearbeitet werden müssen. Zeitliche Abhängigkeiten zwischen Arbeitsgängen bestehen nicht. Arbeitsgänge sind vergleichbar mit Vorgängen im Projektscheduling. Der Begriff dient

der sprachlichen Abgrenzung zwischen den Anwendungsgebieten. Für jeden Arbeitsgang sind eine Dauer und ein Fälligkeitsdatum gegeben. Außerdem lagert jeder Arbeitsgang entweder eine bestimmte Menge in das zentrale Lager ein oder eine bestimmte Menge aus dem zentralen Lager aus. Das zentrale Lager hat keinen Höchstbestand, sondern es ist lediglich zu berücksichtigen, dass der Bestand nicht negativ werden darf. Dieses von Briskorn, Choi u. a. (2010) betrachtete Problem ist eine Spezialisierung des (RCPSP/c) mit Zeitbeziehungen, da es auch wie folgt modelliert werden kann: Die Maschine wird als eine erneuerbare Ressource der Kapazität eins modelliert. Jeder Arbeitsgang benötigt eine Einheit dieser erneuerbaren Ressource. Das Fälligkeitsdatum jedes Arbeitsgangs wird über eine Zeitbeziehung zum fiktiven Startvorgang und einen Zeitabstand in Höhe des negativen Fälligkeitsdatums modelliert. Das zentrale Lager wird als eine kumulative Ressource mit einem Mindestbestand von null modelliert. Die Nutzungsarten sind nicht relevant, da Arbeitsgänge ohnehin nur sequenziell ausgeführt werden können (vgl. Morsy und Pesch 2015, S. 646).

Briskorn, Choi u. a. (2010) untersuchen die Komplexität ihres neuartigen Ein-Maschinen-Schedulingproblems unter Verwendung verschiedener Zielsetzungen und unter Berücksichtigung weiterer Sonderfälle. Briskorn, Jaehn u. a. (2013) und Morsy und Pesch (2015) fokussieren sich auf die Zielsetzung zur Minimierung der gewichteten Fertigstellungszeitpunkte und stellen hierzu Lösungsverfahren vor. Briskorn und Leung (2013) legen den Fokus auf die Zielsetzung der Minimierung der maximalen Verspätung. Briskorn und Pesch (2013) betrachten zusätzlich zum Mindestbestand von null auch einen frei wählbaren Höchstbestand, der für das zentrale Lager einzuhalten ist. Im Gegensatz zum Fälligkeitsdatum in Briskorn, Choi u. a. (2010) berücksichtigen Davari u. a. (2020) ein Freigabedatum, d. h. eine Zeitrestriktion, die für jeden Arbeitsgang einen frühesten Zeitpunkt vorgibt.

### 5.2.2 Branch-and-Bound-Verfahren von Neumann u. a.

Das Branch-and-Bound-Verfahren von Neumann, Schwindt und Trautmann (2005) ist generell zur Lösung des (RCPSP/c) geeignet. Nachfolgend wird zur Vereinfachung nur der Fall betrachtet, dass für kumulative Ressourcen Höchstbestände, aber keine Mindestbestände vorgegeben sind (d. h.  $\underline{R}_k^Y = -\infty$  für alle  $k \in \mathcal{R}^Y$ ). Mindestbestände können jedoch mit denselben Konzepten berücksichtigt werden.

Das Lösungsverfahren beruht auf einem *relaxationsbasierten Enumerationsansatz*. Das bedeutet, es wird wiederholt die Ressourcenrelaxation einer Instanz  $\bar{\mathcal{I}}_{\text{RCPSP}/c}$  gelöst. Bei der Ressourcenrelaxation des (RCPSP/c) werden die erneuerbaren und die kumulativen Ressourcenrestriktionen relaxiert (= entfernt). Eine Lösung der Ressourcenrelaxation wird anschließend darauf untersucht, ob die erneuerbaren und kumulativen Ressourcenrestriktionen eingehalten sind. Falls dies für mindestens einen Zeitpunkt vom Projektstart bis -ende nicht der Fall ist, wird von einem erneuerbaren bzw. kumulativen *Ressourcenkonflikt* gesprochen. In jeder Iteration des Verfahrens wird derjenige Ressourcenkonflikt ausgewählt, der zum kleinsten Zeitpunkt auftritt. Durch Einführung zusätzlicher Restriktionen werden Vorgänge auseinandergezerrt, sodass dieser Ressourcenkonflikt aufgehoben wird.



Da es mehrere Alternativen für solche zusätzlichen Restriktionen geben kann, muss enumeriert werden. Für jede Alternative entsteht ein neuer Zweig im Suchbaum, was als *Branching* bezeichnet wird. Es ist nicht effizient, alle Zweige weiterzuverfolgen. Stattdessen ist es sinnvoll, für jeden Zweig eine *untere Schranke* für den optimalen Zielfunktionswert zu bestimmen, d. h. ein Wert, der ohne Kenntnis einer konkreten Lösung mit Sicherheit nicht mehr unterschritten werden kann. Parallel dazu wird eine *obere Schranke* für den optimalen Zielfunktionswert bestimmt. Falls die untere Schranke größer oder gleich der oberen Schranke ist, ist vom betrachteten Zweig ausgehend keine Verbesserung mehr zu erwarten. Der Suchbaum kann an dieser Stelle abgeschnitten werden, was als *Bounding* bezeichnet wird. Zusammengesetzt ergibt sich ein *Branch-and-Bound-Verfahren*. Aufgrund des Enumerationsansatzes handelt es sich um ein exaktes Lösungsverfahren. Für eine detaillierte Erläuterung wird auf Franck u. a. (2001) oder Zimmermann u. a. (2006) verwiesen. Beide Beiträge befassen sich ausschließlich mit erneuerbaren Ressourcen. Das Verfahrensprinzip kann jedoch unverändert auf kumulative Ressourcen übertragen werden.

Zunächst werden Instanzen  $\mathcal{I}_{\text{RCPSP}/c}$  betrachtet, in denen nur eine **Nutzung zum Vorgangsstart oder -ende** erfolgt, d. h.  $r_{ik}^{y,lin} = 0$  für alle  $i \in V$  und  $k \in \mathcal{R}^y$ . Falls in einer Lösung  $S$  der Ressourcenrelaxation der Höchstbestand einer kumulativen Ressource  $k$  überschritten wird, wird  $t^c := \min\{0 \leq t \leq S_{I+1} \mid r_k^y(S, t) > \bar{R}_k^y\}$  gesetzt. Damit bezeichnet  $t^c$  den kleinsten Zeitpunkt, zu dem ein Ressourcenkonflikt auftritt. Anschließend wird eine Teilmenge  $B$  aller einlagernden Vorgänge gebildet, deren Ausführung bis einschließlich Zeitpunkt  $t^c$  gestartet wurde, d. h.  $B \subset \{i \in V \mid r_{ik}^{y,start} + r_{ik}^{y,end} > 0 \wedge S_i \leq t^c\}$ . Dabei soll  $B$  eine inklusionsminimale Menge sein, sodass die Ungleichung

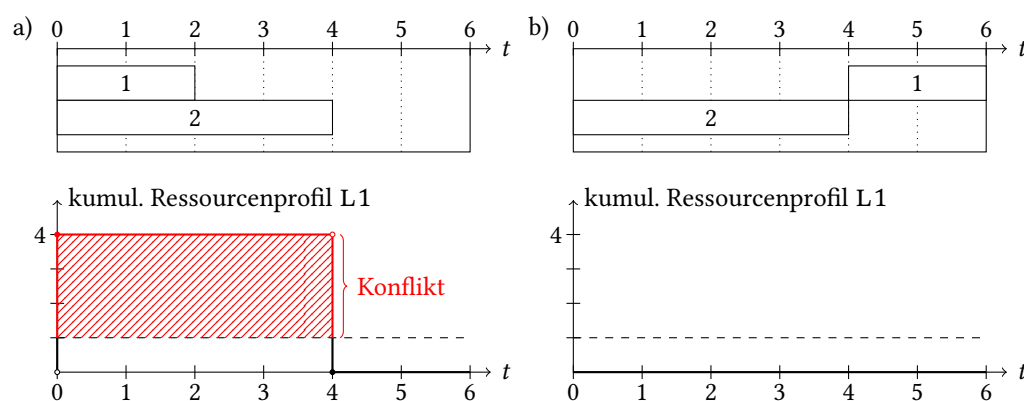
$$\sum_{i \in \mathcal{A}^y(S, t^c) \setminus B} r_{ik}^y(S_i, t^c) \leq \bar{R}_k^y \quad (5.1)$$

eingehalten wird. Das bedeutet, es werden nur gerade so viele Vorgänge zu  $B$  hinzugefügt, dass der Ressourcenkonflikt zum Zeitpunkt  $t^c$  aufgelöst wird, aber nicht mehr. Die Menge  $B$  umfasst diejenigen Vorgänge, die zur Auflösung des Ressourcenkonflikts verschoben werden sollen. Dazu werden Zeitbeziehungen zwischen den Vorgängen der Menge  $B$  und einem Vorgang  $i' \notin \mathcal{A}^y(S, t^c)$  mit  $r_{i'k}^{y,start} + r_{i'k}^{y,end} < 0$  zur Instanz hinzugefügt. Diese Zeitbeziehungen sorgen in der Ressourcenrelaxation dafür, dass die Einlagerungszeitpunkte der einlagernden Vorgänge  $i \in B$  mit dem Auslagerungszeitpunkt des auslagernden Vorgangs  $i'$  zusammenfallen (siehe auch Neumann, Schwindt und Zimmermann 2003, S. 129ff; Schwindt und Trautmann 2000, S. 516f). Beispiel 5.2 veranschaulicht dies exemplarisch für zwei Vorgänge. Über die verschiedenen Alternativen für  $B$  und für  $i'$  wird verzweigt. Zu späteren Zeitpunkten  $t > t^c$  können weitere Ressourcenkonflikte vorhanden sein oder durch die zusätzlichen Zeitbeziehungen neu auftreten, die in nachfolgenden Iterationen nach demselben Schema aufgelöst werden. Jede Ressourcenrelaxation ist ein (TPSP/max), welches mit dem Algorithmus von Neumann, Schwindt und Zimmermann (2003, S. 17) in Polynomialzeit optimal gelöst werden kann.

**Beispiel 5.2:** Eine Instanz  $\mathcal{I}_{\text{RCPSP}/c}$  besteht aus zwei Vorgängen 1 und 2 der Dauer  $d_1 = 2$  und  $d_2 = 4$ , sowie einer kumulativen Ressource L1. Vorgang 1 nutzt  $r_{1,L1}^{y,start} = 4$  Einheiten von

L1. Vorgang 2 nutzt  $r_{2,L1}^{y,end} = -4$  Einheiten von L1. Wird L1 als Lager interpretiert, lagert Vorgang 1 vier Stoffeinheiten zu seinem Start ein und Vorgang 2 diese vier Stoffeinheiten zu seinem Ende wieder aus. Es gilt ein Höchstbestand von  $\bar{R}_{L1}^y = 1$ . Fiktive Vorgänge werden zur Vereinfachung weggelassen. Zwischen den Vorgängen 1 und 2 existiert keine Vorrangbeziehung.

In der initialen Lösung zur Ressourcenrelaxation dieser Instanz werden beide Vorgänge zum Zeitpunkt 0 eingeplant, d. h.  $S_1 = S_2 := 0$ . In dieser Lösung sind jedoch Ressourcenkonflikte vorhanden (vgl. Abbildung 5.2a). Der früheste Zeitpunkt, zu dem ein Ressourcenkonflikt auftritt, ist  $t^c = 0$ . Menge  $B$  wird auf  $B := \{1\}$  gesetzt. Hierbei handelt es sich um die einzige Alternative, da nur Vorgang 1 einlagert. Als auslagernder Vorgang steht nur Vorgang 2 zur Verfügung, d. h.  $i' := 2$ . Durch Hinzufügen einer Zeitbeziehung (2, 1) mit  $\delta = 4$  wird die Zeitrestriktion  $S_1 \geq S_2 + 4$  zur Ressourcenrelaxation hinzugefügt. Dadurch fallen die Einlagerung von Vorgang 1 und die Auslagerung von Vorgang 2 zusammen auf den Zeitpunkt vier, wodurch der Ressourcenkonflikt in der darauffolgenden Iteration aufgelöst ist (vgl. Abbildung 5.2b).  $\square$



**Abbildung 5.2:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.2 bei Nutzung zum Vorgangsstart bzw. -ende: a) initiale Ressourcenrelaxation; b) nach Auflösung des Konflikts.

Falls in Instanzen  $\mathcal{I}_{RCSP/c}$  die Nutzung kumulativer Ressourcen **gleichmäßig während der Vorgangsausführung** erfolgt, wird die Auflösung der Ressourcenkonflikte aufwendiger. Die Veränderungen des kumulativen Ressourcenprofils sind mehr augenblicklicher Art, sondern verteilen sich über einen Zeitraum. Deshalb können nicht mehr einfach Zeitbeziehungen eingeführt werden, sodass die Zeitpunkte von Einlagerungen mit den Zeitpunkten von Auslagerungen zusammenfallen und der Konflikt aufgehoben wird. Stattdessen gehen Neumann, Schwindt und Trautmann (2005) wie folgt vor: Angenommen, sämtliche kumulativen Ressourcennutzungen erfolgen gleichmäßig während der Vorgangsausführung, d. h.  $r_{ik}^{y,start} = r_{ik}^{y,end} = 0$  für alle  $i \in V$  und  $k \in \mathcal{R}^y$ . Falls in einer Lösung  $S$  einer

Ressourcenrelaxation der Höchstbestand einer kumulativen Ressource  $k$  überschritten wird, wird

$$t^c := \min \left( \left\{ 0 \leq t \leq S_{I+1} \mid r_k^y(S, t) > \bar{R}_k^y \right\} \cap \left( \bigcup_{i' \in V \mid r_{i'k}^{y,lin} < 0} \{S_{i'}\} \cup \bigcup_{i' \in V \mid r_{i'k}^{y,lin} > 0} \{S_{i'} + d_{i'}\} \right) \right) \quad (5.2)$$

gesetzt. Verbal bedeutet dies, dass  $t^c$  auf den kleinsten Zeitpunkt gesetzt wird, zu dem ein Ressourcenkonflikt auftritt und ein auslagernder Vorgang startet oder ein einlagernder Vorgang endet. Zu den Zeitpunkten, zu denen ein auslagernder Vorgang startet oder ein einlagernder Vorgang endet, ist die Überschreitung des Höchstbestands jeweils am größten. Daher reicht es aus, lediglich diese Konflikte zu betrachten. Nach Bestimmung von  $t^c$  werden die Vorgänge auf zwei disjunkte Mengen  $A$  und  $B$  aufgeteilt. Die Aufteilung kann vollständig oder teilweise sein, d. h.  $A \cup B \subseteq V$ . Für jede mögliche Aufteilung wird verzweigt, solange der Ressourcenkonflikt nicht behoben ist. Die Menge  $A$  enthält Vorgänge, die zur Auflösung des Konflikts nicht verschoben werden sollen. Dies wird mit den Ungleichungen

$$S_i + d_i \leq t^c \quad (\forall i \in A \mid r_{ik}^{y,lin} > 0) \quad \text{und} \quad (5.3)$$

$$S_i \geq t^c \quad (\forall i \in A \mid r_{ik}^{y,lin} < 0) \quad (5.4)$$

sichergestellt. Sie besagen, dass das Ende einlagernder Vorgänge nicht später als  $t^c$  bzw. der Start auslagernder Vorgänge nicht früher als  $t^c$  stattfinden darf. Beides würde ansonsten zur Auflösung des Konflikts beitragen, was jedoch für  $A$  nicht gewünscht ist. Die Menge  $B$  erfüllt denselben Zweck wie oben: Sie umfasst diejenigen Vorgänge, die zur Auflösung des Ressourcenkonflikts verschoben werden sollen. Genauer ausgedrückt, sollen die einlagernden Vorgänge der Menge  $B$  nach rechts und die auslagernden Vorgänge der Menge  $B$  nach links verschoben werden. Um das Ausmaß der Verschiebung zu bestimmen, werden für alle Vorgänge  $i \in B$  Entscheidungsvariablen  $x_i \in [0, 1]$  eingeführt, die jeweils den Ausführungsfortschritt des Vorgangs  $i$  zum Zeitpunkt  $t^c$  beschreiben. Um den Ressourcenkonflikt aufzuheben, muss

$$\sum_{i \in A \mid r_{ik}^{y,lin} > 0} r_{ik}^{y,lin} + \sum_{i \in B} r_{ik}^{y,lin} x_i \leq \bar{R}_k^y \quad (5.5)$$

eingehalten werden. Die Kopplung zwischen den Entscheidungsvariablen  $x_i$  und den Startzeitpunkten  $S_i$  der Vorgänge der Menge  $B$  wird über die Ungleichungen

$$S_i + d_i x_i \geq t^c \quad (\forall i \in B \mid r_{ik}^{y,lin} > 0) \quad \text{und} \quad (5.6)$$

$$S_i + d_i x_i \leq t^c \quad (\forall i \in B \mid r_{ik}^{y,lin} < 0) \quad (5.7)$$

ausgedrückt. Es ist zu erkennen, dass die Ungleichheitszeichen im Vergleich zu den Ungleichungen (5.3) und (5.4) vertauscht sind. Dadurch wird die erwünschte Rechtsverschiebung der einlagernden Vorgänge bzw. die Linksverschiebung der auslagernden Vorgänge erzielt.

In Ungleichung (5.5) dient  $x_i$  zur Berechnung der anteiligen Ressourcennutzung und in den Ungleichungen (5.6) und (5.7) dient  $x_i$  zur Berechnung des anteilig vergangenen Ausführungszeitraums. Dieser Zusammenhang besteht nur im Fall einer gleichmäßigen Nutzung während der Vorgangsausführung, weshalb die genannten Ungleichungen auch nur für diese und nicht für beliebige Nutzungsarten geeignet sind.

Statt zusätzlicher Zeitbeziehungen, wie bei den Nutzungen zum Vorgangstart und -ende, werden nun in jeder Iteration des Branch-and-Bound-Verfahrens die Ungleichungen (5.3) bis (5.7) für die jeweils gewählten Mengen  $A$  und  $B$  zur Ressourcenrelaxation hinzugefügt. Um gemäß Neumann, Schwindt und Trautmann (2005, S. 504) die „Endlichkeit des Lösungsverfahrens“ zu gewährleisten, darf jedoch  $t^c$  nicht als fixer Wert angenommen werden, sondern muss entscheidungsabhängig formuliert werden. Dazu muss reproduziert werden, welches  $i'$  in Gleichung (5.2) ausschlaggebend für die Festlegung von  $t^c$  war. Dann kann  $t^c$  in (5.3) bis (5.7) durch  $S_{i'}$  bzw.  $S_{i'} + d_{i'}$  ersetzt werden. Beispiel 5.3 veranschaulicht dies.

Eine Lösung mit dem Algorithmus von Neumann, Schwindt und Zimmermann (2003, S. 17) ist nicht möglich, da es sich aufgrund der Entscheidungsvariablen  $x_i$  nicht mehr um ein (TPSP/max) handelt. Es handelt sich jedoch um ein lineares Programm, das Neumann, Schwindt und Trautmann (2005) mit dem kommerziellen Solver IBM ILOG CPLEX (kurz: CPLEX) lösen. Ihr Verfahren wird dadurch zu einem hybriden Verfahren, welches einen Branch-and-Bound-Ansatz mit einer linearen Optimierung kombiniert.

**Beispiel 5.3:** Eine Instanz  $\bar{I}_{\text{RCPSP}/c}$  besteht unverändert zu Beispiel 5.2 aus zwei Vorgängen 1 und 2 mit  $d_1 = 2$  und  $d_2 = 4$ , sowie einer kumulativen Ressource L1 mit  $\bar{R}_{L1}^y = 1$ . Die kumulativen Ressourcennutzungen betragen  $r_{1,L1}^{y,lin} = 4$  und  $r_{2,L1}^{y,lin} = -4$ .

In der initialen Lösung zur Ressourcenrelaxation dieser Instanz werden beide Vorgänge zum Zeitpunkt null eingeplant, d. h.  $S_1 = S_2 := 0$ . In dieser Lösung ist jedoch ein kumulativer Ressourcenkonflikt vorhanden (vgl. Abbildung 5.3a). Gemäß Gleichung (5.2) wird  $t^c = 2$  bestimmt. Hierbei handelt es sich um den Endzeitpunkt des Vorgangs 1, zu dem die Überschreitung des Höchstbestands am größten ist. Eine mögliche Aufteilung der Vorgänge in die beiden Menge  $A$  und  $B$  ist  $A = \{1\}$  und  $B = \{2\}$ . Damit gilt

$$\text{gemäß (5.3): } S_1 + 2 \leq t^c,$$

$$\text{gemäß (5.5): } 4 - 4x_2 \leq 1,$$

$$\text{gemäß (5.7): } S_2 + 4x_2 \leq t^c.$$

Da Vorgang 1 in Gleichung (5.2) ausschlaggebend zur Festlegung von  $t^c$  war, kann  $t^c$  durch  $S_1 + d_1 (= S_1 + 2)$  ersetzt werden, d. h.

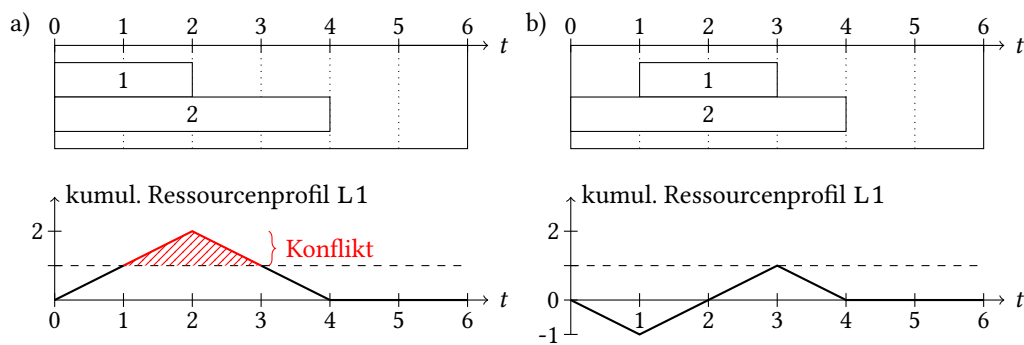
$$\text{gemäß (5.3): } S_1 + 2 \leq S_1 + 2,$$

$$\text{gemäß (5.5): } 4 - 4x_2 \leq 1 \quad \Leftrightarrow x_2 \geq 3/4,$$

$$\text{gemäß (5.7): } S_2 + 4x_2 \leq S_1 + 2 \quad \Leftrightarrow S_1 \geq S_2 + 4x_2 - 2.$$

Die um diese Ungleichungen ergänzte Ressourcenrelaxation lösen Neumann, Schwindt und Trautmann (2005, S. 504) mit CPLEX. In diesem Beispiel lässt sich jedoch argumentativ

eine Lösung herleiten: Die erste Ungleichung ist stets eingehalten. Um die Projektdauer zu minimieren, müssen  $S_1$  und  $S_2$  so klein wie möglich gewählt werden. Deshalb wird  $S_2 := 0$  und  $x_2 := 3/4$  gesetzt, wodurch die rechte Seite der Ungleichung  $S_1 \geq S_2 + 4x_2 - 2$  minimal wird. Damit kann  $S_1 := 1$  gewählt werden. Der Ressourcenkonflikt ist aufgelöst (vgl. Abbildung 5.3b).  $\square$



**Abbildung 5.3:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.3 bei gleichmäßiger Nutzung während der Vorgangsausführung: a) initiale Ressourcenrelaxation; b) nach Auflösung des Konflikts.

### 5.2.3 Prioritätsregelbasiertes Verfahren von Schwindt u. a.

Das prioritätsregelbasierte Verfahren von Schwindt, Fink u. a. (2007) ist zur Lösung des (RCPSP/c) geeignet, solange sich die Nutzungsarten der kumulativen Ressourcen auf die Nutzung zum Vorgangsstart und -ende beschränken, d. h.  $r_{ik}^{y,lin} = 0$  für alle  $i \in V$  und  $k \in \mathcal{R}^y$ . Nachfolgend wird zur Vereinfachung nur der Fall betrachtet, dass für kumulative Ressourcen Höchstbestände, aber keine Mindestbestände vorgegeben sind (d. h.  $\underline{R}_k^y = -\infty$  für alle  $k \in \mathcal{R}^y$ ). Mindestbestände können jedoch mit denselben Konzepten behandelt werden.

Prioritätsregelbasierte Verfahren bestehen aus einer Prioritätsregel und einem Generierungsschema. Schwindt, Fink u. a. (2007) fokussieren sich primär auf die Entwicklung eines Generierungsschemas, das für kumulative Ressourcen geeignet ist. Da ihr Generierungsschema über die Vorgänge iteriert, handelt es sich um ein serielles Generierungsschema, das grundsätzlich wie das in Abschnitt 2.3.1 beschriebene serielle Generierungsschema für das (RCPSP/max) abläuft. Um darüber hinaus die kumulativen Ressourcen zu berücksichtigen, gehen Schwindt, Fink u. a. (2007) wie folgt vor: Nach jeder Einplanung eines Vorgangs  $i^*$  (= der mittels Prioritätsregel ausgewählte, in einer Iteration einzuplanende Vorgang) wird geprüft, ob die Höchstbestände der kumulativen Ressourcen eingehalten sind. Falls zu einem Zeitpunkt  $t^c$  der Höchstbestand einer kumulativen Ressource  $k$  überschritten wird, werden auslagernde Vorgänge  $i \in \mathcal{E}$  (wobei  $\mathcal{E}$  die Menge der einplanbaren Vorgänge umfasst; vgl. Abschnitt 2.3.1) ausgewählt. Die spätestmöglichen Startzeitpunkte dieser ausgewählten Vorgänge  $i$  werden reduziert, sodass die Überschreitung des Höchstbestands

zum Zeitpunkt  $t^c$  nach ihrer Einplanung aufgehoben sein wird. Schwindt, Fink u. a. (2007, S. 1349) sprechen von „capacity-driven latest start times“. Dadurch ist es jedoch möglich, dass keine Vorgänge  $i \in \mathcal{E}$  in der darauffolgenden Iteration mehr einplanbar sind, weil ihre frühestmöglichen Startzeitpunkte ihre spätestmöglichen Startzeitpunkte überschreiten. In diesem Fall ist ein Ausplanungsschritt erforderlich. Ein Ausplanungsschritt besteht darin, einen für den kumulativen Ressourcenkonflikt verantwortlichen Vorgang  $i'$  zu identifizieren und auszulagern. Der frühestmögliche Startzeitpunkt von  $i'$  wird anschließend erhöht, sodass seine Einlagerung mit der Auslagerung eines der ausgewählten Vorgänge  $i \in \mathcal{E}$  zusammenfallen kann. Beispiel 5.4 veranschaulicht die Vorgehensweise exemplarisch für zwei Vorgänge. Es sei angemerkt, dass frühest- und spätestmögliche Startzeitpunkte über Zeitbeziehungen im Generierungsschema abgebildet werden können. Ein frühestmöglicher Startzeitpunkt entspricht einem zeitlichen Mindestabstand zum fiktiven Startvorgang und ein spätestmöglicher Startzeitpunkt einem zeitlichen Höchstabstand.

**Beispiel 5.4:** Eine Instanz  $\mathcal{I}_{\text{RCPSP}/c}$  besteht unverändert zu Beispiel 5.2 aus zwei Vorgängen 1 und 2 mit  $d_1 = 2$  und  $d_2 = 4$ , sowie einer kumulativen Ressource L1 mit  $\bar{R}_{L1}^Y = 1$ . Die kumulativen Ressourcennutzungen betragen  $r_{1,L1}^{Y,start} = 4$  und  $r_{2,L1}^{Y,end} = -4$ .

Angenommen, die Prioritätsregel schreibt vor, dass zunächst Vorgang 1 und dann Vorgang 2 eingeplant werden soll. Die Iterationen des Generierungsschemas nach Schwindt, Fink u. a. (2007) sind dann wie folgt:

- (1) Vorgang  $i^* := 1$  wird zum Zeitpunkt  $S_1 := 0$  eingeplant. Es wird festgestellt, dass zum Zeitpunkt  $t^c = 0$  der Höchstbestand der kumulativen Ressource L1 überschritten wird. Deshalb wird der auslagernde Vorgang  $i := 2 \in \mathcal{E}$  gewählt und dessen spätestmöglicher Startzeitpunkt auf  $-4$  gesetzt.
- (2) Vorgang  $i^* := 2$  kann nicht eingeplant werden, da sein spätestmöglicher Startzeitpunkt kleiner als sein frühestmöglicher Startzeitpunkt ( $= 0$ ) ist. Ausplanungsschritt: Vorgang  $i' := 1$  wird als verantwortlich für den kumulativen Ressourcenkonflikt identifiziert und ausgelagert, d. h.  $S_1 := \infty$ . Sein frühestmöglicher Startzeitpunkt wird auf vier gesetzt, wodurch seine Einlagerung mit der Auslagerung von Vorgang 2 zusammenfällt. Der spätestmöglicher Startzeitpunkt von Vorgang 2 wird gelöscht.
- (3) Vorgang  $i^* := 1$  wird zum Zeitpunkt  $S_1 := 4$  eingeplant. Es wird festgestellt, dass zum Zeitpunkt  $t^c = 4$  der Höchstbestand der kumulativen Ressource L1 überschritten wird. Deshalb wird der auslagernde Vorgang  $i := 2 \in \mathcal{E}$  gewählt und dessen spätestmöglicher Startzeitpunkt auf null gesetzt.
- (4) Vorgang  $i^* := 2$  wird zum Zeitpunkt  $S_2 := 0$  eingeplant. Es sind keine kumulativen Ressourcenkonflikte vorhanden. Das Verfahren terminiert.

□

Da es sich beim Beitrag von Schwindt, Fink u. a. (2007) um einen Konferenzbeitrag handelt, sind zahlreiche Verfahrensschritte nicht ausreichend detailliert erläutert, um das vorge-

schlagene Lösungsverfahren vollständig reproduzieren zu können. Beispielsweise ist nicht geklärt, nach welcher Logik die Vorgänge  $i \in \mathcal{E}$ , deren spätestmögliche Startzeitpunkte reduziert werden sollen, ausgewählt werden. Dasselbe gilt für die Auswahl des für den kumulativen Ressourcenkonflikt verantwortlichen Vorgangs  $i'$ . Außerdem wird nicht darauf eingegangen, welche Prioritätsregeln verwendet werden und ob kumulative Ressourcen darauf einen Einfluss haben.

Carlier, Moukrim und Xu (2009) stellen ein Lösungsverfahren vor, das auf *linearen Ordnungen* von ein- und auslagernden Ereignissen basiert. Eine lineare Ordnung ist eine Menge aufeinanderfolgender gerichteter Kanten, die die Ereignisse linear miteinander verbinden. Sind alle Ereignisse eines Projekts in einer linearen Ordnung enthalten, wird von einer vollständigen linearen Ordnung gesprochen. Ist eine solche vollständige lineare Ordnung gegeben, kann das Verfahren von Carlier, Moukrim und Xu (2009) in Polynomialzeit einen zulässigen Schedule berechnen bzw. die Unzulässigkeit zeigen. Dazu werden in einem vorgelagerten Schritt Zeitbeziehungen eingeführt, sodass kumulative Ressourcenkonflikte gar nicht erst auftreten. Die lineare Ordnung erfüllt dabei eine ähnliche Funktion wie eine Prioritätsregel. Sie gibt dem Verfahren auf eindeutige Weise vor, welche Ereignisse vorgezogen werden müssen, um Konflikte zu verhindern. Nicht für jede vollständige lineare Ordnung kann ein zulässiger Schedule bestimmt werden, sondern das Verfahren wird fehlschlagen, sobald durch zusätzliche Zeitbeziehungen Kreise positiver Länge auftreten. Hierin liegt der Nachteil des Verfahrens von Carlier, Moukrim und Xu (2009) im Vergleich zu Schwindt, Fink u. a. (2007): Da bei Schwindt, Fink u. a. (2007) Vorgänge in einer Schleife ein-, ausgeplant und zeitlich nach rechts verschoben werden können, kann es generell für eine größere Bandbreite an möglichen Prioritätsregeln einen zulässigen Schedule berechnen. Im Verfahren von Carlier, Moukrim und Xu (2009) muss hingegen eine zulässige, vollständige lineare Ordnung verwendet werden, was sich bei einer großen Anzahl kumulativer Ressourcen und Zeitbeziehungen als herausfordernd gestalten kann.

#### 5.2.4 Übertragbarkeit der Ansätze zur Lösung des (RCPSP/M)

Exakte Lösungsverfahren für das (RCPSP/c) bzw. das (RCPSP/M) stoßen aufgrund der NP-Schwere bei großen Problem instanzen an die Grenzen ihrer Leistungsfähigkeit. Im Branch-and-Bound-Verfahren von Neumann, Schwindt und Trautmann (2005) macht sich die Komplexität daran bemerkbar, dass die Anzahl der möglichen Alternativen für die Menge  $B$  exponentiell mit der Kardinalität der aktiven Menge  $\mathcal{A}^Y(S, t^c)$  wächst (Neumann und Schwindt 2002, S. 524). Deshalb konnten Neumann, Schwindt und Trautmann (2005, S. 506f) auch nur Instanzen mit bis zu 90 Vorgängen exakt lösen. Falls die kumulativen Ressourcen nur zum Start oder zum Ende eines Vorgangs genutzt werden, konnte ihr Verfahren auch Instanzen mit bis zu 750 Vorgängen lösen, allerdings nicht in jedem Fall optimal.

Die vorliegende Arbeit ist darauf ausgerichtet, mit dem (RCPSP/M) und einem dazu geeigneten Lösungsverfahren ein Optimierungswerkzeug zur Planung von Rückbauprojekten bereitzustellen. Für die Praxistauglichkeit eines solchen Optimierungswerkzeugs ist es von

zentraler Bedeutung, innerhalb kurzer Rechenzeiten für jede (nahezu) beliebige Problem-Instanz gute Projektpläne berechnen zu können. Dies gilt vor allem, falls verschiedene Szenarien für die Problemparameter verglichen werden sollen und nicht für jedes Szenario eine Rechenzeit von mehreren Stunden aufgebracht werden soll. Gleichzeitig ist zu erwarten, dass das Optimierungswerkzeug überwiegend für Großprojekte, wie kerntechnische Rückbauprojekte, eingesetzt werden wird. In solchen Großprojekten können die Problem-Instanzen sehr umfangreich werden – insbesondere bei Verwendung von granularen Stoffstromvorgängen. Für das Branch-and-Bound-Verfahren von Neumann, Schwindt und Trautmann (2005) ist nicht abzusehen, welche Rechenzeiten für die Berechnung einer zulässigen Lösung benötigt werden. Auch wenn im Vergleich zu den Rechenexperimenten in Schwindt und Trautmann (2000), Neumann und Schwindt (2002) und Neumann, Schwindt und Trautmann (2005) mit mittlerweile deutlich leistungsfähigeren Rechnern bessere Ergebnisse zu erwarten sind, beschränkt der exponentiell ansteigende Rechenaufwand weiterhin die handhabbare Instanzgröße. Deshalb wird entschieden, im weiteren Verlauf dieser Arbeit ein heuristisches Lösungsverfahren für das (RCPSM/M) zu entwickeln.

Als weiterer Grund für den Fokus auf ein heuristisches Lösungsverfahren kann angeführt werden, dass mit dem IBM ILOG CP Optimizer (kurz: CP Optimizer) bereits ein leistungsfähiges exaktes Lösungsverfahren als Referenz zur Verfügung steht. In Abschnitt 5.8 wird ausführlich erläutert, wie dieser Solver für Constraintprogramme zur Lösung des (RCPSM/M) eingesetzt werden kann. In Kapitel 6 erfolgt ein Vergleich zwischen dem heuristischen und exakten Lösungsansatz, welcher eine nachträgliche Validierung der gewählten Vorgehensweise ermöglicht.

Da es sich beim prioritätsregelbasierten Verfahren von Schwindt, Fink u. a. (2007) um ein heuristisches Lösungsverfahren handelt, erscheint es grundsätzlich dafür geeignet, auf das (RCPSM/M) übertragen zu werden. Aufgrund der geringen Tiefe kann das Verfahren jedoch, wie erläutert, nicht vollständig reproduziert werden, sondern es sind Präzisierungen erforderlich (vgl. Abschnitt 5.5). Außerdem sind problemspezifische Anpassungen sinnvoll, die die divergierende Struktur der Instanzen  $\mathcal{I}_{\text{RCPSM/M-gra}}$  und  $\mathcal{I}_{\text{RCPSM/M-agg}}$  ausnutzen, um das Verfahren effizienter zu gestalten (vgl. Abschnitt 5.3 und 5.4).

Die theoretischen Erkenntnisse von Carlier, Moukrim und Xu (2009) fließen in die Konzeption des Lösungsverfahrens ein und werden an den entsprechenden Stellen aufgegriffen. Wie erläutert, ist jedoch zu erwarten, dass das im Beitrag vorgeschlagene Lösungsverfahren aufgrund fehlender Ausplanungsschritte häufig fehlschlagen wird. Somit wird es nicht weiter in Betracht gezogen.

Die genannten Beiträge um Briskorn, Choi u. a. (2010) sind aufgrund der zahlreichen Spezialisierungen gegenüber dem (RCPSM/c) für diese Arbeit nur von geringer Relevanz. Falls allerdings ein Rückbauprojekt existieren sollte, dessen Rahmenbedingungen mit dem Modellumfang des Ein-Maschinen-Schedulingproblems übereinstimmen, können die spezialisierten Lösungsverfahren in Betracht gezogen werden. Es ist jedoch anzunehmen, dass die Anforderung, keine Arbeitsgänge parallel ausführen zu dürfen, in der Rückbaupraxis nicht zutrifft. Deshalb werden die Erkenntnisse von Briskorn, Choi u. a. (2010) und den



darauf aufbauenden Beiträgen bei der weiteren Entwicklung des Lösungsverfahrens nicht einbezogen.

## 5.3 Lösungsverfahren für granulare Instanzen des (RCPSP/M)

Nachfolgend wird ein Lösungsverfahren vorgestellt, das die speziellen Eigenschaften der mit dem granularen Transformationsschema erzeugten Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  ausnutzt. Dabei gilt es insbesondere zu beachten, dass die Granularität für reale Probleminstanzen eine sehr hohe Anzahl an Stoffstromvorgängen erwarten lässt.

### 5.3.1 Grundlegende Konzepte

Bei der Entwicklung eines Lösungsverfahrens für Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  sind die in Abschnitt 4.2.3 definierten Stoffstromstrukturen von zentraler Bedeutung. Kurz wiederholt besteht eine Stoffstromstruktur  $M_i := \{i\} \cup \overline{\text{Succ}^o(i)}$  aus einem Rückbauvorgang und seinen transitiven stoffstrominduzierten Nachfolgern. In Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  sind alle Stoffstromstrukturen disjunkt, da  $\overline{\text{Succ}^o(i)} \cap \overline{\text{Succ}^o(i')} = \emptyset$  für alle  $i, i' \in \{0, \dots, I+1\}$  mit  $i \neq i'$  gilt. Der von einer Stoffstromstruktur  $M_i$  induzierte Teilgraph des Vorgangsknotennetzplans  $N^{\text{temp}}$  ist ein Out-Tree, der nur an der Wurzel  $i$  verzweigt. In Bezug auf das zu planende Rückbauprojekt bedeutet dies, dass die Verarbeitung nach jeder einzelnen Stoffeinheit unterbrochen werden kann. Diese Modellierung erlaubt eine größtmögliche Flexibilität bei der Belegung der Ressourcen im Stoffstrombereich.

Die Stoffstromvorgänge in  $\overline{\text{Succ}^o(i)}$  sorgen im Modell dafür, dass alle von Rückbauvorgang  $i$  freigesetzten Stoffeinheiten zur Senke bewegt werden. Angenommen, eine Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  wird auf einen einzelnen Rückbauvorgang  $i$  und seine transitiven stoffstrominduzierten Nachfolger  $\overline{\text{Succ}^o(i)}$  reduziert, d. h., alle anderen Rückbau- und Stoffstromvorgänge werden gelöscht. Dann ist in jedem zulässigen Schedule  $(S_i, (S_j^o)_{j \in \overline{\text{Succ}^o(i)}})$  zu dieser reduzierten Instanz das kumulative Ressourcenprofil (= Lagerbestand) zum Zeitpunkt  $t = \max_{j \in \overline{\text{Succ}^o(i)}} S_j^o$  gleich null. Bei diesem Zeitpunkt handelt es sich um den spätesten Zeitpunkt, zu dem ein Stoffstromvorgang eine Stoffeinheit aus einer kumulativen Ressource auslagert. Ein zulässiger Schedule  $(S_i, (S_j^o)_{j \in \overline{\text{Succ}^o(i)}})$  kann bei Kenntnis der Stoffstromstrukturen einfach bestimmt werden, ohne die kumulativen Ressourcenrestriktionen im (RCPSP/M) explizit zu berücksichtigen: Jeder Zweig des durch  $M_i$  induzierten Out-Trees beschreibt eine *lineare Ordnung* für die Stoffstromvorgänge, die die Verarbeitung einer Stoffeinheit modellieren. Als lineare Ordnung definieren Carlier, Moukrim und Xu (2009) eine Menge zusammenhängender, gerichteter Kanten der Form  $\{(j_1, j_2), (j_2, j_3), (j_3, j_4), \dots\}$ . Wenn die Stoffstromvorgänge eines Zweigs entlang dieser linearen Ordnung eingeplant werden, wird die Stoffeinheit stets von einem granularen Stoffstromvorgang an den nächsten granularen Stoffstromvorgang übergeben und schnellstmöglich zur Senke geführt. Wenn die linearen Ordnungen für jeden Zweig des Out-Trees aneinandergelängt werden, ergibt sich eine vollständige lineare Ordnung für die reduzierte Probleminstanz. Nach

Carlier, Moukrim und Xu (2009, Abschnitte 4.2 und 4.3) kann hierfür in Polynomialzeit ein zulässiger Schedule berechnet werden oder gezeigt werden, dass die lineare Ordnung unzulässig ist. Die Erkenntnisse von Carlier, Moukrim und Xu (2009) sind hier anwendbar, da erneuerbare Ressourcen auch mit kumulativen Ressourcen modelliert werden können, da die granularen Stoffstromvorgänge die kumulativen Ressourcen nur zum Vorgangsstart und -ende nutzen, da Mindest- und Höchstbestände für kumulative Ressourcen ineinander transformierbar sind und da die ereignis- und die vorgangsbasierte Formulierung ineinander transformierbar sind. Im nachfolgenden Abschnitt wird anhand des Beispiels 5.5 veranschaulicht, wie für eine Stoffstromstruktur  $M_i$  eine vollständige lineare Ordnung erzeugt wird.

Durch überschneidungsfreie Aneinanderreihung zulässiger Schedules  $(S_i, (S_j^o)_{j \in \overline{\text{Succ}^o(i)}})$  für die auf die jeweiligen Stoffstromstrukturen  $M_i$  ( $i = 0, \dots, I + 1$ ) reduzierten Instanzen ergibt sich ein zulässiger Schedule  $(S, S^o)$  für die gesamte Instanz  $\mathcal{I}_{\text{RCPS}/M\text{-gra}}$ . Unter einer überschneidungsfreien Aneinanderreihung wird verstanden, dass jeweils der späteste Endzeitpunkt eines Stoffstromvorgangs in  $M_i$  dem Startzeitpunkt eines beliebigen anderen Rückbauvorgangs  $i' \neq i$  entspricht. Dies zeigt, dass für Instanzen  $\mathcal{I}_{\text{RCPS}/M\text{-gra}}$  aufgrund der stoffstrominduzierten Zeitstruktur effizienter Lösungen generiert werden können als für generelle Instanzen mit kumulativen Ressourcen.

Es ist zu erwarten, dass die überschneidungsfreie Aneinanderreihung zu Schedules mit schlechter Lösungsqualität, d. h. hohen Projektdauern, führt. Hinsichtlich der Zielsetzung zur Projektdauerminimierung ist es deshalb sinnvoll, Überschneidungen zuzulassen und Rückbau- und Stoffstromvorgänge immer so früh wie möglich einzuplanen. Dabei können jedoch kumulative Ressourcenkonflikte auftreten. Im nachfolgenden Abschnitt wird ein Generierungsschema vorgestellt, welches diesbezüglich einen Kompromiss darstellt: Es orientiert sich am skizzierten Konzept der linearen Ordnungen, um möglichst effizient zulässige Schedules zu erzeugen und enthält auch einen Mechanismus zum Umgang mit kumulativen Ressourcenkonflikten, die bei frühestmöglicher Einplanung von Rückbau- und Stoffstromvorgängen entstehen können.

### 5.3.2 Generierungsschema nach einem Dekompositionsansatz

Wie auch in der Literatur üblich (Kolisch 2015, S. 3), wird das in diesem Abschnitt entwickelte heuristische Lösungsverfahren im Kern aus einem Generierungsschema bestehen. Dabei wird die serielle Variante gewählt. Diese ermöglicht eine Anlehnung an Schwindt, Fink u. a. (2007) und fördert die Vergleichbarkeit. Außerdem ist die serielle Variante aufgrund der im (RCPS/M) vorliegenden Zweiteilung der Vorgangsmenge in eine Menge der Rückbau- und eine Menge der Stoffstromvorgänge besser geeignet: Das serielle Generierungsschema kann sich entlang der Stoffstromstruktur bewegen. Falls alternativ ein paralleles Generierungsschema gewählt würde, welches über Zeitpunkte iteriert, muss die Unterscheidung zwischen Rückbau- und Stoffstromvorgängen aufwendiger in die Prioritätsregel eingearbeitet werden. Weitere Gründe, die gegen die Verwendung eines parallelen Generierungsschemas sprechen, wurden bereits in Abschnitt 2.3.1 dargelegt.

Aufgrund der großen Menge an Stoffstromvorgängen, welche typischerweise in einer Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  zu erwarten sind, wird das nachfolgend vorgestellte Generierungsschema einen Schedule  $(S, S^o)$  auf zwei Ebenen erzeugen: Auf Ebene der Rückbauvorgänge und auf Ebene der Stoffstromstrukturen. Dieser *Dekompositionsansatz* bietet sich an, da die jeweiligen Stoffstromstrukturen disjunkt sind. Dadurch wird erreicht, dass bei der Einplanung jeder Stoffstromstruktur keine nachträgliche Veränderung der Startzeitpunkte anderer Stoffstromstrukturen erforderlich ist. Bildlich gesprochen, wird in jeder Iteration des Generierungsschemas eine Stoffstromstruktur an den Schedule der bestehenden Stoffstromstrukturen angehängt, ohne Ressourcenrestriktionen zu verletzen. Es handelt sich mit Blick auf die Zielsetzung der Projektdauerminimierung nicht wie oben beschrieben um eine überschneidungsfreie Aneinanderreihung, sondern um eine frühestmögliche Aneinanderreihung.

Es sei angemerkt, dass der Dekompositionsansatz für das Lösungsverfahren nicht mit der Bezeichnung des (RCPSP/M) in Verbindung steht. Das (RCPSP/M) wurde ausgeschrieben als „Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen“ bezeichnet. Das Attribut „integriert“ bezieht sich dabei auf die Modellierung: Die Stoffströme werden in das Rückbauprojekt integriert, indem sie als Stoffstromvorgänge modelliert werden. Bei dem nachfolgend vorgestellten Generierungsschema werden Rückbau- und Stoffstromvorgänge jedoch getrennt voneinander eingeplant, weshalb der Begriff des Dekompositionsansatzes gewählt wurde.

Die übergeordnete Ebene des Generierungsschemas wird als SuperSchedule bezeichnet und ist formal in Algorithmus 4 beschrieben. Die Zeilenreferenzen in diesem Absatz beziehen sich auf Algorithmus 4. Neben Instanz  $\mathcal{I}_{\text{RCPSP/M-gra}}$  wird für SuperSchedule eine Prioritätsregel  $\pi$  als Eingabe benötigt, wobei  $\pi(i)$  die Priorität jedes Rückbauvorgangs  $i \in V$  angibt. Initial sind die zu bestimmenden Startzeitpunkte unbekannt und werden deshalb auf die Dummybelegung von  $-\infty$  gesetzt (vgl. Zeile 1). Alle Rückbau- und Stoffstromvorgänge gelten damit als uneingeplant. Der fiktive Startvorgang muss zum Zeitpunkt 0 starten, weshalb  $S_0 := 0$  (vgl. Zeile 2). Anschließend wird solange eine Schleife durchlaufen, wie es noch uneingeplante Rückbauvorgänge gibt (vgl. Zeile 3). In Zeile 4 wird die Menge der einplanbaren Rückbauvorgänge  $\mathcal{E}$  als diejenige Menge der uneingeplanten Rückbauvorgänge bestimmt, deren direkte Vorgänger gemäß Gleichung (2.1) eingeplant sind. In Zeile 5 wird daraus der als nächstes einzuplanende Rückbauvorgang  $i^*$  anhand der übergebenen Prioritätsregel  $\pi$  ausgewählt. Für die zugehörige Stoffstromstruktur  $M_{i^*}$  wird in Zeile 6 ein weiterer Algorithmus aufgerufen, der als SubScheduleGra bezeichnet wird (Erläuterung siehe unten). SubScheduleGra berechnet einen Schedule  $S^M := (S_{i^*}, (S_j^o)_{j \in \overline{\text{Succ}^o(i^*)}})$  für  $M_{i^*}$ . Die Werte in  $S^M$  werden in den partiellen Schedule  $(S, S^o)$  übernommen (vgl. Zeile 7). Nach Abschluss der Schleife wird in Zeile 9 der Startzeitpunkt des fiktiven End-Stoffstromvorgangs  $(J+1)^o$  gesetzt. SuperSchedule terminiert und liefert einen zulässigen Schedule  $(S, S^o)$  für  $\mathcal{I}_{\text{RCPSP/M-gra}}$ . SuperSchedule kann ohne Weiteres so umgestaltet werden, dass die Eingabe anstelle einer Prioritätsregel  $\pi$  aus einer vorrangzulässigen Folge aller Rückbauvorgänge  $\ell = (0, i_1, i_2, \dots, i_I, I+1)$  besteht. Dann wird in der Schleife in Zeile 3 über die Rückbauvorgänge in der Folge iteriert. Die Zeilen 4

und 5 entfallen, da  $\ell$  vorgibt, welcher Rückbauvorgang jeweils als nächster einzuplanen ist.

---

**Algorithmus 4:** SuperSchedule

---

**Eingabe:** Instanz  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$ ; Prioritätsregel  $\pi$

- 1 initialisiere partiellen Schedule  $(S, S^o)$  mit  $S_i := -\infty$  für alle  $i \in V$  und  $S_j^o := -\infty$  für alle  $j \in V^o$ ;
- 2 setze  $S_0 := 0$ ;
- 3 **solange** Rückbauvorgänge  $i \in V$  mit  $S_i = -\infty$  verbleiben **tue**
- 4     bestimme  $\mathcal{E} := \{i \in V \mid S_i = -\infty \wedge S_{i'} \geq 0 \text{ für alle } i' \in \text{Pred}(i)\}$ ;
- 5     setze  $i^* := \min\{i \in \mathcal{E} \mid \pi(i) = \max_{i' \in \mathcal{E}} \pi(i')\}$ ;
- 6     bestimme  $S^M := \text{SubScheduleGra}(M_{i^*}, (S, S^o), \mathcal{I}_{\text{RCSP}/M\text{-gra}})$ ;
- 7     übernehme die Werte  $S^M$  in den partiellen Schedule  $(S, S^o)$ ;
- 8 **Ende**
- 9 setze  $S_{(J+1)^o}^o := \max_{j \in V^o} (S_j^o + d_j)$ ;

**Ausgabe:** zulässiger Schedule  $(S, S^o)$  für  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$

---

Falls in `SubScheduleGra` kumulative Ressourcenkonflikte auftreten, wäre es am einfachsten, direkt den Rückbauvorgang der entsprechenden Stoffstromstruktur zeitlich nach rechts zu verschieben, da es im partiellen Schedule  $(S, S^o)$  auf Ebene von `SuperSchedule` immer einen Zeitpunkt  $t = \max_{i=0, \dots, I+1} \max_{j \in \overline{\text{Succ}^o(i)}} S_j^o$  gibt, zu dem das kumulative Ressourcenprofil null ist (dadurch, dass alle Stoffeinheiten von den Stoffstromvorgängen zur Senke geführt werden). Dies würde allerdings der Zielsetzung widersprechen, das Rückbauprojekt so früh wie möglich zu beenden. Falls stattdessen Konflikte zunächst durch Verschiebung von Stoffstromvorgängen aufgelöst werden, können gegebenenfalls vorgeschaltete kumulative Ressourcen zur Pufferung genutzt werden, ohne das Rückbauprojekt zu verzögern. Stoffstromvorgänge liegen nämlich, wie in Abschnitt 4.3.2 festgestellt, nie auf dem kritischen Pfad. Dieses Kalkül wird bei der Ausgestaltung von `SubScheduleGra` berücksichtigt.

`SubScheduleGra` ist formal in Algorithmus 5 beschrieben. Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 5. Zur Veranschaulichung dient auch ein Flussdiagramm in Abbildung 5.4. Als Eingabe bekommt `SubScheduleGra` die Stoffstromstruktur  $M_i$ , die eingeplant werden soll, den partiellen Schedule  $(S, S^o)$  und die Instanz  $\mathcal{I}_{\text{RCSP}/M\text{-gra}}$  übergeben. Der Rückbauvorgang  $i$  in `SubScheduleGra` entspricht  $i^*$  in `SuperSchedule`. Um die Stoffstromvorgänge einzuplanen, muss ihre Reihenfolge festgelegt werden. Das heißt, es wird eine Prioritätsregel benötigt. Hierfür ist es naheliegend, die über die Stoffstromstrukturen  $M_i$  beschriebenen linearen Ordnungen als implizite Prioritätsregel zu verwenden. Das bedeutet, der vollständigen Verarbeitung jeder einzelnen Stoffeinheit, die von einem Rückbauvorgang freigesetzt wird, wird Priorität eingeräumt. Dadurch wird in einem Schedule dafür gesorgt, dass Stoffe möglichst schnell aus dem Stoffstrombereich entfernt werden. In Abschnitt 5.3.1 wurde für die vollständige lineare Ordnung aller Stoffstromvorgänge  $j \in \overline{\text{Succ}^o(i)}$  jedoch nicht festgelegt, nach welcher Reihenfolge die einzelnen über die Zweige beschriebenen linearen Ordnungen zusammengefügt werden. Es erscheint sinnvoll, die Zweige nach nichtfallenden Zeitabständen zu ihren jeweiligen Freisetzungsbeziehungen

zusammenzufügen. Das bedeutet, dass Stoffeinheiten, die früher von Rückbauvorgang  $i$  freigesetzt werden, prioritär verarbeitet werden. Zur Vereinfachung der Ausdrucksweise wird anstelle einer linearen Ordnung eine *Stoffstromvorgangsfolge*  $\mu := (j_1, \dots, j_{|\overline{Succ^o(i)}|})$  verwendet, die die Reihenfolge für die Einplanung von Stoffstromvorgängen beschreibt. Sie wird in Zeile 1 zusammengestellt, indem der von  $M_i$  induzierte Baum mittels einer Tiefensuche durchlaufen wird, wobei über kleinere Kantengewichte (= Zeitabstände) zu erreichende Knoten bevorzugt werden. Damit wird genau das Verhalten erzielt, welches zuvor argumentativ dargelegt wurde (d. h., die vollständige Verarbeitung einer Stoffeinheit hat Priorität und früher freigesetzte Stoffeinheiten haben Priorität). Beispiel 5.5 veranschaulicht dies.

**Beispiel 5.5 (Fortsetzung des Beispiels 4.2):** In Beispiel 4.2 wurde mit Hilfe des granularen Transformationsschemas eine Instanz  $\overline{I}_{\text{RCPSP/M-gra}}$  erzeugt, die aus Rückbauvorgang 1 mit  $\overline{Succ^o}(1) = \{1^o, \dots, 13^o\}$  besteht. Die stoffstrominduzierte Zeitstruktur ist in Abbildung 4.3 dargestellt. Für die zeitlichen Mindestabstände zu den Freisetzungsbeziehungen gilt  $d_{1,1^o}^{rel} = d_{1,8^o}^{rel} = d_{1,11^o}^{rel} = d_{1,13^o}^{rel} := 10$  und  $d_{1,5^o}^{rel} = 5$ . Mittels Tiefensuche wird  $\mu = (5^o, 6^o, 7^o, 1^o, \dots, 4^o, 8^o, \dots, 13^o)$  berechnet. Stoffstromvorgänge im Zweig beginnend mit  $5^o$  sind darin zuerst enthalten, da  $5^o$  aufgrund von  $d_{1,5^o}^{rel} = 5$  am frühesten starten kann. Anschließend folgen alle weiteren Zweige, wobei Stoffstromvorgänge mit kleinerer Nummer zuerst gewählt wurden.  $\square$

---

**Algorithmus 5:** SubScheduleGra

---

**Eingabe:** Stoffstromstruktur  $M_i$ ; partieller Schedule  $(S, S^o)$ ; Instanz  $\overline{I}_{\text{RCPSP/M-gra}}$

- 1 berechne  $\mu$  mittels Tiefensuche im von  $M_i$  induzierten Out-Tree;
- 2 **wiederhole**
- 3     setze  $S_i$  auf das frühestmögliche  $t \geq S_i$ , sodass (3.1) und (4.11) eingehalten sind;
- 4     durchlaufe  $\mu$  und setze  $S_j^o$  auf das frühestmögliche  $t \geq S_j^o$ , sodass (4.1), (4.2) und (4.11) eingehalten sind;
- 5     **wenn** bei einer Ressource  $k \in \mathcal{R}^Y$  ein Konflikt existiert **dann**
- 6         **für jedes**  $t^c$  mit  $f_k^Y(S, t^c) + r_k^Y(S^o, t^c) > R_k^Y$  **tue**
- 7             bilde Mengen  $B$  und  $A$ ;
- 8             setze  $S_{j'}^o := S_j^o - d_j$  für alle  $(j, j') \in (B \rightarrow A)$ ;
- 9             **Ende**
- 10         **wenn** der Konflikt für  $k$  nicht aufgelöst werden konnte **dann**
- 11             erhöhe  $S_i$  bis Ungleichungen (5.9) eingehalten sind;
- 12             setze  $S_j^o := -\infty$  für alle  $j \in \overline{Succ^o}(i)$ ;
- 13         **Ende**
- 14     **Ende**
- 15 **bis** partieller Schedule  $(S, S^o)$  zulässig ist;

**Ausgabe:** zulässiger Schedule  $S^M := (S_i, (S_j^o)_{j \in \overline{Succ^o}(i)})$  für  $M_i$

---

In Zeile 3 wird  $S_i$  auf den frühestmöglichen Zeitpunkt  $t \geq S_i$  gesetzt, bei dem die Vorrangrestriktionen (3.1) und die erneuerbaren Ressourcenrestriktionen (4.11) eingehalten sind.

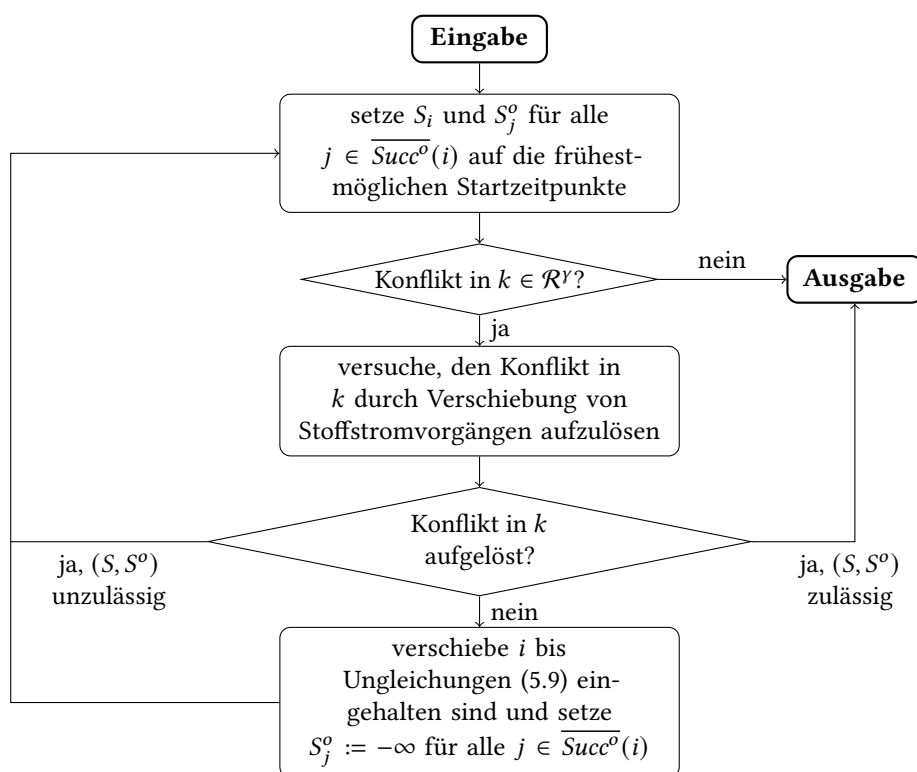


Abbildung 5.4: Flussdiagramm für SubScheduleGra.

Da keine zeitlichen Höchstabstände existieren, wird es hierfür immer einen zulässigen Zeitpunkt geben. Die Variable  $S_i$  kann bereits aus vorangegangenen Iterationen mit einem Wert belegt sein, weshalb in dieser Anweisung explizit  $t \geq S_i$  gefordert wird. Analog dazu wird mit den Stoffstromvorgängen verfahren. In Zeile 4 werden sie in der Reihenfolge der Stoffstromvorgangsfolge  $\mu$  durchlaufen. Für jeden Stoffstromvorgang  $j$  wird  $S_j^o$  auf den frühestmöglichen Zeitpunkt  $t \geq S_j^o$  gesetzt, bei dem die Freisetzungseinschränkungen (4.1), die stoffstrominduzierten Zeitrestriktionen (4.2) und die erneuerbaren Ressourcenrestriktionen (4.11) eingehalten sind. Auch hierfür wird es immer einen zulässigen Zeitpunkt geben und eine vorangegangene Belegung ist möglich.

Anschließend sind die kumulativen Ressourcenrestriktionen zu prüfen. Falls diese eingehalten werden, terminiert SubScheduleGra und gibt den Schedule  $S^M$  an SuperSchedule zurück. Ansonsten wird eine kumulative Ressource  $k \in \mathcal{R}^Y$  ausgewählt, bei der ein Konflikt vorliegt (vgl. Zeile 5). Falls hierfür mehrere kumulative Ressourcen zur Auswahl stehen, ist es sinnvoll, eine zu wählen, die im Flussnetzwerk in Senkennähe liegt (z. B. L5 in Abbildung 3.5). Dadurch werden die Konflikte „von unten nach oben“ aufgelöst und es ist jeweils sichergestellt, dass die nachfolgenden kumulativen Ressourcen alle Stoffe aufnehmen können. Wird stattdessen „von oben nach unten“ vorgegangen, ist es möglich, dass nach Auflösung von Konflikten wieder neue Konflikte in vorgeschalteten kumulativen Ressourcen auftreten, die bereits zuvor aufgelöst wurden. Dies würde die Anzahl der Iterationen erhöhen.

Das Auflösen eines Konflikts erfolgt in den Zeilen 6 bis 13. Wie erläutert, wird zunächst versucht, Konflikte durch Rechtsverschiebung von Stoffstromvorgängen aufzulösen. Dazu werden in Zeile 6 aufsteigend alle Zeitpunkte  $t^c$  durchlaufen, zu denen ein kumulativer Ressourcenkonflikt vorliegt. Für einen Zeitpunkt  $t^c$  wird eine Teilmenge  $B$  aller einlagernden Stoffstromvorgänge der Stoffstromstruktur gebildet, die bis einschließlich Zeitpunkt  $t^c$  enden, d. h.  $B \subseteq \{j \in \overline{\text{Succ}}^o(i) \mid r_{jk}^{y,end} = 1 \wedge 0 \leq S_j^o + d_j \leq t^c\}$ . Dabei soll  $B$  eine inklusionsmaximale Menge sein, sodass die Ungleichung

$$f_k^y(S, t^c) + r_k^y(S^o, t^c) - \sum_{j \in B} r_{jk}^{y,end} \geq R_k^y \quad (5.8)$$

eingehalten wird. Das bedeutet, es werden so viele Stoffstromvorgänge wie möglich zu  $B$  hinzugefügt, wodurch die Überschreitung der kumulativen Ressourcenkapazität so weit wie möglich reduziert wird. Es darf jedoch keine Unterschreitung der kumulativen Ressourcenkapazität erfolgen, da ansonsten unnötige Verschiebungen durchgeführt werden würden. Im besten Fall ist Ungleichung (5.8) mit Gleichheit erfüllt und der Konflikt konnte aufgelöst werden. Falls mehrere Möglichkeiten zur Bildung der Menge  $B$  bestehen, werden diejenigen Stoffstromvorgänge  $j$  gewählt, deren Endzeitpunkte  $(S_j^o + d_j)$  am größten sind.

Menge  $B$  umfasst in Anlehnung an Neumann, Schwindt und Trautmann (2005, vgl. Abschnitt 5.2.2 dieser Arbeit) die Menge derjenigen Stoffstromvorgänge, die zur Auflösung des Ressourcenkonflikts verschoben werden sollen. Diese Rechtsverschiebung muss dabei mindestens so groß sein, damit die Einlagerungen durch die Stoffstromvorgänge  $j \in B$  genau mit Auslagerungen anderer Stoffstromvorgänge  $j' \notin B$  zusammenfallen. Dazu wird eine Teilmenge  $A$  aller auslagernden Stoffstromvorgänge gebildet, deren Ausführung nach Zeitpunkt  $t^c$  startet, d. h.  $A \subset \{j' \in V^o \mid r_{jk}^{y,start} = -1 \wedge S_j^o > t^c\}$ . Dabei muss  $|A| = |B|$  gelten, damit jeder einlagernde Stoffstromvorgang in  $B$  einem einlagernden Stoffstromvorgang in  $A$  zugeordnet werden kann. Eine Erfüllung dieser Bedingung ist immer möglich, da jeder einlagernde Stoffstromvorgang einen auslagernden Stoffstromvorgang als Nachfolger hat. Davon abgesehen kommen für  $A$  auch alle anderen Stoffstromvorgänge infrage – insbesondere solche, die nicht Teil der Stoffstromstruktur sind. Zur Beschleunigung des Verfahrens können diejenigen Stoffstromvorgänge für die Bildung von  $A$  ausgeschlossen werden, nach deren Start sich das Ressourcenprofil von  $k$  nicht um mindestens eine Einheit verringert. Der Start eines solchen auslagernden Stoffstromvorgangs fällt dann bereits mit dem Ende eines anderen einlagernden Stoffstromvorgangs zusammen, sodass kein Beitrag zur Lösung des Konflikts zu erwarten ist. Falls mehrere Möglichkeiten zur Bildung der Menge  $A$  bestehen, werden diejenigen Stoffstromvorgänge  $j'$  gewählt, deren Startzeitpunkte  $S_{j'}^o$  am kleinsten sind.

Nach Bildung der beiden Mengen in Zeile 7 wird eine bijektive Relation  $B \rightarrow A$  erzeugt, die jedem Stoffstromvorgang  $j \in B$  einen Stoffstromvorgang  $j' \in A$  zuordnet. Für jede Zuordnung  $(j, j') \in (B \rightarrow A)$  wird in Zeile 8  $S_j^o + d_j := S_{j'}^o \Leftrightarrow S_{j'}^o := S_j^o - d_j$  gesetzt. Das bedeutet, das Ende des einlagernden Stoffstromvorgangs  $j \in B$  wird bis zum Start des auslagernden Stoffstromvorgangs  $j' \in A$  verschoben.

Falls Ungleichung (5.8) bei der Bestimmung von Menge  $B$  nicht mit Gleichheit erfüllt war, konnte der Konflikt nicht vollständig aufgelöst werden. Aufgrund der inklusionsmaxi-

malen Zusammensetzung von  $B$  wurde zwar versucht, möglichst viele Stoffeinheiten in vorgeschalteten kumulativen Ressourcen zu puffern. Dennoch muss nun Rückbauvorgang  $i$  zeitlich nach rechts verschoben werden, um den Konflikt aufzulösen. Dazu wird in Zeile 11  $S_i$  solange erhöht, bis die Ungleichungen

$$f_k^Y(S, t) + r_k^Y(S^o, t) \leq R_k^Y \quad \left( \forall t = S_i, \dots, \max_{j \in \overline{Succ}^o(i)} S_j^o \right) \quad (5.9)$$

erfüllt sind. Die neue Belegung von  $S_i$  ist derjenige Zeitpunkt, zu dem alle von  $i$  freigesetzten Stoffeinheiten in die kumulativen Ressourcen passen, sofern sich die zuvor bestimmten Startzeitpunkte der Stoffstromvorgänge  $j \in \overline{Succ}^o(i)$  nicht mehr ändern. Die zuvor bestimmten Startzeitpunkte können sich jedoch ändern, da nach der Verschiebung von  $i$  nicht mehr sichergestellt ist, dass alle Freisetzungs- und erneuerbaren Ressourcenrestriktionen eingehalten sind. Außerdem ist es möglich, dass zuvor wegen kumulativer Ressourcenkonflikte verschobene Stoffstromvorgänge nach der Verschiebung von  $i$  früher starten können. Deshalb werden in Zeile 12 die Startzeitpunkte aller Stoffstromvorgänge  $j \in \overline{Succ}^o(i)$  auf  $-\infty$  gesetzt, was einer Ausplanung entspricht. Die Abbruchbedingung in Zeile 15 greift nicht, da noch nicht alle Stoffstromvorgänge der Stoffstromstruktur eingeplant sind. Im anschließenden neuen Schleifendurchlauf wird dafür gesorgt, dass der verschobene Rückbauvorgang  $i$  hinsichtlich aller Restriktionen zulässig eingeplant wird.

SubScheduleGra terminiert, sobald der partielle Schedule  $(S, S^o)$  zulässig ist. Dazu ist anzumerken, dass für uneingeplante Rückbau- und Stoffstromvorgänge außerhalb der betrachteten Stoffstromstruktur, d. h. Rückbauvorgänge  $i' \neq i$  mit  $S_{i'} = -\infty$  und Stoffstromvorgänge  $j \notin \overline{Succ}^o(i)$  mit  $S_j^o = -\infty$ , alle zeitlichen Restriktionen des (RCPSM/M) als erfüllt angesehen werden. In Beispiel 5.6 wird der Ablauf des Generierungsschemas anhand einer exemplarischen Instanz erläutert.

**Beispiel 5.6:** Eine Instanz  $\mathcal{I}_{\text{RCPSM/M-gra}}$  besteht aus zwei realen Rückbauvorgängen 1 und 2, vier realen, granularen Stoffstromvorgängen  $1^o, \dots, 4^o$ , einer erneuerbaren Ressource R1 und zwei kumulativen Ressourcen L1 und L2. Für Rückbauvorgang 1 gilt  $d_1 = 4$  und  $\overline{Succ}^o(1) = \{1^o, 2^o\}$ . Für Rückbauvorgang 2 gilt  $d_2 = 3$  und  $\overline{Succ}^o(2) = \{3^o, 4^o\}$ . Abbildung 5.5 zeigt den Vorgangsknotennetzplan zu dieser Instanz, der die Informationen zu den Vorrangbeziehungen, den Freisetzungsbeziehungen und den stoffstrominduzierten Zeitbeziehungen mit den jeweiligen zeitlichen Mindestabständen (= Kantengewichte) bereitstellt. Rückbauvorgänge 0 und 3 sowie Stoffstromvorgang  $5^o$  sind fiktiv.

Für alle Stoffstromvorgänge  $j = 1^o, \dots, 4^o$  gilt  $d_j = 1$  und  $r_{j, R1}^\alpha = 1$ . Die Kapazität von R1 beträgt  $R_{R1}^\alpha = 1$ . Somit können die Stoffstromvorgänge nicht parallel ausgeführt werden. Rückbauvorgänge nutzen die erneuerbare Ressource R1 nicht. Die Rückbauvorgänge setzen jeweils eine Stoffeinheit zu ihrem Ende in L1 frei. Diese wird von den granularen Stoffstromvorgängen  $1^o$  bzw.  $3^o$  zu ihrem Start entnommen, verarbeitet und zu ihrem Ende in L2 eingelagert. Danach entnehmen  $2^o$  bzw.  $4^o$  die jeweilige Stoffeinheit zu ihrem Start und führen sie zur Senke. Die kumulativen Ressourcenkapazitäten betragen  $R_{L1}^Y = R_{L2}^Y = 0$ .

Angenommen, Prioritätsregel  $\pi$  besagt, dass Rückbauvorgang 1 vor Rückbauvorgang 2 eingeplant wird. Beim Aufruf von SubScheduleGra für  $M_1$  werden die frühestmöglichen



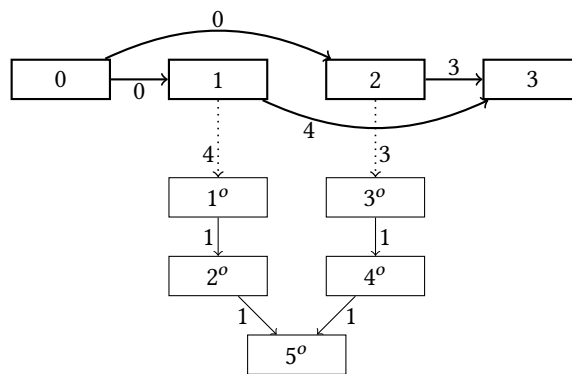


Abbildung 5.5: Vorgangsknotennetzplan zu Beispiel 5.6.

Startzeitpunkte  $S_1 := 0$ ,  $S_{1^\circ} := 4$  und  $S_{2^\circ} := 5$  bestimmt. Alle Stoffeinheiten werden hier direkt übergeben, sodass das Ressourcenprofil beider kumulativen Ressourcen bei null bleibt. SubScheduleGra terminiert für  $M_1$ .

Beim Aufruf von SubScheduleGra für  $M_2$  werden die Startzeitpunkte  $S_2 := 0$ ,  $S_{3^\circ} := 3$  und  $S_{4^\circ} := 6$  bestimmt. Stoffstromvorgang  $4^\circ$  kann aufgrund der erneuerbaren Ressourcenrestriktionen nicht früher als zum Zeitpunkt sechs starten. Abbildung 5.6a zeigt hierzu das Gantt-Diagramm zusammen mit den kumulativen Ressourcenprofilen. Es ist ersichtlich, dass zum Zeitpunkt  $t^c = 4$  ein kumulativer Ressourcenkonflikt vorliegt. Daher wird Menge  $B := \{3^\circ\}$  gebildet. Für Menge  $A$  kommen die auslagernden Stoffstromvorgänge  $2^\circ$  und  $4^\circ$  infrage. Nach Start von  $2^\circ$  verringert sich das Ressourcenprofil jedoch nicht, da  $2^\circ$  bereits die Stoffeinheit von  $4^\circ$  übernimmt. Somit kann  $2^\circ$  zur Beschleunigung des Verfahrens ausgeschlossen werden und  $A := \{4^\circ\}$  gebildet werden. Es wird  $S_{3^\circ} := S_{4^\circ} - d_{3^\circ} = 5$  gesetzt.

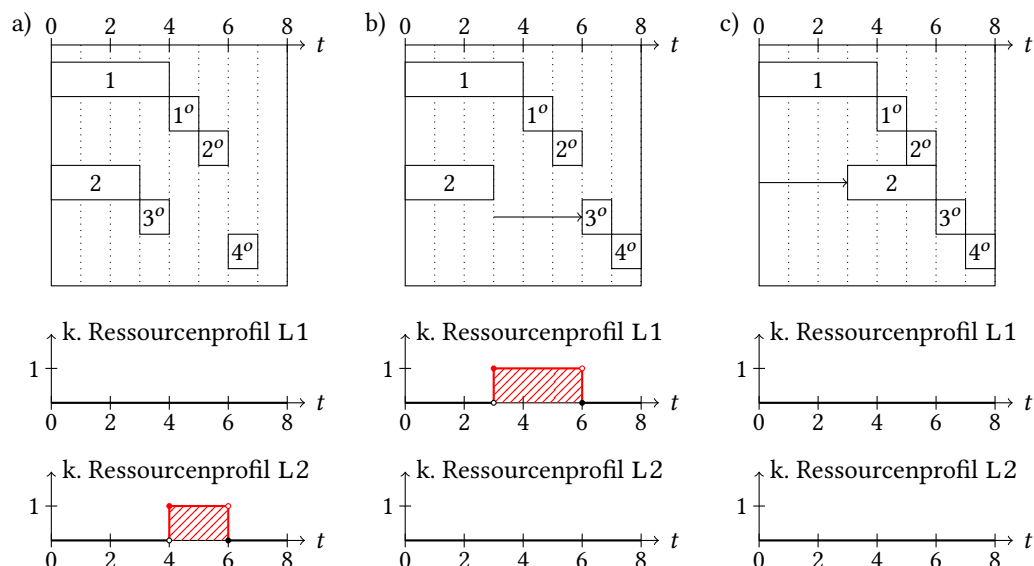


Abbildung 5.6: Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.6.

Mit der Rechtsverschiebung von  $3^o$  ist der Schedule  $(S, S^o)$  nicht zulässig, da  $S_{2^o}^o = S_{3^o}^o = 5$  und damit die erneuerbare Ressourcenkapazität überschritten wird. Somit ist für  $M_2$  eine zweite Iteration in SubScheduleGra erforderlich: Die Startzeitpunkte werden auf  $S_{3^o}^o := 6$  und  $S_{4^o}^o := 7$  erhöht, um die Zulässigkeit hinsichtlich R1 wiederherzustellen. Die kumulative Ressourcenkapazität von L2 ist jetzt eingehalten. Allerdings wurde der Konflikt in L1 verlagert, da die Stoffeinheit von Rückbauvorgang 2 nun dort gepuffert wird. Abbildung 5.6b zeigt hierzu das Gantt-Diagramm zusammen mit den kumulativen Ressourcenprofilen. Zum Zeitpunkt  $t^c = 3$  steht für die Menge  $B$  kein Stoffstromvorgang zur Auswahl, sodass die Ungleichung (5.8) nicht mit Gleichheit erfüllt und der Konflikt nicht aufgelöst werden kann. Daher muss Rückbauvorgang 2 gemäß Ungleichungen (5.9) auf  $S_2 := 3$  verschoben werden. Seine nachfolgenden Stoffstromvorgänge werden ausgeplant, d. h.  $S_{3^o}^o = S_{4^o}^o := -\infty$ .

In einer dritten Iteration für  $M_2$  werden  $S_{3^o}^o := 6$  und  $S_{4^o}^o := 7$  gesetzt. Damit ist der Schedule  $(S, S^o)$  zulässig und  $S^M$  bestimmt (vgl. Abbildung 5.6c). SubScheduleGra terminiert für  $M_2$ , woraufhin auch SuperSchedule nach Einplanung der fiktiven Endvorgänge terminiert.  $\square$

In jeder Iteration von SubScheduleGra wird mindestens ein Stoffstromvorgang  $j \in \overline{Succ^o}(i)$  oder der Rückbauvorgang  $i$  um mindestens eine Periode nach rechts verschoben. Dadurch wird SubScheduleGra grundsätzlich einen zulässigen Schedule  $S^M$  liefern. Es gibt jedoch zwei Fälle, die zu Beginn der Schleife in Zeile 2 auftreten können und in denen das Verfahren fehlschlägt:

1. Falls Rückbauvorgang  $i$  soweit nach rechts verschoben wurde, dass

$$S_i > \max \left\{ \max_{i' \in V \setminus \{i\}} (S_{i'} + d_{i'}), \max_{j \in V^o} (S_j^o + d_j) \right\}.$$

Dann befindet sich kein anderer Rückbau- oder Stoffstromvorgang außerhalb von  $M_i$  mehr in Ausführung. Es wird auch für spätere Zeitpunkte nicht möglich sein, die Stoffstromstruktur einzuplanen. Die Restriktionen binden so stark, dass für die Stoffstromstruktur mit der betrachteten Stoffstromvorgangsfolge  $\mu$  keine Lösung gefunden werden kann. Es ist möglich, dass mit einer anderen Stoffstromvorgangsfolge (z. B. mittels Breitensuche berechnet) eine zulässige Lösung gefunden werden kann. Falls dies auch nicht möglich ist, ist die Instanz unzulässig.

2. Falls ein Stoffstromvorgang  $j$  soweit nach rechts verschoben wurde, dass

$$S_j^o > \max \left\{ \max_{i \in V} (S_i + d_i), \max_{j' \in V^o \setminus \overline{Succ^o}(j)} (S_{j'}^o + d_{j'}) \right\}.$$

Dann befindet sich kein Rückbau- oder Stoffstromvorgang außer seinen transitiven Nachfolgern mehr in Ausführung. Es wird auch für spätere Zeitpunkte nicht möglich sein,  $j$  zulässig einzuplanen. Dann ist die Instanz unzulässig.

Diese Fälle sind aus Darstellungsgründen nicht explizit in Algorithmus 5 berücksichtigt.

Abschließend sei darauf hingewiesen, dass die Ausgestaltung von `SubScheduleGra` die optimale Lösung ausschließen kann. Dies liegt daran, dass lediglich die mittels Tiefensuche berechnete Stoffstromvorgangsfolge betrachtet wird. Außerdem führt die Zerlegung in die Ebene der Rückbauvorgänge und die Ebene der Stoffstromstrukturen dazu, dass der Lösungsraum nicht vollständig abgesucht werden kann. Inwiefern sich diese Schwäche praktisch bemerkbar macht, wird in Kapitel 6 experimentell untersucht. In Abschnitt 5.7 wird skizziert, welche Erweiterung erforderlich wäre, um den vollständigen Lösungsraum abzusuchen.

## 5.4 Lösungsverfahren für aggregierte Instanzen des (RCPSP/M)

Nachfolgend wird ein Lösungsverfahren vorgestellt, das die speziellen Eigenschaften der mit dem aggregierten Transformationsschema erzeugten Instanzen  $\mathcal{I}_{\text{RCPSP/M-agg}}$  ausnutzt. Für dieses Lösungsverfahren stehen zwei alternative Generierungsschemas zur Auswahl, die in den Abschnitten 5.4.2 und 5.4.3 hergeleitet werden.

### 5.4.1 Grundlegende Konzepte

Das Konzept der Stoffstromstrukturen gilt auch für Instanzen  $\mathcal{I}_{\text{RCPSP/M-agg}}$ , da auch hier alle transitiven Nachfolgermengen der Rückbauvorgänge disjunkt sind. Der von einer Stoffstromstruktur  $M_i := \{i\} \cup \overline{\text{Succ}^0(i)}$  induzierte Teilgraph des Vorgangsknotennetzplans  $N^{\text{temp}}$  ist ein Out-Tree mit beliebigen Verzweigungen (vgl. Abbildung 4.9). Die Stoffstromvorgänge in  $\overline{\text{Succ}^0(i)}$  sorgen dafür, dass alle von Rückbauvorgang  $i$  freigesetzten Stoffeinheiten zur Senke bewegt werden. Es ist analog zum granularen Fall möglich, eine Instanz auf eine einzelne Stoffstromstruktur zu reduzieren und hierzu einen Schedule  $(S_i, (S_j^o)_{j \in \overline{\text{Succ}^0(i)}})$  zu berechnen.

Die kumulativen Ressourcennutzungen der Stoffstromvorgänge in Instanzen  $\mathcal{I}_{\text{RCPSP/M-agg}}$  können ganzzahlige Werte annehmen und sind nicht wie im granularen Fall auf  $\{-1, 0, 1\}$  beschränkt. Daher sind Rückbau- und Stoffstromvorgänge im aggregierten Fall ähnlicher zueinander als im granularen Fall. Die Stofffreisetzung durch Rückbauvorgänge kann wie eine Einlagerung durch Stoffstromvorgänge behandelt werden. Dadurch entstehende Konflikte können auf dieselbe Weise aufgelöst werden. Zur Vereinfachung der Formulierungen wird daher eine Notation benötigt, die nicht zwischen Rückbau- und Stoffstromvorgängen unterscheidet. Hierfür wird im Folgenden Variable  $l$  verwendet, die sowohl ein Rückbau- als auch ein Stoffstromvorgang sein kann. Es wird allgemein von einem *Vorgang* gesprochen. Der Startzeitpunkt eines Vorgangs  $l$  wird als  $S_l$  bezeichnet. In dieser Notation wird auf das Superskript „o“ verzichtet, auch wenn  $l$  ein Stoffstromvorgang  $j \in V^o$  ist, dessen Startzeitpunkt dann strenggenommen als  $S_j^o$  bezeichnet werden müsste.

Alle kumulativen Ressourcennutzungen der Stoffstromvorgänge erfolgen in Instanzen  $\bar{I}_{\text{RCPSP}/M\text{-agg}}$  gleichmäßig über ihre Ausführung. Deshalb, und da die Ressourcennutzungen nicht auf  $\{-1, 0, 1\}$  beschränkt sind, ist es für aggregierte Instanzen generell aufwendiger, kumulative Ressourcenkonflikte aufzulösen. Ein Lösungsverfahren kann nicht mehr wie im granularen Fall gestaltet werden, sodass zunächst eine komplette Stoffstromstruktur eingeplant wird und nachträglich über den Mechanismus mit den Mengen  $B$  und  $A$  alle Konflikte aufgelöst werden. Stattdessen muss eine Stoffstromstruktur weiter unterteilt werden, um auf Basis dieser Unterteilung in einem Generierungsschema iterieren zu können. Dazu wird das Konzept der *relevanten Vorgänger bzw. Nachfolger* eingeführt. Der Begriff „relevant“ bezieht sich dabei auf die Relevanz hinsichtlich der Ein- und Auslagerungen einer bestimmten Anzahl an Stoffeinheiten. Bezeichne

$$\mathcal{R}^Y(l) := \begin{cases} \{k \in \mathcal{R}^Y \mid f_{lk}^Y(\cdot, \infty) > 0\} & \text{falls } l \in V, \\ \{k \in \mathcal{R}^Y \mid r_{lk}^Y(\cdot, \infty) > 0\} & \text{falls } l \in V^o, \end{cases} \quad (5.10)$$

die Menge aller kumulativen Ressourcen, in die ein Vorgang  $l \in V \cup V^o$  Stoffeinheiten freisetzt bzw. einlagert. Bei den relevanten Nachfolgern  $\text{Succ}^-(i)$  eines Rückbauvorgangs  $i \in V$  handelt es sich dann um alle Stoffstromvorgänge, die ausgeführt werden müssen, bis die gesamten von  $i$  freigesetzten Stoffeinheiten aus den kumulativen Ressourcen  $k \in \mathcal{R}^Y(i)$  ausgelagert sind, in die sie von  $i$  freigesetzt wurden. Analog dazu handelt es sich bei den relevanten Nachfolgern  $\text{Succ}^-(j)$  eines Stoffstromvorgangs  $j \in V^o$  um alle Stoffstromvorgänge, die ausgeführt werden müssen, bis die gesamten von  $j$  eingelagerten Stoffeinheiten aus den kumulativen Ressourcen  $k \in \mathcal{R}^Y(j)$  ausgelagert sind, in die sie von  $j$  eingelagert wurden. Das hochgestellte Minussymbol deutet darauf hin, dass relevante Nachfolger hinsichtlich ihrer Auslagerungen als relevant gelten. Falls ein Rückbauvorgang  $i \in V$  keine Stoffe freisetzt, ist  $\mathcal{R}^Y(i) = \emptyset$  und daher auch  $\text{Succ}^-(i) = \emptyset$ . Falls ein Stoffstromvorgang  $j \in V^o$  hinsichtlich keiner kumulativen Ressource als einlagernd gilt, ist  $\mathcal{R}^Y(j) = \emptyset$  und daher auch  $\text{Succ}^-(j) = \emptyset$ . Generell gilt in Instanzen  $\bar{I}_{\text{RCPSP}/M\text{-agg}}$  für alle  $l \in V \cup V^o$ , dass  $\text{Succ}^o(l) \subseteq \text{Succ}^-(l) \subseteq \overline{\text{Succ}^o(l)}$ .

Zur Bestimmung der relevanten Nachfolger eines Vorgangs  $l$  in einer Instanz  $\bar{I}_{\text{RCPSP}/M\text{-agg}}$  wird der Out-Tree für jede kumulative Ressource  $k \in \mathcal{R}^Y(l)$  ausgehend von  $l$  rekursiv durchlaufen. Die rekursiv aufgerufene Funktion ist in Algorithmus 6 beschrieben und wird als `RelevanteNachfolgerAgg` bezeichnet. Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 6. Als Eingabe benötigt `RelevanteNachfolgerAgg` die Instanz  $\bar{I}_{\text{RCPSP}/M\text{-agg}}$ , den aktuell betrachteten Vorgang  $l$ , die aktuell betrachtete kumulative Ressource  $k$ , die Menge der relevanten Nachfolger  $\text{Succ}^-(l)$  und eine Hilfsmenge  $NEUTR$ . Sollen für einen bestimmten Vorgang  $l \in V \cup V^o$  die relevanten Nachfolger berechnet werden, wird zunächst  $\text{Succ}^-(l) := \emptyset$  initialisiert. Danach wird die Rekursion für jede kumulative Ressource  $k \in \mathcal{R}^Y(l)$  mit dem Aufruf  $\text{Succ}^-(l) := \text{RelevanteNachfolgerAgg}(\bar{I}_{\text{RCPSP}/M\text{-agg}}, l, k, \text{Succ}^-(l), \emptyset)$  gestartet. Innerhalb von `RelevanteNachfolgerAgg` werden die direkten Nachfolger  $l' \in \text{Succ}^o(l)$  durchlaufen, solange  $l' \neq (J+1)^o$  (vgl. Zeile 1). Für jeden direkten Nachfolger  $l'$  erfolgt eine Fallunterscheidung: Falls  $l'$  aus  $k$  auslagert (vgl. Zeile 2), ist er ein relevanter Nachfolger und wird zusammen mit  $NEUTR$  zu  $\text{Succ}^-(l)$  hinzugefügt (vgl. Zeile 3). Die Hilfsmenge  $NEUTR$  wird gelöscht (vgl. Zeile 4) und der Out-Tree wird über einen rekursiven Aufruf von `RelevanteNachfolgerAgg` weiter durchsucht (vgl. Zeile 5).

Falls  $l'$  hinsichtlich aller kumulativer Ressourcen als neutral gilt (vgl. Zeile 6), wird er zu  $NEUTR$  hinzugefügt (vgl. Zeile 7) und der Out-Tree wird über einen rekursiven Aufruf von  $RelevanteNachfolgerAgg$  weiter durchsucht (vgl. Zeile 8). Falls keiner der beiden oben genannten Fälle vorliegt (vgl. Zeile 9), d. h.,  $l'$  lagert nicht aus  $k$ , sondern aus mindestens einer anderen kumulativen Ressource aus, ist er kein relevanter Nachfolger von  $l$  mehr. Das Durchlaufen des Out-Trees wird beendet und die Hilfsmenge  $NEUTR$  wird gelöscht (vgl. Zeile 10). Die Hilfsmenge  $NEUTR$  dient dazu, die beim Durchlaufen des Out-Trees angetroffenen neutralen Stoffstromvorgänge solange mitzuführen, bis Aufschluss darüber besteht, ob sie sich auf einem Pfad zu einem relevanten oder zu einem nichtrelevanten Nachfolger befinden. Abhängig davon werden sie zu  $Succ^-(l)$  hinzugefügt oder nicht. Beispiel 5.7 veranschaulicht die Funktionsweise von  $RelevanteNachfolgerAgg$ .

---

**Algorithmus 6:**  $RelevanteNachfolgerAgg$

---

**Eingabe:** Instanz  $\mathcal{I}_{RC)PSP/M-agg}$ ; Vorgang  $l$ ; kumulative Ressource  $k$ ; relevante Nachfolger  $Succ^-(l)$ ; Menge  $NEUTR$

1 **für jeden** direkten Nachfolger  $l' \in Succ^o(l)$  mit  $l' \neq (J+1)^o$  **tue**

2     **wenn**  $r_{l'k}^y(\cdot, \infty) < 0$  **dann**

3         setze  $Succ^-(l) := Succ^-(l) \cup NEUTR \cup \{l'\}$ ;

4         setze  $NEUTR := \emptyset$ ;

5         berechne

$Succ^-(l) := RelevanteNachfolgerAgg(\mathcal{I}_{RC)PSP/M-agg}, l', k, Succ^-(l), NEUTR)$ ;

6     **sonst wenn**  $r_{l'k}^y(\cdot, \infty) = 0$  für alle  $k \in \mathcal{R}^y$  **dann**

7         setze  $NEUTR := NEUTR \cup \{l'\}$ ;

8         berechne

$Succ^-(l) := RelevanteNachfolgerAgg(\mathcal{I}_{RC)PSP/M-agg}, l', k, Succ^-(l), NEUTR)$ ;

9     **sonst**

10         setze  $NEUTR := \emptyset$ ;

11     **Ende**

12 **Ende**

**Ausgabe:** relevante Nachfolger  $Succ^-(l)$

---

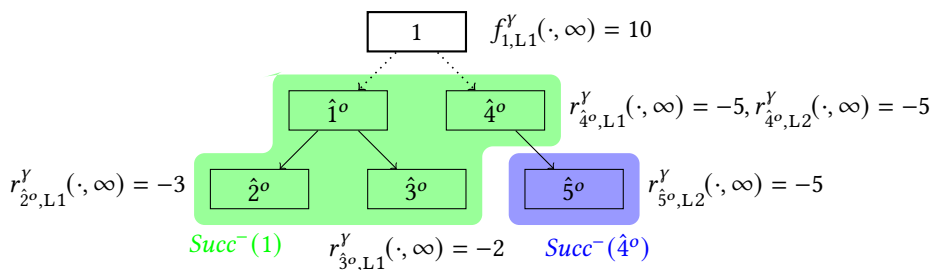
**Beispiel 5.7:** Eine Instanz  $\mathcal{I}_{RC)PSP/M-agg}$  besteht aus einem realen Rückbauvorgang 1, fünf realen Stoffstromvorgängen  $\hat{1}^o, \dots, \hat{5}^o$  und zwei kumulativen Ressourcen L1 und L2. Für Rückbauvorgang 1 gilt  $Succ^o(1) = \{\hat{1}^o, \dots, \hat{5}^o\}$ . Abbildung 5.7 zeigt den Vorgangsknoten-netzplan zu dieser Instanz. Der fiktive Stoffstromvorgang  $\hat{6}^o$  ist aus Darstellungsgründen nicht eingezeichnet. Die gesamten Stofffreisetzungen und kumulativen Ressourcennutzungen der jeweiligen Vorgänge sind ebenfalls in Abbildung 5.7 angegeben, sofern sie nicht null sind. Es ist ersichtlich, dass Stoffstromvorgang  $\hat{1}^o$  bezüglich aller kumulativer Ressourcen neutral ist.

Um für Rückbauvorgang 1 die relevanten Nachfolger zu berechnen, wird  $Succ^-(1) := \emptyset$  initialisiert. Da  $\mathcal{R}^y(1) = \{L1\}$ , wird anschließend die rekursive Berechnung mit  $Succ^-(1) := RelevanteNachfolgerAgg(\mathcal{I}_{RC)PSP/M-agg}, 1, L1, \emptyset, \emptyset)$  gestartet. Es gilt  $Succ^o(1) = \{\hat{1}^o, \hat{4}^o\}$ . Daher wird zunächst  $l' := \hat{1}^o$  gesetzt. Da  $r_{\hat{1}^o, L1}^y(\cdot, \infty) = 0$  für alle  $k \in \mathcal{R}^y$ , tritt der in

Zeile 6 beschriebene Fall ein und  $\hat{1}^o$  wird zu  $NEUTR$  hinzugefügt. Es folgt nun der rekursive Aufruf  $Succ^{-}(1) := \text{RelevanteNachfolgerAgg}(\mathcal{I}_{\text{RCPSP/M-agg}}, \hat{1}^o, L1, \emptyset, \{\hat{1}^o\})$ . Es gilt  $Succ(\hat{1}^o) = \{\hat{2}^o, \hat{3}^o\}$ . Daher wird  $l' := \hat{2}^o$  gesetzt. Da  $r_{\hat{2}^o, L1}^Y(\cdot, \infty) = -3$ , tritt der in Zeile 2 beschriebene Fall ein: Die Menge  $NEUTR$  sowie  $\hat{2}^o$  werden zu  $Succ^{-}(1)$  hinzugefügt, d. h.  $Succ^{-}(1) := \{\hat{1}^o, \hat{2}^o\}$ . Der anschließende rekursive Aufruf von  $\text{RelevanteNachfolgerAgg}$  beendet sofort, da der fiktive Stoffstromvorgang erreicht wird. Dann wird  $l' := \hat{3}^o$  gesetzt. Da  $r_{\hat{3}^o, L1}^Y(\cdot, \infty) = -2$ , tritt der in Zeile 2 beschriebene Fall ein: Die (hier leere) Menge  $NEUTR$  sowie  $\hat{3}^o$  werden zu  $Succ^{-}(1)$  hinzugefügt, d. h.  $Succ^{-}(1) := \{\hat{1}^o, \hat{2}^o, \hat{3}^o\}$ . Der anschließende rekursive Aufruf von  $\text{RelevanteNachfolgerAgg}$  beendet ebenfalls sofort, da der fiktive Stoffstromvorgang erreicht wird. Im Stapel der rekursiven Aufrufe wird zum ersten Aufruf zurückgegangen und  $l' := \hat{4}^o$  gesetzt. Da  $r_{\hat{4}^o, L1}^Y(\cdot, \infty) = -5$ , tritt der in Zeile 2 beschriebene Fall ein: Die (hier leere) Menge  $NEUTR$  sowie  $\hat{4}^o$  werden zu  $Succ^{-}(1)$  hinzugefügt, d. h.  $Succ^{-}(1) := \{\hat{1}^o, \hat{2}^o, \hat{3}^o, \hat{4}^o\}$ . Der anschließende rekursive Aufruf von  $\text{RelevanteNachfolgerAgg}$  beendet, ohne  $Succ^{-}(1)$  zu verändern, da bei  $\hat{5}^o$  der in Zeile 9 beschriebene Fall eintritt: Stoffstromvorgang  $\hat{5}^o$  ist kein relevanter Nachfolger von Rückbauvorgang 1 mehr, da er nicht aus L1 auslagert, sondern aus L2.

Für die Vorgänge  $l = \hat{1}^o, \hat{2}^o, \hat{3}^o, \hat{5}^o$  gilt jeweils  $\mathcal{R}^Y(l) = \emptyset$ , d. h., die Rekursion wird nicht aufgerufen und damit bleibt  $Succ^{-}(l) = \emptyset$ . Um für Stoffstromvorgang  $\hat{4}^o$  die relevanten Nachfolger zu berechnen, wird  $Succ^{-}(\hat{4}^o) := \emptyset$  initialisiert. Da  $\mathcal{R}^Y(\hat{4}^o) = \{L2\}$ , wird anschließend die rekursive Berechnung mit  $Succ^{-}(\hat{4}^o) := \text{RelevanteNachfolgerAgg}(\mathcal{I}_{\text{RCPSP/M-agg}}, \hat{4}^o, L2, \emptyset, \emptyset)$  gestartet. Es gilt  $Succ^o(\hat{4}^o) = \{\hat{5}^o\}$ . Daher wird  $l' := \hat{5}^o$  gesetzt. Da  $r_{\hat{5}^o, L2}^Y(\cdot, \infty) = -5$ , tritt der in Zeile 2 beschriebene Fall ein: Die (hier leere) Menge  $NEUTR$  sowie  $\hat{5}^o$  werden zu  $Succ^{-}(\hat{4}^o)$  hinzugefügt, d. h.  $Succ^{-}(\hat{4}^o) := \{\hat{5}^o\}$ . Der anschließende rekursive Aufruf von  $\text{RelevanteNachfolgerAgg}$  beendet sofort, da der fiktive Stoffstromvorgang erreicht wird.

Die relevanten Nachfolgermengen sind in Abbildung 5.7 farblich hinterlegt. Zusammengefasst hat Rückbauvorgang 1 die relevanten Nachfolger  $Succ^{-}(1) := \{\hat{1}^o, \hat{2}^o, \hat{3}^o, \hat{4}^o\}$  und Stoffstromvorgang  $\hat{5}^o$  die relevanten Nachfolger  $Succ^{-}(\hat{4}^o) := \{\hat{5}^o\}$ .  $\square$



**Abbildung 5.7:** Vorgangsknotennetzplan zu Beispiel 5.7.

Für jeden relevanten Nachfolger  $l' \in Succ^{-}(l)$  eines Vorgangs  $l \in V \cup V^o$  wird umgekehrt definiert, dass  $l$  sein *relevanter Vorgänger*  $l^+$  ist. Das hochgestellte Plusymbol deutet darauf hin, dass der relevante Vorgänger hinsichtlich seiner Einlagerungen als relevant gilt.

Rückbauvorgänge haben keine relevanten Vorgänger. Wenn mit `RelevanteNachfolgerAgg` für alle  $l \in V \cup V^o$  in einer Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$  die relevanten Nachfolger berechnet werden, wird die Menge der Stoffstromvorgänge  $V^o$  dadurch vollständig in disjunkte Teilmengen  $\text{Succ}^-(l)$  zerlegt. Es ist somit nicht möglich, dass ein Stoffstromvorgang zwei- oder mehrmals in einer Menge der relevanten Nachfolger enthalten ist. Das heißt, jeder reale Stoffstromvorgang hat einen eindeutigen relevanten Vorgänger. Dies ist auch in Beispiel 5.7 ersichtlich.

Gemäß obiger Definition der relevanten Nachfolger gilt, dass alle durch einen Vorgang  $l$  freigesetzten oder eingelagerten Stoffeinheiten zum Zeitpunkt  $t = \max_{l' \in \text{Succ}^-(l)} (S_{l'} + d_{l'})$  vollständig ausgelagert sind. Dies wird nur dadurch ermöglicht, dass in jedem Zweig in der Baumstruktur der Instanzen  $\mathcal{I}_{\text{RCPSP/M-agg}}$  die Flusserhaltung gewährleistet ist. Eingelegerte Stoffeinheiten werden immer durch nachfolgende Stoffstromvorgänge ausgelagert. Zur Planung der in eine kumulative Ressource ein- und ausgehenden Stoffströme ist es daher ausreichend, die stoffstrominduzierte Zeitstruktur zu durchlaufen, statt explizit die kumulativen Ressourcennutzungen (= Ein- und Auslagerungen) zu analysieren. Das Durchlaufen der stoffstrominduzierten Zeitstruktur erfolgt rekursiv in `RelevanteNachfolgerAgg`. Somit dient das Konzept der relevanten Nachfolger dazu, die in die Instanzen  $\mathcal{I}_{\text{RCPSP/M-agg}}$  eingebetteten Stoffströme zu reproduzieren. Dadurch lassen sich Einlagerungen und Auslagerungen einander eindeutig zuordnen, was Voraussetzung für die Auflösung kumulativer Ressourcenkonflikte in einem Lösungsverfahren ist. Für Instanzen  $\mathcal{I}_{\text{RCPSP/M}}$ , die nicht mit dem aggregierten Transformationsschema erzeugt wurden, ist die Zuordnung von Ein- und Auslagerungen basierend auf der stoffstrominduzierten Zeitstruktur nicht möglich, wie in Abschnitt 5.5 gezeigt werden wird.

Angenommen, für einen Rückbauvorgang  $i \in V$  starten alle Stoffstromvorgänge  $j' \in \text{Succ}^-(i)$  zu ihren frühestmöglichen, hinsichtlich der Freisetzungs- und stoffstrominduzierten Zeitbeziehungen zulässigen Startzeitpunkten. Dann werden alle von  $i$  freigesetzten Stoffeinheiten so früh wie möglich wieder ausgelagert. Das heißt, das sich daraus ergebende kumulative Ressourcenprofil beschreibt eine untere Schranke für das tatsächlich in einem Schedule beobachtete kumulative Ressourcenprofil aller kumulativen Ressourcen  $k \in \mathcal{R}^y(i)$ . Diese untere Schranke wird als *zeitzulässige Mindestauslastung* bezeichnet und bei der nachfolgenden Entwicklung eines Lösungsverfahrens einbezogen. Formal berechnet sich die zeitzulässige Mindestauslastung einer kumulativen Ressource  $k \in \mathcal{R}^y(i)$  für einen Rückbauvorgang  $i \in V$  zu einem Zeitpunkt  $t \geq S_i$  zu

$$\underbrace{f_{ik}^y(S_i, t)}_{\text{I}} + \underbrace{\sum_{j' \in \text{Succ}^-(i) | S_i + d_{ij'} \leq t} r_{j'k}^y(S_i + d_{ij'}, t)}_{\text{II}}. \quad (5.11)$$

Summand I umfasst die Stofffreisetzungen von Rückbauvorgang  $i$  zum Zeitpunkt  $t$ . Summand II umfasst die Auslagerung durch die relevanten Nachfolger von  $i$  zum Zeitpunkt  $t$ . Dabei bezeichnet  $d_{ij'}$  die Länge des Pfades von  $i$  zu  $j'$  im von  $M_i$  induzierten Out-Tree. Beim Zeitpunkt  $(S_i + d_{ij'})$  handelt es sich somit um den frühestmöglichen, hinsichtlich der Freisetzungs- und stoffstrominduzierten Zeitbeziehungen zulässigen Startzeitpunkt für  $j'$ .

Im Summenausdruck von Summand II wird nur über diejenigen relevanten Nachfolger  $j'$  von  $i$  summiert, die zum Zeitpunkt  $t$  frühestmöglich gestartet sein können, d. h.  $S_i + d_{ij'} \leq t$ .

Für einen nichtneutralen Stoffstromvorgang  $j \in V^o$  und seine relevanten Nachfolger  $j' \in Succ^-(j)$  berechnet sich die zeitzulässige Mindestauslastung einer kumulativen Ressource  $k \in \mathcal{R}^Y(j)$  zu einem Zeitpunkt  $t \geq S_j^o$  analog zu

$$r_{jk}^Y(S_j^o, t) + \sum_{j' \in Succ^-(j) | S_j^o + d_{jj'} \leq t} r_{j'k}^Y(S_j^o + d_{jj'}, t). \quad (5.12)$$

### 5.4.2 Generierungsschema nach einem Dekompositionsansatz

In diesem Abschnitt wird vergleichbar mit dem in Abschnitt 5.3.2 betrachteten granularen Fall ein serielles Generierungsschema nach einem Dekompositionsansatz vorgestellt. Das bedeutet, die Erzeugung eines Schedules  $(S, S^o)$  zu einer Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$  wird auf eine übergeordnete Ebene der Rückbauvorgänge und auf eine untergeordnete Ebene der Stoffstromstrukturen aufgeteilt. Auf der Ebene der Rückbauvorgänge kommt nahezu unverändert das in Algorithmus 4 beschriebene Verfahren SuperSchedule zum Einsatz. Dabei ist lediglich die Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  durch  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$  zu ersetzen. Außerdem muss für jeden einzuplanenden Rückbauvorgang  $i^*$  anstelle von SubScheduleGra das in Algorithmus 7 beschriebene Verfahren SubScheduleAgg aufgerufen werden. Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 7.

In den Zeilen 1 und 2 werden die relevanten Nachfolger für den Rückbauvorgang  $i$  und alle Stoffstromvorgänge der Stoffstromstruktur  $M_i$  berechnet. Anschließend ist die Reihenfolge festzulegen, nach der die Stoffstromvorgänge  $j \in Succ^o(i)$  eingeplant werden. Diese Reihenfolge wird über die Stoffstromvorgangsfolge  $\mu$  beschrieben (vgl. Abschnitt 5.3.2) und in Zeile 3 berechnet. Dabei bezeichnet  $()$  eine leere Folge. Analog zum granularen Fall sollen Stoffeinheiten so schnell wie möglich zur Senke geführt werden, um in den kumulativen Ressourcen Platz für nachrückende Stoffeinheiten zu schaffen. Im hier betrachteten aggregierten Fall bedeutet dies, dass die relevanten Nachfolger eines Vorgangs stets prioritär eingeplant werden. Dies lässt sich umsetzen, indem  $\mu$  rekursiv durch Hinzufügen der relevanten Nachfolger zusammengestellt wird, wie in Algorithmus 8 beschrieben. Die Notation  $Succ^-$  fasst dabei alle Mengen der relevanten Nachfolger zusammen, d. h.  $Succ^- := (Succ^-(l))_{l \in V \cup V^o}$ . Für die in Beispiel 5.7 beschriebene Instanz ergibt sich  $\mu = (\hat{1}^o, \hat{2}^o, \hat{3}^o, \hat{4}^o, \hat{5}^o)$ , falls davon ausgegangen wird, dass alle Kantengewichte gleich groß sind.

In Zeile 4 werden Hilfsvariablen  $ES_l$  zur Zwischenspeicherung der frühestmöglichen, zur Auflösung eines vorangegangenen Konflikts bestimmten Startzeitpunkte initialisiert und auf  $-\infty$  gesetzt. Diese Hilfsvariablen verringern die Zahl der Iterationen, wie in einem Beispiel am Ende dieses Abschnitts gezeigt werden wird. Der in jeder Iteration von SubScheduleAgg einzuplanende Rückbau- oder Stoffstromvorgang wird als  $l^*$  bezeichnet. In Zeile 5 wird  $l^*$  auf den Rückbauvorgang  $i$  gesetzt, dessen Stoffstromstruktur eingeplant werden soll. Der Rückbauvorgang  $i$  in SubScheduleAgg entspricht  $i^*$  in SuperSchedule.



---

**Algorithmus 7:** SubScheduleAgg

---

**Eingabe:** Stoffstromstruktur  $M_i$ ; partieller Schedule  $(S, S^o)$ ; Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$

- 1 initialisiere  $Succ^-(l) := \emptyset$  für alle  $l \in M_i$ ;
- 2 berechne  $Succ^-(l) = \text{RelevanteNachfolgerAgg}(\mathcal{I}_{\text{RCPSP/M-agg}}, l, k, Succ^-(l), \emptyset)$  für alle  $l \in M_i$  und alle  $k \in \mathcal{R}^Y(l)$ ;
- 3 berechne  $\mu := \text{StoffstromvorgangsfolgeAgg}(\cdot, i, Succ^-)$ ;
- 4 initialisiere  $ES_l := -\infty$  für alle  $l \in M_i$ ;
- 5 setze  $l^* := i$ ;
- 6 **wiederhole**
  - 7 setze  $S_{l^*}$  auf das frühestmögliche  $t \geq ES_{l^*}$ , sodass (3.1), (4.1), (4.2), (4.11) eingehalten sind und Kapazität für zeitzulässige Mindestauslastung existiert;
  - 8 lese den relevanten Vorgänger  $l^+$  von  $l^*$  aus, wenn möglich;
  - 9 **wenn**  $l^+$  *ausgelesen werden konnte* **und**  $S_{j'}^o \geq 0$  für alle  $j' \in Succ^-(l^+)$  **und** Restriktionen (5.13) verletzt werden **dann**
    - 10 erhöhe  $S_{l^+}$  bis Restriktionen (5.13) eingehalten sind;
    - 11 setze  $S_l := -\infty$  für alle  $l \in Succ^-(l^+)$ ;
    - 12 setze  $ES_{l^+} := S_{l^+}$  und  $S_{l^+} := -\infty$ ;
  - 13 **Ende**
  - 14 **wenn**  $S_i = -\infty$  **dann**
    - 15 setze  $l^* := i$ ;
  - 16 **sonst wenn** ein  $j$  in  $\mu$  mit  $S_j^o = -\infty$  existiert **dann**
    - 17 setze  $l^*$  auf den ersten Stoffstromvorgang  $j$  in  $\mu$  mit  $S_j^o = -\infty$ ;
  - 18 **Ende**
- 19 **bis**  $S_j^o \geq 0$  für alle  $j \in \overline{Succ^o(i)}$ ;

**Ausgabe:** zulässiger Schedule  $S^M := (S_i, (S_j^o)_{j \in \overline{Succ^o(i)}})$  für  $M_i$

---

In Zeile 6 beginnt die Schleife für die Einplanung von Rückbau- und Stoffstromvorgängen. Dazu wird in Zeile 7 der Startzeitpunkt  $S_{l^*}$  auf den frühestmöglichen Startzeitpunkt  $t \geq ES_{l^*}$  gesetzt, bei dem die Vorrangrestriktionen (3.1), die Freisetzungseinschränkungen (4.1), die stoffstrominduzierten Zeitrestriktionen (4.2), die erneuerbaren Ressourcenrestriktionen (4.11) eingehalten sind und ausreichend Kapazität für die zeitzulässige Mindestauslastung existiert. Letzteres bedeutet, dass eine Erhöhung des kumulativen Ressourcenprofils für alle  $k \in \mathcal{R}^Y(l^*)$  um die mittels (5.11) bzw. (5.12) berechnete zeitzulässige Mindestauslastung nicht zu einer Überschreitung der jeweiligen Kapazität  $R_k^Y$  führen darf.

Anschließend wird der relevante Vorgänger  $l^+$  von  $l^*$  ausgelesen (vgl. Zeile 8). Dann erfolgt eine Fallunterscheidung in Zeile 9, die sich wie folgt begründen lässt: Falls noch nicht alle relevanten Nachfolger  $j' \in Succ^-(l^+)$  eingeplant wurden, sind noch nicht alle Stoffeinheiten aus den kumulativen Ressourcen  $k \in \mathcal{R}^Y(l^+)$  ausgelagert. Es können daher Konflikte existieren, die durch die Einplanung der noch uneingeplanten relevanten Nachfolger wieder aufgelöst werden. In diesem Fall wäre es ungeeignet, die Zulässigkeit der kumulativen Ressourcenrestriktionen zu prüfen. Selbiges gilt, falls  $l^*$  ein Rückbauvorgang ist und daher in Zeile 8 kein relevanter Vorgänger ausgelesen werden konnte. Erst wenn

**Algorithmus 8:** StoffstromvorgangsfolgeAgg

**Eingabe:** Stoffstromvorgangsfolge  $\mu$ ; Rückbau- oder Stoffstromvorgang  $l$ ; Mengen der relevanten Nachfolger  $Succ^-$

- 1 sortiere  $Succ^-(l)$  mittels Tiefensuche im induzierten Out-Tree, wobei über kleinere Kantengewichte zu erreichende Knoten bevorzugt werden;
- 2 durchlaufe sortierte  $Succ^-(l)$  und füge alle  $j' \in Succ^-(l)$  zu  $\mu$  hinzu;
- 3 **für jeden nichtneutralen Stoffstromvorgang  $j'$  in sortierter  $Succ^-(l)$  tue**
- 4 |  $\mu := \text{StoffstromvorgangsfolgeAgg}(\mu, j', Succ^-)$ ;
- 5 **Ende**

**Ausgabe:** Stoffstromvorgangsfolge  $\mu$

bei der Fallunterscheidung in Zeile 9 alle relevanten Nachfolger von  $l^+$  eingeplant sind, besteht Aufschluss darüber, welche zusätzliche Auslastung der kumulativen Ressourcen anfällt. Falls dann Konflikte bestehen, müssen sie durch Ausplanung und zeitliche Rechtsverschiebung von  $l^+$  aufgelöst werden. Anstelle der kumulativen Ressourcenrestriktionen (4.15) wird dazu die Einhaltung der Ungleichungen

$$f_k^y(S, t) + r_k^y(S^o, t) \leq R_k^y \quad \left( \forall k \in \mathcal{R}^y(l^+), t = S_{l^+}, \dots, \max_{j' \in Succ^-(l^+)} (S_{j'}^o + d_{j'}) \right) \quad (5.13)$$

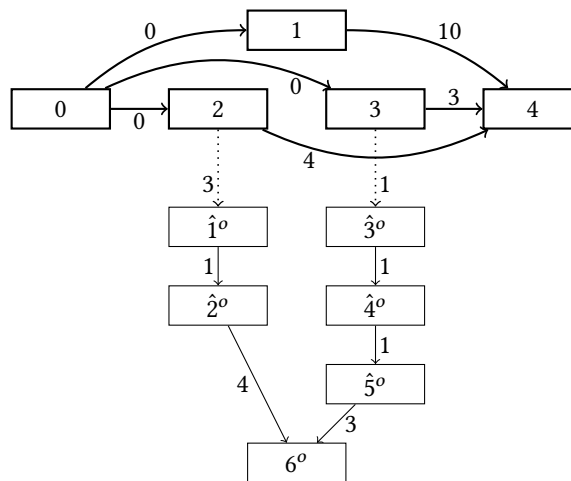
geprüft, die als *relevante kumulative Ressourcenrestriktionen* bezeichnet werden. Sie beschränken sich auf die kumulativen Ressourcen  $k \in \mathcal{R}^y(l^+)$  und die Zeitpunkte ab dem Start von  $l^+$  bis zum spätesten Ende seiner relevanten Nachfolger. Andere kumulativen Ressourcen  $k \notin \mathcal{R}^y(l^+)$  zu betrachten wäre ungeeignet, da Konflikte darin ohnehin nicht durch Rechtsverschiebung von  $l^+$  aufgelöst werden können. Dasselbe gilt für Konflikte zu Zeitpunkten vor dem Start von  $l^+$  oder nach dem spätesten Ende seiner relevanten Nachfolger. In Zeile 10 wird  $S_{l^+}$  solange erhöht, wie gemäß der relevanten kumulativen Ressourcenrestriktionen (5.13) ein Konflikt existiert (= die Restriktionen verletzt sind). Da die relevanten Nachfolger von  $l^+$  vollständig eingeplant sind und damit alle Stoffeinheiten von  $l^+$  zu einem bestimmten Zeitpunkt ausgelagert sein werden, wird immer ein  $S_{l^+}$  zu finden sein, für das die Restriktionen (5.13) eingehalten sind.

Anschließend werden in Zeile 11 alle relevanten Nachfolger von  $l^+$  ausgeplant. Diese Ausplanungen sind erforderlich, um die Rechtsverschiebung von  $l^+$  zu ermöglichen und sie nicht wegen Verletzungen der übrigen Restriktionen (d. h. Vorrang-, Freisetzung-, stoffstrominduzierte Zeit- und erneuerbare Ressourcenrestriktionen) zu blockieren. In den späteren Iterationen werden die ausgeplanten Stoffstromvorgänge dann wieder hinsichtlich der übrigen Restriktionen zulässig eingeplant. In Zeile 12 wird der neu berechnete Startzeitpunkt für  $l^+$  in der Hilfsvariable  $ES_{l^+}$  gespeichert und anschließend auch  $l^+$  ausgeplant.

In den verbleibenden Zeilen 14 bis 18 wird der in der nachfolgenden Iteration einzuplanende Vorgang  $l^*$  ausgewählt. Falls Rückbauvorgang  $i$  verschoben werden muss und deshalb ausgeplant wurde, wird  $l^* := i$  gesetzt (vgl. Zeile 15). Sonst wird  $l^*$  auf den ersten Stoffstromvorgang in  $\mu$  gesetzt, der noch uneingeplant ist.

SubScheduleAgg terminiert, sobald alle Stoffstromvorgänge  $j \in \overline{Succ^o}(i)$  zulässig eingeplant sind. In Beispiel 5.8 wird der Ablauf des Generierungsschemas anhand einer exemplarischen Instanz erläutert.

**Beispiel 5.8:** Eine Instanz  $\mathcal{I}_{RCPSP/M-agg}$  besteht aus drei realen Rückbauvorgängen 1, 2 und 3, fünf realen, aggregierten Stoffstromvorgängen  $\hat{1}^o, \dots, \hat{5}^o$ , einer erneuerbaren Ressource R1 und zwei kumulativen Ressourcen L1 und L2. Rückbauvorgang 1 dauert  $d_1 = 10$  und setzt keine Stoffe frei. Für Rückbauvorgang 2 gilt  $d_2 = 4$  und  $\overline{Succ^o}(2) = \{\hat{1}^o, \hat{2}^o\}$ . Für Rückbauvorgang 3 gilt  $d_3 = 3$  und  $\overline{Succ^o}(3) = \{\hat{3}^o, \hat{4}^o, \hat{5}^o\}$ . Die Dauern der Stoffstromvorgänge betragen  $d_{\hat{1}^o} = 2$ ,  $d_{\hat{2}^o} = 4$  und  $d_{\hat{3}^o} = d_{\hat{4}^o} = d_{\hat{5}^o} = 3$ . Abbildung 5.8 zeigt den Vorgangsknotennetzplan zu dieser Instanz, der die Informationen zu den Vorrangbeziehungen, den Freisetzungsbeziehungen und den stoffstrominduzierten Zeitbeziehungen mit den jeweiligen zeitlichen Mindestabständen (= Kantengewichte) bereitstellt. Rückbauvorgänge 0 und 4 sowie Stoffstromvorgang  $6^o$  sind fiktiv.



**Abbildung 5.8:** Vorgangsknotennetzplan zu Beispiel 5.8.

Rückbauvorgang 1 und Stoffstromvorgang  $\hat{5}^o$  nutzen  $r_{1,R1}^\alpha = 1$  bzw.  $r_{\hat{5}^o,R1}^\alpha = 1$  Einheiten der erneuerbaren Ressource R1, deren Kapazität  $R_{R1}^\alpha = 1$  beträgt. Die anderen Rückbau- und Stoffstromvorgänge nutzen keine erneuerbaren Ressourcen. Rückbauvorgang 2 setzt vier Stoffeinheiten in L1 frei. Diese werden von Stoffstromvorgang  $\hat{1}^o$  ausgelagert und in L2 eingelagert. Aus L2 werden sie anschließend von  $\hat{2}^o$  ausgelagert und erreichen danach die Senke. Rückbauvorgang 3 setzt drei Stoffeinheiten in L1 frei. Diese werden von Stoffstromvorgang  $\hat{3}^o$  verarbeitet. Weil Stoffstromvorgang  $\hat{3}^o$  die Stoffeinheiten jedoch wieder in L1 zurücklegt, handelt es sich um einen neutralen Stoffstromvorgang. Danach folgt  $\hat{4}^o$ , der die Stoffeinheiten aus L1 ausgelagert und in L2 einlagert. Aus L2 werden sie anschließend von  $\hat{5}^o$  ausgelagert und erreichen danach die Senke. Alle Freisetzungen, Ein- und Auslagerungen erfolgen gleichmäßig während der Vorgangsausführung, wie bei Instanzen  $\mathcal{I}_{RCPSP/M-agg}$  vorgesehen. Die kumulativen Ressourcenkapazitäten betragen  $R_{L1}^Y = R_{L2}^Y = 2$ .

Für die relevanten Nachfolger werden  $Succ^-(1) = \emptyset$ ,  $Succ^-(2) = \{\hat{1}^o\}$ ,  $Succ^-(\hat{1}^o) = \{\hat{2}^o\}$ ,  $Succ^-(3) = \{\hat{3}^o, \hat{4}^o\}$ ,  $Succ^-(\hat{4}^o) = \{\hat{5}^o\}$  und  $Succ^-(\hat{2}^o) = Succ^-(\hat{5}^o) = \emptyset$  berechnet. Angenommen, Prioritätsregel  $\pi$  besagt, dass zuerst Rückbauvorgang 1, dann Rückbauvorgang 2 und dann Rückbauvorgang 3 eingeplant wird. Tabelle 5.1 fasst die nachfolgend beschriebenen Iterationen der einzelnen Aufrufe von SubScheduleAgg für reale Rückbauvorgänge zusammen.

It.	$l^*$	$S_{l^*}$	Konflikte	Abb.
SubScheduleAgg( $M_1, \dots$ )				
1	1	0	nicht geprüft, da keine relevanten Vorgänger	
SubScheduleAgg( $M_2, \dots$ )				
1	2	0	nicht geprüft, da keine relevanten Vorgänger	
2	$\hat{1}^o$	3	geprüft für $l^+ := 2$ : kein Konflikt in $\mathcal{R}^Y(2) = \{L1\}$	5.9a
3	$\hat{2}^o$	4	geprüft für $l^+ := \hat{1}^o$ : kein Konflikt in $\mathcal{R}^Y(\hat{1}^o) = \{L2\}$	5.9b
SubScheduleAgg( $M_3, \dots$ )				
1	3	2	nicht geprüft, da keine relevanten Vorgänger	
2	$\hat{3}^o$	3	nicht geprüft, da noch nicht alle relevanten Nachfolger von $l^+ := 3$ eingeplant	
3	$\hat{4}^o$	4	geprüft für $l^+ := 3$ : kein Konflikt in $\mathcal{R}^Y(3) = \{L1\}$	5.9c
4	$\hat{5}^o$	10	geprüft für $l^+ := \hat{4}^o$ : Konflikt in $\mathcal{R}^Y(\hat{4}^o) = \{L2\}$ ; Ausplanung von $\hat{5}^o$ ; $ES_{\hat{4}^o} := 7$ ; Ausplanung von $\hat{4}^o$	5.9d
5	$\hat{4}^o$	7	geprüft für $l^+ := 3$ : Konflikt in $\mathcal{R}^Y(3) = \{L1\}$ ; Ausplanung von $\hat{3}^o$ u. $\hat{4}^o$ ; $ES_3 := 4$ ; Ausplanung von 3	5.9e
6	3	4	nicht geprüft, da keine relevanten Vorgänger	
7	$\hat{3}^o$	5	nicht geprüft, da noch nicht alle relevanten Nachfolger von $l^+ := 3$ eingeplant	
8	$\hat{4}^o$	7	geprüft für $l^+ := 3$ : kein Konflikt in $\mathcal{R}^Y(3) = \{L1\}$	
9	$\hat{5}^o$	10	geprüft für $l^+ := \hat{4}^o$ : kein Konflikt in $\mathcal{R}^Y(\hat{4}^o) = \{L2\}$	5.9f

**Tabelle 5.1:** Iterationen in Beispiel 5.8 (It.: Iteration).

Beim Aufruf von SubScheduleAgg für  $M_1$  wird der frühestmögliche Startzeitpunkt  $S_1 := 0$  bestimmt. Da in  $M_1$  keine Stoffstromvorgänge vorhanden sind, terminiert SubScheduleAgg.

Beim Aufruf von SubScheduleAgg für  $M_2$  wird in der ersten Iteration Rückbauvorgang 2 zum Zeitpunkt  $S_2 := 0$  eingeplant. In der zweiten Iteration wird  $S_{\hat{1}^o}^o := 3$  gesetzt. Da nun alle relevanten Nachfolger von  $l^+ := 2$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(2) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall (vgl. Abbildung 5.9a).

In der dritten Iteration wird  $S_{\hat{2}^o}^o := 4$  gesetzt. Da nun alle relevanten Nachfolger von  $l^+ := \hat{1}^o$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(\hat{1}^o) = \{L2\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall (vgl. Abbildung 5.9b). Alle Stoffstromvorgänge  $j \in Succ^o(1)$  sind eingeplant, womit SubScheduleAgg terminiert.

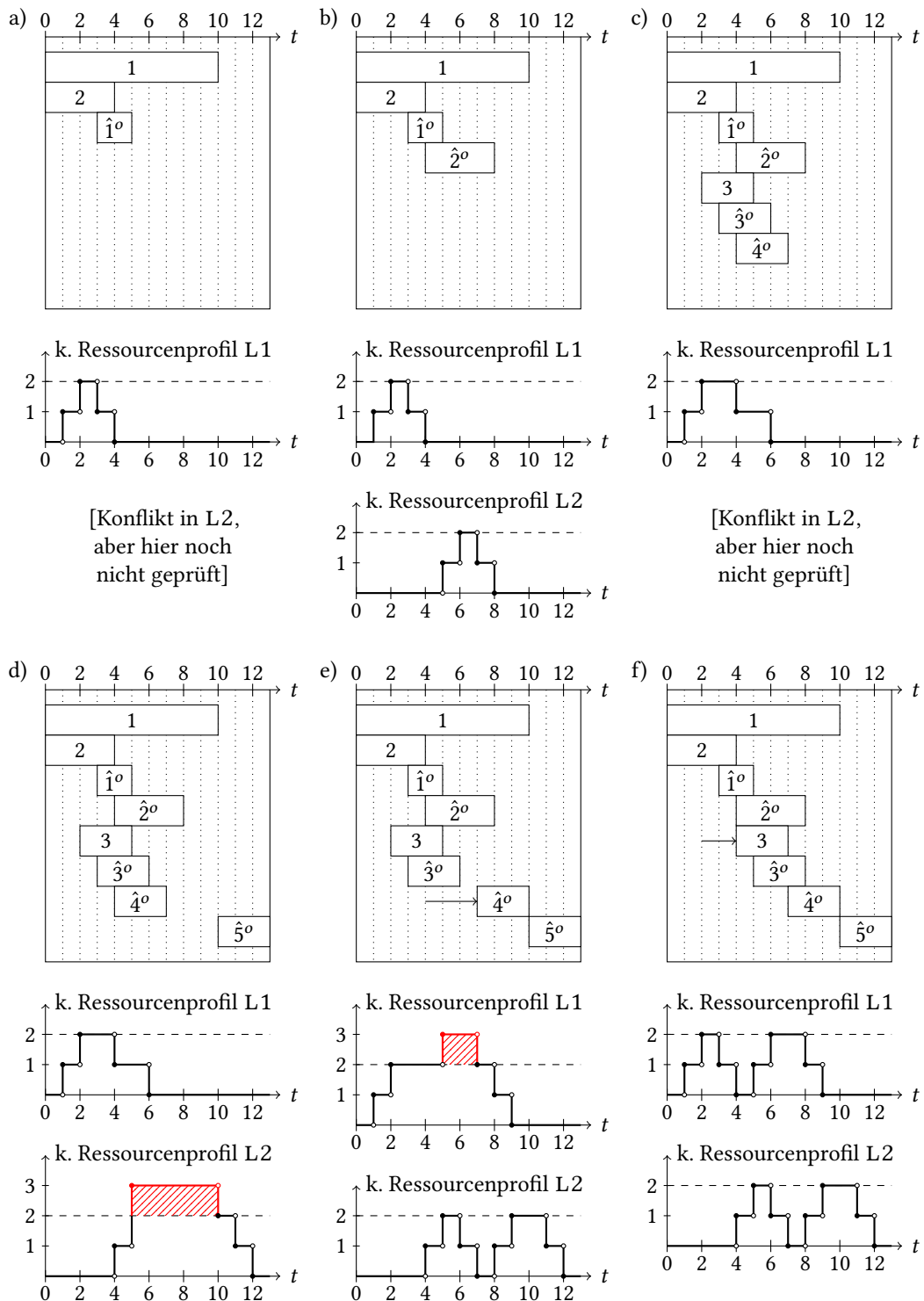


Abbildung 5.9: Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.8.

Beim Aufruf von SubScheduleAgg für  $M_3$  wird in der ersten Iteration Rückbauvorgang 3 zum Zeitpunkt  $S_3 := 2$  eingeplant. Eine frühere Einplanung ist aufgrund der zeitzulässigen Mindestauslastung nicht möglich: Selbst wenn die relevanten Nachfolger  $Succ^-(3) = \{\hat{3}^o, \hat{4}^o\}$  so früh wie möglich starten, wird immer eine Stoffeinheit in L1 eingelagert. Somit ist es aufgrund der bereits vorhandenen Auslastung in L1 frühestens zum Zeitpunkt 2 möglich, Rückbauvorgang 3 starten zu lassen. In der zweiten Iteration wird  $S_{\hat{3}^o}^o := 3$  gesetzt und in der dritten Iteration  $S_{\hat{4}^o}^o := 4$ . Da nun alle relevanten Nachfolger von  $l^+ := 3$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(3) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall (vgl. Abbildung 5.9c).

In der vierten Iteration wird  $S_{\hat{5}^o}^o := 10$  gesetzt. Eine frühere Einplanung ist aufgrund der erneuerbaren Ressourcenrestriktionen nicht möglich (Stoffstromvorgang  $\hat{5}^o$  kann nicht parallel zu Rückbauvorgang 1 ausgeführt werden). Da nun alle relevanten Nachfolger von  $l^+ := \hat{4}^o$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(\hat{4}^o) = \{L2\}$  kumulative Ressourcenkonflikte existieren. Das ist der Fall (vgl. Abbildung 5.9d). Gemäß relevanter kumulativer Ressourcenrestriktionen (5.13) muss der Startzeitpunkt von  $\hat{4}^o$  auf  $S_{\hat{4}^o}^o = 7$  erhöht werden. Frühestens dann kann die Kapazität  $R_{L2}^Y = 2$  eingehalten werden. Zunächst wird der relevante Nachfolger  $\hat{5}^o$  ausgeplant. Dann wird  $ES_{\hat{4}^o} := 7$  festgehalten und anschließend auch  $\hat{4}^o$  ausgeplant.

In der fünften Iteration wird mit  $\hat{4}^o$  fortgefahren und  $S_{\hat{4}^o}^o := ES_{\hat{4}^o} = 7$  gesetzt. Da nun alle relevanten Nachfolger von  $l^+ := 3$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(3) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist der Fall (vgl. Abbildung 5.9e). Gemäß relevanter kumulativer Ressourcenrestriktionen (5.13) muss der Startzeitpunkt von Rückbauvorgang 3 auf  $S_3 = 4$  erhöht werden. Frühestens dann kann die Kapazität  $R_{L1}^Y = 2$  eingehalten werden. Zunächst werden die relevanten Nachfolger  $\hat{3}^o$  und  $\hat{4}^o$  ausgeplant. Dann wird  $ES_3 := 4$  festgehalten und anschließend auch Rückbauvorgang 3 ausgeplant.

In der sechsten Iteration wird mit Rückbauvorgang 3 fortgefahren und  $S_3 := ES_3 = 4$  gesetzt. In der siebten Iteration wird  $S_{\hat{3}^o}^o := 5$  gesetzt und in der achten Iteration  $S_{\hat{4}^o}^o := ES_{\hat{4}^o} = 7$ . Hier wird ersichtlich, weshalb die Einführung der Hilfsvariablen  $ES_l$  sinnvoll ist: Ohne die Hilfsvariable würde  $S_{\hat{4}^o}^o$  in der achten Iteration auf den frühestmöglichen Startzeitpunkt sechs gesetzt werden, da die Information zum daraufhin zu erwartenden Konflikt in L2 verloren gegangen wäre. SubScheduleAgg würde zwar weiterhin eine zulässige Lösung liefern, allerdings werden dafür mehr Iterationen benötigt. Da nach Einplanung von  $\hat{4}^o$  alle relevanten Nachfolger von  $l^+ := 3$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(3) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall. In der neunten Iteration wird  $S_{\hat{5}^o}^o := 10$  gesetzt. Da nun alle relevanten Nachfolger von  $l^+ := \hat{4}^o$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(\hat{4}^o) = \{L2\}$  kumulative Ressourcenkonflikte existieren. Das ist ebenfalls nicht der Fall (vgl. Abbildung 5.9f). Alle Stoffstromvorgänge  $j \in Succ^o(2)$  sind eingeplant, womit SubScheduleAgg terminiert. Daraufhin terminiert auch SuperSchedule nach Einplanung der fiktiven Endvorgänge.  $\square$

Bei jeder Auflösung eines Konflikts in SubScheduleAgg wird ein Stoffstromvorgang  $j \in Succ^o(i)$  oder der Rückbauvorgang  $i$  um mindestens eine Periode nach rechts verschoben.

Dadurch wird `SubScheduleAgg` grundsätzlich einen zulässigen Schedule  $S^M$  liefern. Es gibt jedoch zwei Fälle, die zu Beginn der Schleife in Zeile 6 auftreten können und in denen das Verfahren fehlschlägt:

1. Falls Vorgang  $l^*$  soweit nach rechts verschoben wurde, dass

$$S_{l^*} > \max \left\{ \max_{i \in V \setminus \{l^*\}} (S_i + d_i), \max_{j \in V^o \setminus \overline{\text{Succ}}^o(l^*)} (S_j^o + d_j) \right\}.$$

Dann befindet sich kein anderer Vorgang außerhalb von  $M_i$  mehr in Ausführung. Es wird auch für spätere Zeitpunkte nicht möglich sein,  $l^*$  und seine relevanten Nachfolger einzuplanen. Die Restriktionen binden so stark, dass für die Stoffstromstruktur mit der in `StoffstromvorgangsfolgeAgg` berechneten Stoffstromvorgangsfolge  $\mu$  keine Lösung gefunden werden kann. Es ist möglich, dass mit einer anderen Stoffstromvorgangsfolge eine zulässige Lösung gefunden werden kann. Falls dies auch nicht möglich ist, ist die Instanz unzulässig.

2. Falls die zeitzulässige Mindestauslastung für mindestens einen Zeitpunkt  $t \geq 0$  und eine kumulativen Ressource  $k$  größer als deren Kapazität  $R_k^Y$  ist. Dann ist die Instanz unzulässig.

Diese Fälle sind aus Darstellungsgründen nicht explizit in Algorithmus 7 berücksichtigt.

`SubScheduleAgg` würde dasselbe Ergebnis liefern, wenn die Prüfung der zeitzulässigen Mindestauslastung in Zeile 7 übersprungen werden würde. Dann sind jedoch mehr Iterationen erforderlich, in denen jeweils ein Vorgang ausgeplant und verschoben wird. Dadurch, dass die zeitzulässige Mindestauslastung ohnehin nicht unterschritten werden kann, ist es vorteilhaft, sie bereits im Voraus einzubeziehen.

Abschließend sei darauf hingewiesen, dass die Ausgestaltung von `SubScheduleAgg` wie auch im granularen Fall die optimale Lösung ausschließen kann. Das liegt daran, dass lediglich die in `StoffstromvorgangsfolgeAgg` mittels Tiefensuche berechnete Stoffstromvorgangsfolge betrachtet wird. Außerdem führt die Zerlegung in die Ebene der Rückbauvorgänge und die Ebene der Stoffstromstrukturen dazu, dass der Lösungsraum nicht vollständig abgesucht werden kann.

### 5.4.3 Generierungsschema nach einem Integrationsansatz

Im Generierungsschema für aggregierte Instanzen wird bei der Auflösung kumulativer Ressourcenkonflikte nicht zwischen Rückbau- und Stoffstromvorgängen unterschieden, da sich beide Vorgangstypen in dieser Hinsicht ähneln. Es ist deshalb nicht zwingend erforderlich, die Aufteilung des Generierungsschemas in eine übergeordnete Ebene der Rückbauvorgänge und eine untergeordnete Ebene der Stoffstromstrukturen beizubehalten. Stattdessen kann das entwickelte Konzept der relevanten Nachfolger auch auf einen *Integrationsansatz* übertragen werden, bei dem Rückbau- und Stoffstromvorgänge gleichermaßen auf einer Ebene eingeplant werden. An dieser Stelle sei erneut darauf hingewiesen,

dass das Attribut „integriert“ nicht in Verbindung mit der Bezeichnung des (RCPSP/M) als „Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen“ steht, sondern sich ausschließlich auf die Ausgestaltung des Lösungsverfahrens bezieht.

Statt einer Prioritätsregel  $\pi$  wird für den Integrationsansatz eine Vorgangsfolge  $\ell := (0, l_1, l_2, \dots, l_{I+J}, I+1, (J+1)^o)$  verwendet, die die Einplanungsreihenfolge für das Generierungsschema vorgibt. Dies erleichtert die formale Beschreibung des Verfahrens. Die Vorgangsfolge  $\ell$  enthält alle Rückbau- und alle Stoffstromvorgänge und es gilt  $l_1, l_2, \dots, l_{I+J} \in V \cup V^o$ . Während die Position der fiktiven Rückbau- und Stoffstromvorgänge vorgegeben ist, kann die Reihenfolge der realen Rückbau- und Stoffstromvorgänge beliebig variieren, solange  $\ell$  zeitulässig ist. Zeitulässig bedeutet, dass sich für jeden Rückbau- oder Stoffstromvorgang alle direkten Vorgänger auf vorhergehenden Positionen in  $\ell$  befinden. Eine zeitulässige Vorgangsfolge kann in einem dem Generierungsschema vorgelagerten Schritt basierend auf einer Prioritätsregel berechnet werden.

Algorithmus 9 beschreibt das Generierungsschema nach einem Integrationsansatz, welches als `ScheduleAgg` bezeichnet wird. Der Verfahrensablauf ist zu großen Teilen ähnlich wie in `SubScheduleAgg`, weshalb er hier nicht mehr im Detail erläutert wird. Stattdessen werden nachfolgend lediglich die Unterschiede zum Dekompositionsansatz hervorgehoben. Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 9.

Die im Dekompositionsansatz verwendete und deterministisch bestimmte Stoffstromvorgangsfolge  $\mu$  wird in `ScheduleAgg` nicht mehr benötigt, da die Einplanungsreihenfolge der Stoffstromvorgänge durch  $\ell$  beschrieben wird.

In  $\ell$  ist nicht mehr sichergestellt, dass die relevanten Nachfolger wie beim Dekompositionsansatz immer direkt nach der Einplanung ihres relevanten Vorgängers eingeplant werden. Es ist somit möglich, dass zunächst mehrere Vorgänge eingeplant werden, die in eine kumulative Ressource einlagern, bevor diese Stoffeinheiten wieder ausgelagert werden. Um dennoch zulässige Startzeitpunkte bestimmen zu können, ist es erforderlich, dass Einlagerungen durch einen Vorgang  $l$  bei der Prüfung der kumulativen Ressourcenrestriktionen erst einberechnet werden, sobald auch die relevanten Nachfolger von  $l$  eingeplant sind. Dazu werden die relevanten kumulativen Ressourcenrestriktionen neu definiert als

$$\begin{aligned}
 & \underbrace{\sum_{i \in \mathcal{AV}(S, t)} \min\{f_{ik}^Y(S_i, t), \chi_i(S^o)\}}_{\text{I}} + \underbrace{\sum_{j \in \mathcal{AV}(S^o, t) | r_{jk}^Y(\cdot, \infty) > 0} \min\{r_{jk}^Y(S_j^o, t), \chi_j(S^o)\}}_{\text{II}} \\
 & + \underbrace{\sum_{j \in \mathcal{AV}(S^o, t) | r_{jk}^Y(\cdot, \infty) \leq 0} r_{jk}^Y(S_j^o, t)}_{\text{III}} \leq R_k^Y \\
 & \left( \forall k \in \mathcal{R}^Y(I^+), t = S_{I^+}, \dots, \max_{j' \in \text{Succ}^-(I^+)} (S_{j'}^o + d_{j'}) \right) \quad (5.14)
 \end{aligned}$$



mit

$$\chi_l(S^o) := \begin{cases} \infty & \text{falls } S_{j'}^o \geq 0 \text{ für alle } j' \in \text{Succ}^-(l), \\ 0 & \text{sonst.} \end{cases}$$

Summand I umfasst die Stofffreisetzungen aller Rückbauvorgänge in die kumulative Ressource  $k$  zum Zeitpunkt  $t$ . Summand II umfasst die Einlagerungen aller in  $k$  einlagernden Stoffstromvorgänge zum Zeitpunkt  $t$ . Sowohl in Summand I als auch in Summand II führt der Minimumsausdruck in Kombination mit der Funktion  $\chi_l(S^o)$  dazu, dass Einlagerungen erst berücksichtigt werden, sobald alle relevanten Nachfolger eines einlagernden Vorgangs eingeplant sind. Summand III umfasst die kumulativen Ressourcennutzungen aller übrigen Stoffstromvorgänge (= neutrale und auslagernde Stoffstromvorgänge). Die Restriktionen (5.14) beschränken sich wie beim Dekompositionsansatz nur auf  $k \in \mathcal{R}^Y(l^+)$  und auf Zeitpunkte ab dem Start von  $l^+$  bis zum spätesten Ende seiner relevanten Nachfolger.

---

**Algorithmus 9:** ScheduleAgg

---

**Eingabe:** Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$ ; Vorgangsfolge  $\ell$

- 1 initialisiere partiellen Schedule  $(S, S^o)$  mit  $S_i := -\infty$  für alle  $i \in V$  und  $S_j^o := -\infty$  für alle  $j \in V^o$ ;
  - 2 initialisiere  $\text{Succ}^-(l) := \emptyset$  für alle  $l \in V \cup V^o$ ;
  - 3 berechne  $\text{Succ}^-(l) = \text{RelevanteNachfolgerAgg}(\mathcal{I}_{\text{RCPSP/M-agg}}, l, k, \text{Succ}^-(l), \emptyset)$  für alle  $l \in V \cup V^o$  und alle  $k \in \mathcal{R}^Y(l)$ ;
  - 4 initialisiere  $ES_l := -\infty$  für alle  $l \in V \cup V^o$ ;
  - 5 setze  $l^* := 0$ ;
  - 6 **wiederhole**
    - 7 setze  $S_{l^*}$  auf das frühestmögliche  $t \geq ES_{l^*}$ , sodass (3.1), (4.1), (4.2), (4.11) eingehalten sind und Kapazität für zeitzulässige Mindestauslastung existiert;
    - 8 lese den relevanten Vorgänger  $l^+$  von  $l^*$  aus, falls möglich;
    - 9 **wenn**  $l^+$  *ausgelesen werden konnte* **und**  $S_{j'}^o \geq 0$  für alle  $j' \in \text{Succ}^-(l^+)$  **und** Restriktionen (5.14) *verletzt werden* **dann**
      - 10 setze  $S_{old} := S_{l^+}$ ;
      - 11 erhöhe  $S_{l^+}$  bis Restriktionen (5.14) eingehalten sind;
      - 12 setze  $ES_l := -\infty$  und  $S_l := -\infty$  für alle  $l \in (V \cup V^o) \setminus \{l^+\}$  mit  $S_l \geq S_{old}$ ;
      - 13 setze  $ES_{l^+} := S_{l^+}$  und  $S_{l^+} := -\infty$ ;
    - 14 **Ende**
    - 15 **wenn** ein Vorgang  $l$  in  $\ell$  mit  $S_l = -\infty$  existiert **dann**
      - 16 setze  $l^*$  auf den ersten Vorgang  $l$  in  $\ell$  mit  $S_l = -\infty$ ;
      - 17 **Ende**
  - 18 **bis**  $S_{(J+1)^o}^o \geq 0$ ;
- Ausgabe:** zulässiger Schedule  $(S, S^o)$  für  $\mathcal{I}_{\text{RCPSP/M-agg}}$
- 

Für einen partiellen Schedule  $(S, S^o)$  und alle Zeitpunkte  $t \geq 0$  beschreibt die linke Seite von (5.14) das kumulative Ressourcenprofil für eine kumulative Ressource  $k$ . Dieses wird in Zeile 7 herangezogen, um zu prüfen, ob die Kapazität für die zeitzulässige Mindestauslastung ausreicht.

In Zeile 11 wird  $S_{l^+}$  solange erhöht, wie gemäß (5.14) ein kumulativer Ressourcenkonflikt existiert. Da die relevanten Nachfolger von  $l^+$  vollständig eingeplant sind und damit alle Stoffeinheiten von  $l^+$  zu einem bestimmten Zeitpunkt ausgelagert sein werden, wird immer ein  $S_{l^+}$  zu finden sein, für das die Restriktionen (5.14) eingehalten sind.

Vor der Erhöhung von  $S_{l^+}$  wurde in Zeile 10 der bisherige Startzeitpunkt in der Hilfsvariable  $S_{old}$  festgehalten. Er wird in Zeile 12 benötigt. Dort werden nicht mehr nur alle relevanten Nachfolger von  $l^+$  ausgeplant, sondern alle Rückbau- und Stoffstromvorgänge, die zu einem Startzeitpunkt  $S_l \geq S_{old}$  eingeplant sind. Die umfangreichen Ausplanungen sind erforderlich, um die Rechtsverschiebung von  $l^+$  zu ermöglichen und sie nicht wegen Verletzungen anderer Restriktionen zu blockieren. Gleichzeitig können dadurch auch Rückbau- und Stoffstromvorgänge, die keine transitiven Nachfolger von  $l^+$  sind, nach der Verschiebung von  $l^+$  eventuell früher eingeplant werden (siehe Beispiel 5.9). Zimmermann u. a. (2006, S. 253f) und Neumann, Schwindt und Zimmermann (2003, S. 86f) gehen bei den Ausplanungsschritten in ihren Generierungsschemas zum (RCPSp/max) analog vor. Um die frühere Einplanung von Rückbau- und Stoffstromvorgängen  $l$  mit  $S_l \geq S_{old}$  zu ermöglichen, werden in Zeile 12 auch die Belegungen der Hilfsvariablen  $ES_l$  für alle auszuplanenden Vorgänge gelöscht.

Als Abbruchkriterium ist es in Zeile 18 ausreichend, zu prüfen, ob der sich auf der letzten Position in  $\ell$  befindliche fiktive Stoffstromvorgang  $(J + 1)^o$  eingeplant ist. Dann ist auch der Schedule  $(S, S^o)$  zulässig, da  $(J + 1)^o$  ansonsten nicht zur Einplanung ausgewählt worden wäre. In Beispiel 5.9 wird der Ablauf des Generierungsschemas anhand einer exemplarischen Instanz erläutert.

**Beispiel 5.9:** Eine Instanz  $\mathcal{I}_{RCPSp/M-agg}$  besteht aus zwei realen Rückbauvorgängen 1 und 2, zwei realen Stoffstromvorgängen  $\hat{1}^o$  und  $\hat{2}^o$ , einer erneuerbaren Ressource R1 und einer kumulativen Ressource L1. Für Rückbauvorgang 1 gilt  $d_1 = 3$  und  $Succ^o(1) = \{\hat{1}^o\}$ . Für Rückbauvorgang 2 gilt  $d_2 = 3$  und  $Succ^o(2) = \{\hat{2}^o\}$ . Die Dauern der Stoffstromvorgänge betragen  $d_{\hat{1}^o} = d_{\hat{2}^o} = 3$ . Abbildung 5.10 zeigt den Vorgangsknotennetzplan zu dieser Instanz, der die Informationen zu den Vorrangbeziehungen, den Freisetzungsbeziehungen und den stoffstrominduzierten Zeitbeziehungen mit den jeweiligen zeitlichen Mindestabständen (= Kantengewichte) bereitstellt. Rückbauvorgänge 0 und 3 sowie Stoffstromvorgang  $3^o$  sind fiktiv.

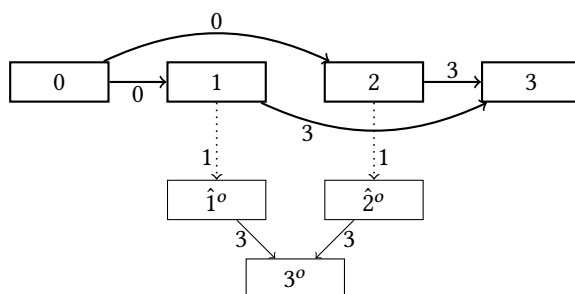


Abbildung 5.10: Vorgangsknotennetzplan zu Beispiel 5.9.

Alle realen Rückbau- und Stoffstromvorgänge nutzen eine Einheit der erneuerbaren Ressource R1, deren Kapazität  $R_{R1}^\alpha = 1$  beträgt. Das heißt, es ist keine parallele Ausführung von Vorgängen möglich. Rückbauvorgang 1 setzt drei Stoffeinheiten in L1 frei. Diese werden von Stoffstromvorgang  $\hat{1}^o$  ausgelagert und erreichen danach die Senke. Für Rückbauvorgang 2 gilt Gleiches: Er setzt drei Stoffeinheiten in L1 frei, die von  $\hat{2}^o$  ausgelagert werden. Alle Freisetzungen und Auslagerungen erfolgen gleichmäßig während der Vorgangsausführung, wie bei Instanzen  $\bar{I}_{RCPSP/M-agg}$  vorgesehen. Die kumulative Ressourcenkapazität beträgt  $R_{L1}^Y = 2$ .

Für die relevanten Nachfolger werden  $Succ^-(1) = \{\hat{1}^o\}$ ,  $Succ^-(2) = \{\hat{2}^o\}$ ,  $Succ^-(\hat{1}^o) = Succ^-(\hat{2}^o) = \emptyset$  berechnet. Angenommen, als Vorgangsfolge wird  $\ell := (0, 1, 2, \hat{1}^o, \hat{2}^o, 3, 3^o)$  gewählt. Sie ist zeitulässig, da alle Stoffstromvorgänge nach ihren vorangehenden Rückbauvorgängen kommen. Tabelle 5.2 fasst die nachfolgend beschriebenen Iterationen von ScheduleAgg zusammen.

In der ersten Iteration wird der fiktive Rückbauvorgang 0 zum Zeitpunkt  $S_0 := 0$  eingeplant. In der zweiten Iteration wird Rückbauvorgang 1 zum Zeitpunkt  $S_1 := 0$  eingeplant. In der dritten Iteration wird Rückbauvorgang 2 aufgrund der erneuerbaren Ressourcenrestriktionen zum Zeitpunkt  $S_2 := 3$  eingeplant. In der vierten Iteration wird Stoffstromvorgang  $\hat{1}^o$  aufgrund der erneuerbaren Ressourcenrestriktionen zum Zeitpunkt  $S_{\hat{1}^o}^o = 6$  eingeplant. Da nun alle relevanten Nachfolger von  $l^+ := 1$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(1) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist der Fall (vgl. Abbildung 5.11a). Gemäß relevanter kumulativer Ressourcenrestriktionen (5.14) muss der Startzeitpunkt von Rückbauvorgang 1 auf  $S_1 = 3$  erhöht werden. Frühestens dann kann die Kapazität von  $R_{L1}^Y = 2$  eingehalten werden. Im Zuge der Rechtsverschiebung von Rückbauvorgang 1 werden Rückbauvorgang 2 und Stoffstromvorgang  $\hat{1}^o$  ausgeplant, da  $S_{old} = 0$  und somit  $S_2 > S_{old}$  und  $S_{\hat{1}^o}^o > S_{old}$ . Dann wird  $ES_1 := 3$  festgehalten und anschließend auch Rückbauvorgang 1 ausgeplant.

Bei der Rechtsverschiebung von Rückbauvorgang 1 macht sich bemerkbar, dass in den relevanten kumulativen Ressourcenrestriktionen (5.14) die Einlagerungen von Rückbauvorgang 2 noch nicht berücksichtigt werden: Da die relevanten Nachfolger  $Succ^-(2) = \{\hat{2}^o\}$  noch uneingeplant sind, gilt  $\chi_2(S^o) = 0$ . Dadurch wird  $f_{2,L1}^Y(3, t)$  in Summand I von (5.14) nicht einbezogen, sondern das Minimum von 0 gewählt. Ohne einen solchen Mechanismus könnte nie ein zulässiger Startzeitpunkt für Rückbauvorgang 1 gefunden werden, da die Kapazität von L1 für alle  $t \geq 3$  überschritten wäre.

In der vierten, fünften und sechsten Iteration wird  $S_1 := ES_1 = 3$ ,  $S_2 := 0$  und  $S_{\hat{1}^o}^o = 6$  bestimmt. Da nun alle relevanten Nachfolger von  $l^+ := 1$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(1) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall (vgl. Abbildung 5.11b).

In der siebten Iteration wird Stoffstromvorgang  $\hat{2}^o$  aufgrund der erneuerbaren Ressourcenrestriktionen zum Zeitpunkt  $S_{\hat{2}^o}^o = 9$  eingeplant. Da nun alle relevanten Nachfolger von  $l^+ := 2$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^Y(2) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist der Fall (vgl. Abbildung 5.11c). Gemäß relevanter kumulativer

It.	$l^*$	$S_{l^*}$	Konflikte	Abb.
1	0	0	nicht geprüft, da keine relevanten Vorgänger	
2	1	0	nicht geprüft, da keine relevanten Vorgänger	
3	2	3	nicht geprüft, da keine relevanten Vorgänger	
4	$\hat{1}^o$	6	geprüft für $l^+ := 1$ : Konflikt in $\mathcal{R}^V(1) = \{L1\}$ ; Ausplanung von 2 u. $\hat{1}^o$ ; $ES_1 := 3$ ; Ausplanung von 1	5.11a
5	1	3	nicht geprüft, da keine relevanten Vorgänger	
6	2	0	nicht geprüft, da keine relevanten Vorgänger	
7	$\hat{1}^o$	6	geprüft für $l^+ := 1$ : kein Konflikt in $\mathcal{R}^V(1) = \{L1\}$	5.11b
8	$\hat{2}^o$	9	geprüft für $l^+ := 2$ : Konflikt in $\mathcal{R}^V(2) = \{L1\}$ ; Ausplanung von 1, $\hat{1}^o$ u. $\hat{2}^o$ ; $ES_1 := -\infty$ ; $ES_2 := 6$ ; Ausplanung von 2	5.11c
9	1	0	nicht geprüft, da keine relevanten Vorgänger	
10	2	6	nicht geprüft, da keine relevanten Vorgänger	
11	$\hat{1}^o$	3	geprüft für $l^+ := 1$ : kein Konflikt in $\mathcal{R}^V(1) = \{L1\}$	
12	$\hat{2}^o$	9	geprüft für $l^+ := 2$ : kein Konflikt in $\mathcal{R}^V(2) = \{L1\}$	5.11d
13	3	9	nicht geprüft, da keine relevanten Vorgänger	
11	$3^o$	12	nicht geprüft, da keine relevanten Vorgänger	

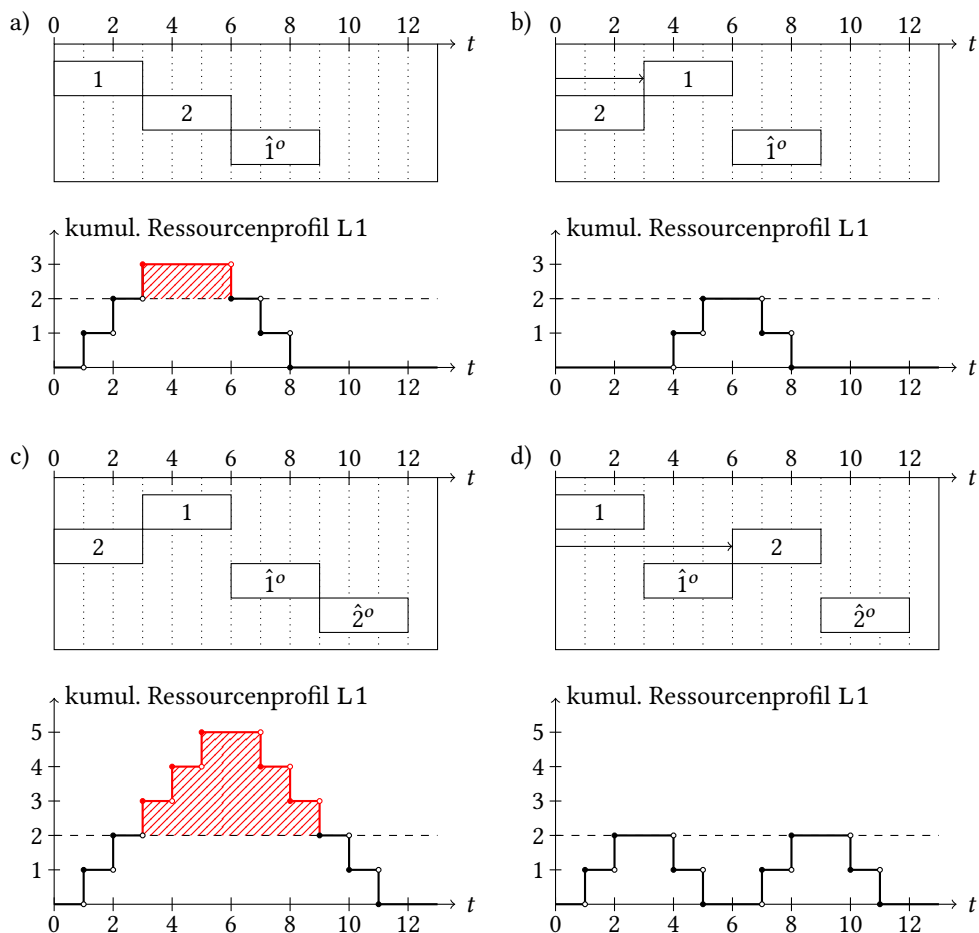
**Tabelle 5.2:** Iterationen in Beispiel 5.9 (It.: Iteration).

Ressourcenrestriktionen (5.14) muss der Startzeitpunkt von Rückbauvorgang 2 auf  $S_2 = 6$  erhöht werden. Rückbauvorgang 1 und die beiden Stoffstromvorgänge werden im Zuge der Rechtsverschiebung von Rückbauvorgang 2 ausgeplant. Für Rückbauvorgang 1 wird auch  $ES_1 := -\infty$  zurückgesetzt. Dadurch wird ermöglicht, dass er in der darauffolgenden Iteration früher eingeplant werden kann. Dann wird  $ES_2 := 6$  festgehalten und anschließend auch Rückbauvorgang 2 ausgeplant.

In der achten, neunten und zehnten Iteration wird  $S_1 := 0$ ,  $S_2 := ES_2 = 6$  und  $S_{\hat{1}^o}^o = 3$  bestimmt. Da nun alle relevanten Nachfolger von  $l^+ := 1$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^V(1) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall.

In der elften Iteration wird  $S_{\hat{2}^o}^o = 9$  bestimmt. Da nun alle relevanten Nachfolger von  $l^+ := 2$  eingeplant sind, wird geprüft, ob in  $\mathcal{R}^V(2) = \{L1\}$  kumulative Ressourcenkonflikte existieren. Das ist nicht der Fall (vgl. Abbildung 5.11d). Danach können der fiktive Rückbauvorgang 3 zum Zeitpunkt  $S_3 := 9$  und der fiktive Stoffstromvorgang  $3^o$  zum Zeitpunkt  $S_{3^o}^o = 12$  eingeplant werden, woraufhin `ScheduleAgg` terminiert.

Wird alternativ Vorgangsfolge  $\ell := (0, 1, \hat{1}^o, 2, \hat{2}^o, 3, 3^o)$  verwendet, kann `ScheduleArb` bereits in sieben Iterationen (d. h. ohne Ausplanungsschritte) einen zulässigen `Schedule` bestimmen. Der Ablauf des Verfahrens ist dann identisch zum Generierungsschema nach dem Dekompositionsansatz, da die relevanten Nachfolger jeweils in der Iteration direkt nach der Einplanung ihres Vorgängers eingeplant werden.  $\square$



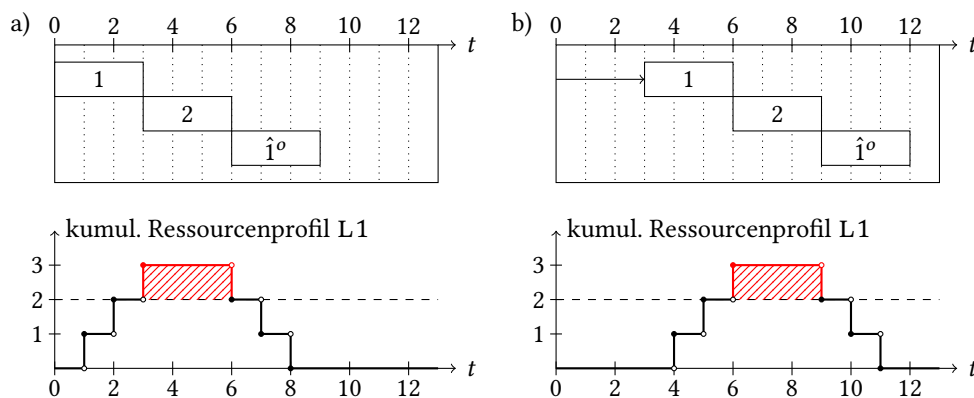
**Abbildung 5.11:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.9.

Dadurch, dass in  $\ell$  jede beliebige zeit zulässige Reihenfolge möglich ist, können Stoffstromvorgänge weit hinter der Position ihrer relevanten Vorgänger liegen. Bei eng bindenden Restriktionen kann für solche Vorgangsfolgen unter Umständen kein zulässiger Schedule bestimmt werden. `ScheduleAgg` wird nicht terminieren, sondern sich in einer Endlosschleife aus Ein- und Ausplanungsschritten verfangen. Zwar wird bei jedem Ausplanungsschritt ein Rückbau- oder Stoffstromvorgang um mindestens eine Periode nach rechts verschoben. Im (RCPSP/M) gibt es jedoch keine Obergrenze für die Projektdauer, weshalb endlose Rechtsverschiebungen theoretisch möglich sind. Ein vergleichbares Verhalten lässt sich auch bei anderen Generierungsschemas mit Ausplanungsschritten beobachten, wie in Zimmermann u. a. (2006, S. 253f) und Neumann, Schwindt und Zimmermann (2003, S. 86f) beim (RCPSP/max). Eine Endlosschleife aus Ein- und Ausplanungsschritten effizient zu erkennen und damit das Generierungsschema als fehlgeschlagen bezeichnen zu können, stellt aufgrund der Problemkomplexität eine Herausforderung dar. Zimmermann u. a. (2006, S. 253f) und Neumann, Schwindt und Zimmermann (2003, S. 86f) verwenden eine nicht näher definierte maximale Anzahl an Ausplanungsschritten, nach denen das Verfahren abbricht und als fehlgeschlagen gilt. Diese maximale Anzahl an Ausplanungsschritten kann jedoch nicht pauschal festgelegt werden, da je nach Instanzgröße und -beschaffenheit

bereits eine sehr kleine Zahl oder erst eine sehr große Zahl ausreicht, um eine Aussage zu treffen, ob die Schleife in `ScheduleAgg` als endlos gilt. Deshalb wird hier ein anderer Ansatz gewählt: Sobald ein Rückbau- oder Stoffstromvorgang so weit nach rechts verschoben wurde, dass er zu einem Zeitpunkt  $t > \sum_{i \in V} d_i + \sum_{j \in V^o} d_j$  eingeplant werden soll, gilt `ScheduleAgg` als fehlgeschlagen. Dieser Fall ist in Zeile 7 zu prüfen. Die aufsummierte Dauer aller Rückbau- und Stoffstromvorgänge entspricht derjenigen Projektdauer, die bei sequenzieller Ausführung aller Vorgänge zu beobachten ist. Es wird davon ausgegangen, dass dies eine obere Schranke für die Projektdauer darstellt und eine spätere Einplanung zu keinem zulässigen Schedule mehr führen wird. Es ist nicht ausgeschlossen, dass sich mittels aufwendigerer Mechanismen bereits früher Fehlschläge erkennen lassen. Im Rahmen dieser Arbeit wird hier jedoch nicht weiter in die Tiefe gegangen. Beispiel 5.10 veranschaulicht eine Situation, in der `ScheduleAgg` fehlschlagen wird.

**Beispiel 5.10:** Die Instanz  $\mathcal{I}_{\text{RCPSP/M-agg}}$  aus dem vorangegangenen Beispiel 5.9 wird unverändert übernommen. Es wird lediglich eine Vorrangbeziehung (1, 2) hinzugefügt, d. h., Rückbauvorgang 1 muss vor dem Start von Rückbauvorgang 2 enden.

Bei Verwendung der Vorgangsfolge  $\ell := (0, 1, 2, \hat{1}^o, \hat{2}^o, 3, 3^o)$  werden wie zuvor  $S_0 := 0$ ,  $S_1 := 0$ ,  $S_2 := 3$  und  $S_{\hat{1}^o}^o = 6$  bestimmt (vgl. Abbildung 5.12a). Da in L1 ein Konflikt vorliegt, wird Rückbauvorgang 1 um drei Perioden nach rechts verschoben, d. h.  $ES_1 := 3$ . Rückbauvorgang 2 und Stoffstromvorgang  $\hat{1}^o$  werden ausgeplant.



**Abbildung 5.12:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.10.

In den darauffolgenden Iterationen wird  $S_1 := ES_1 = 3$ ,  $S_2 := 6$  und  $S_{\hat{1}^o}^o = 9$  bestimmt. Es liegt dann erneut ein kumulativer Ressourcenkonflikt in L1 vor (vgl. Abbildung 5.12b). Dieser Konflikt unterscheidet sich vom vorangegangenen nur dadurch, dass er um drei Perioden nach rechts verschoben ist. `ScheduleAgg` löst ihn auf dieselbe Weise auf, wodurch er drei Perioden später wieder auftritt. Die Schleife würde endlos weiter laufen. Mit dem beschriebenen, zusätzlichen Abbruchkriterium wird ab  $S_1 > 12$  ein Fehlschlag festgestellt und `ScheduleAgg` terminiert. Grund für den Fehlschlag ist das Zusammenspiel aus Vorrang-, erneuerbaren und kumulativen Ressourcenrestriktionen, die für die gegebene Vorgangsfolge  $\ell$  nicht eingehalten werden können. Falls stattdessen  $\ell := (0, 1, \hat{1}^o, 2, \hat{2}^o, 3, 3^o)$  verwendet wird, berechnet `ScheduleAgg` einen zulässigen Schedule.  $\square$

## 5.5 Lösungsverfahren für beliebige Instanzen des (RCPSP/M)

Falls Instanzen des (RCPSP/M) nicht mit dem granularen oder aggregierten Transformationsschema erzeugt wurden, wird von *beliebigen Instanzen*  $\mathcal{I}_{\text{RCPSP/M}}$  gesprochen. Nachfolgend wird ein Lösungsverfahren vorgestellt, das für beliebige Instanzen  $\mathcal{I}_{\text{RCPSP/M}}$  anwendbar ist. Darüber hinaus ist dieses Lösungsverfahren auch für Instanzen des (RCPSP/c) anwendbar, d. h. auf die Generalisierung des (RCPSP/M), welche nicht zwischen Rückbau- und Stoffstromvorgängen unterscheidet.

### 5.5.1 Grundlegende Konzepte

Für beliebige Instanzen  $\mathcal{I}_{\text{RCPSP/M}}$  ist zunächst zu klären, wann ein Stoffstromvorgang als ein- oder auslagernd gilt. Für das granulare oder aggregierte Generierungsschema war es ausreichend, die Stoffstromvorgänge  $j \in V^o$  bezüglich einer kumulativen Ressource  $k \in \mathcal{R}^Y$  wie in Abschnitt 4.1 eingeführt als einlagernd (falls  $r_{jk}^Y(\cdot, \infty) > 0$ ), auslagernd (falls  $r_{jk}^Y(\cdot, \infty) < 0$ ) oder neutral (falls  $r_{jk}^Y(\cdot, \infty) = 0$ ) zu unterscheiden. Dies ist ausreichend, da die Nutzungen stets dem vorgegebenen Schema folgen. In beliebigen Instanzen ist es jedoch beispielsweise möglich, dass ein Stoffstromvorgang eine kumulative Ressource zu seinem Start positiv und zu seinem Ende negativ nutzt. Dann ist er sowohl ein- als auch ein auslagernder Stoffstromvorgang und muss im Generierungsschema entsprechend behandelt werden. Zur sprachlichen Unterscheidung wird daher im Zusammenhang mit beliebigen Instanzen auf die Partizipien „einlagernd“ und „auslagernd“ verzichtet, und stattdessen von „positiv nutzenden“ und „negativ nutzenden“ Stoffstromvorgängen gesprochen.

Formal muss für jeden Stoffstromvorgang  $j \in V^o$  und jede kumulative Ressource  $k \in \mathcal{R}^Y$  unterschieden werden, ob die kumulativen Ressourcennutzungen positiv oder negativ sind. Dazu werden

$$\begin{aligned} r_{jk}^{Y+,start} &:= \max\{r_{jk}^{Y,start}, 0\} \text{ und } r_{jk}^{Y-,start} := \min\{r_{jk}^{Y,start}, 0\}, \\ r_{jk}^{Y+,lin} &:= \max\{r_{jk}^{Y,lin}, 0\} \text{ und } r_{jk}^{Y-,lin} := \min\{r_{jk}^{Y,lin}, 0\}, \\ r_{jk}^{Y+,end} &:= \max\{r_{jk}^{Y,end}, 0\} \text{ und } r_{jk}^{Y-,end} := \min\{r_{jk}^{Y,end}, 0\} \end{aligned}$$

bestimmt. Dann kann zwischen der Anzahl der durch einen Stoffstromvorgang  $j \in V^o$  mit gegebenem Startzeitpunkt  $S_j^o$  zu einem Zeitpunkt  $t \geq S_j^o$  positiv genutzten Einheiten

$$r_{jk}^{Y+}(S_j^o, t) := \begin{cases} r_{jk}^{Y+,start} & \text{falls } t = S_j^o, \\ r_{jk}^{Y+,start} + \left\lfloor (t - S_j^o) r_{jk}^{Y+,lin} / d_j \right\rfloor & \text{falls } S_j^o < t < S_j^o + d_j, \\ r_{jk}^{Y+,start} + r_{ik}^{Y+,lin} + r_{jk}^{Y+,end} & \text{falls } t \geq S_j^o + d_j, \end{cases} \quad (5.15)$$

und negativ genutzten Einheiten

$$r_{jk}^{Y-}(S_j^o, t) := \begin{cases} r_{jk}^{Y-,start} & \text{falls } t = S_j^o, \\ r_{jk}^{Y-,start} + \left\lfloor (t + 1 - S_j^o) r_{jk}^{Y-,lin} / d_j \right\rfloor & \text{falls } S_j^o < t < S_j^o + d_j, \\ r_{jk}^{Y-,start} + r_{ik}^{Y-,lin} + r_{jk}^{Y-,end} & \text{falls } t \geq S_j^o + d_j, \end{cases} \quad (5.16)$$

einer kumulativen Ressource  $k \in \mathcal{R}^Y$  unterschieden werden. Diese Gleichungen sind analog zu Gleichung (4.12) aufgebaut, mit welcher  $r_{jk}^Y(S_j^o, t)$  definiert wurde. Sie unterscheiden sich lediglich dadurch, dass die Nutzungen nach ihrem Vorzeichen gefiltert werden. Infolgedessen kann auch auf die Fallunterscheidung zwischen  $r_{jk}^{Y,lin} \geq 0$  und  $r_{jk}^{Y,lin} < 0$ , welche in Gleichung (4.12) vorgenommen wird, verzichtet werden.

Die gesamte Anzahl der durch Stoffstromvorgang  $j$  positiv bzw. negativ genutzten Einheiten einer kumulativen Ressource  $k$  wird als  $r_{jk}^{Y+}(\cdot, \infty)$  bzw.  $r_{jk}^{Y-}(\cdot, \infty)$  bezeichnet. Ein Stoffstromvorgang  $j$  gilt bezüglich einer kumulativen Ressource  $k$  als *positiv nutzend*, falls  $r_{jk}^{Y+}(\cdot, \infty) > 0$  und als *negativ nutzend*, falls  $r_{jk}^{Y-}(\cdot, \infty) < 0$ . Es ist durchaus möglich, dass ein Stoffstromvorgang bezüglich einer kumulativen Ressource  $k$  sowohl als positiv als auch negativ nutzend gilt.

Ferner bezeichnen

$$\mathcal{R}^{Y+}(l) := \begin{cases} \{k \in \mathcal{R}^Y \mid f_{lk}^Y(\cdot, \infty) > 0\} & \text{falls } l \in V, \\ \{k \in \mathcal{R}^Y \mid r_{lk}^{Y+}(\cdot, \infty) > 0\} & \text{falls } l \in V^o, \text{ bzw.} \end{cases} \quad (5.17)$$

$$\mathcal{R}^{Y-}(l) := \begin{cases} \emptyset & \text{falls } l \in V, \\ \{k \in \mathcal{R}^Y \mid r_{lk}^{Y-}(\cdot, \infty) < 0\} & \text{falls } l \in V^o, \end{cases} \quad (5.18)$$

die Mengen der kumulativen Ressourcen, die ein Vorgang  $l \in V \cup V^o$  positiv bzw. negativ nutzt. Für Rückbauvorgänge  $i \in V$  gilt stets  $\mathcal{R}^{Y-}(i) = \emptyset$ , da sie per Definition nur Stoffe freisetzen können.

In beliebigen Instanzen  $\bar{\mathcal{I}}_{\text{RCPSP}/M}$  liegt nicht notwendigerweise eine stoffstrominduzierte Zeitstruktur vor, in deren jeweiligen Zweigen die Flusserhaltung gewährleistet ist (= alle freigesetzten bzw. einlagerten Stoffeinheiten werden durch nachfolgende Stoffstromvorgänge ausgelagert). Daher ist es nicht mehr möglich, wie im vorangegangenen Abschnitt die relevanten Nachfolger mittels Durchlaufen der stoffstrominduzierten Zeitstruktur zu bestimmen. Dennoch sollen kumulative Ressourcenkonflikte weiterhin wie bisher durch zeitliche Rechtsverschiebung positiv nutzender Vorgänge aufgelöst werden. Dies ist nur möglich, wenn den positiven Nutzungen, die zum Überschreiten der Ressourcenkapazitäten führen können, ausgleichende negative Nutzungen zugeordnet werden. Übertragen auf die Vorgänge bedeutet dies, dass positiv nutzenden Vorgängen negative nutzende Vorgänge zugeordnet werden müssen. Nur dann ist eindeutig geklärt, wann welche Ausplanungsschritte und Rechtsverschiebungen durchzuführen sind, um Unzulässigkeiten aufzulösen, die während der Ausführung des Generierungsschemas auftreten können (vgl. Abschnitt 5.1).



Da es sich beim (RC)PSP/M um ein kombinatorisches Optimierungsproblem handelt, steigt die Anzahl der möglichen Zuordnungen von positiv und negativ nutzenden Vorgängen grundsätzlich exponentiell an. Es ist daher nicht möglich, in angemessener Rechenzeit über alle möglichen Zuordnungen zu enumerieren. Ein leistungsfähiges Lösungsverfahren sollte dennoch so gestaltet werden, dass es einen möglichst großen Teil des Lösungsraums absucht. Daher wäre es auch nicht angemessen, sich lediglich auf eine fixe Zuordnung von positiv und negativ nutzenden Vorgängen zu beschränken. Stattdessen ist es naheliegender, die Zuordnung basierend auf der Vorgangsfolge  $\ell$  zu bestimmen. Die Vorgangsfolge erfüllt schließlich genau den Zweck, die erwünschte Variabilität beim Durchsuchen des Lösungsraums umzusetzen: Auf Ebene der Metaheuristik wird die Vorgangsfolge iterativ abgeändert, wodurch bei jedem Aufruf des Generierungsschemas eine andere Lösung berechnet wird.

Das Konzept der relevanten Nachfolger und die Notation (vgl. Abschnitt 5.4.1) werden nachfolgend beibehalten, um die Zuordnung von negativ nutzenden Vorgängen (= relevante Nachfolger) zu positiv nutzenden Vorgängen (= relevante Vorgänger) zu formalisieren. Algorithmus 10 beschreibt die Vorgehensweise zur Berechnung aller relevanten Nachfolger für eine gegebene Instanz  $\mathcal{I}_{\text{RC}PSP/M}$  und eine Vorgangsfolge  $\ell$ . Er wird als `RelevanteNachfolgerArb` bezeichnet. Der Suffix „Arb“ steht für „arbitrary“. Die Zeilenreferenzen in diesem Abschnitt beziehen sich auf Algorithmus 10. In Zeile 1 wird zunächst für jeden Vorgang  $l \in V \cup V^o$  die Menge der relevanten Nachfolger  $Succ^-(l)$  als leere Menge initialisiert. Zudem wird in Zeile 2 für jede kumulative Ressource  $k \in \mathcal{R}^V$  eine Hilfsliste mit Speicherprinzip „First In – First Out“ erstellt. Diese Liste, bezeichnet als  $FIFO_k$ , speichert Tupel der Form  $(l^+, r^+)$ , wobei  $l^+$  ein positiv nutzender Vorgang und  $r^+$  seine positive Nutzung ist. Wenn ein solcher Tupel in der Liste enthalten ist, bedeutet dies, dass dem Vorgang  $l^+$  für  $r^+$  seiner positiven Nutzungen noch negativ nutzende Stoffstromvorgänge zugeordnet werden können.

In Zeile 3 wird über die Vorgänge der Vorgangsfolge  $\ell$  iteriert. In Zeile 4 werden alle kumulativen Ressourcen durchlaufen, die der aktuell betrachtete Vorgang  $l$  negativ nutzt, d. h.  $k \in \mathcal{R}^{V^-}(l)$ . Die negative Nutzung von Vorgang  $l$  wird in Zeile 5 ausgelesen und als Hilfsvariable  $r^-$  festgehalten. Solange  $r^- < 0$  und  $FIFO_k$  nichtleer ist (vgl. Zeile 6), kann der negativ nutzende Vorgang  $l$  noch anderen, positiv nutzenden Vorgängen als relevanter Nachfolger dienen. Dann wird in Zeile 7 der erste Tupel  $(l^+, r^+)$  aus  $FIFO_k$  ausgelesen. In Zeile 8 wird  $l$  den relevanten Nachfolgern von  $l^+$  hinzugefügt. Anschließend erfolgt eine Fallunterscheidung (vgl. Zeile 9): Falls  $r^+ + r^- \leq 0$ , werden alle verbleibenden positiven Nutzungen  $r^+$  von  $l^+$  durch  $l$  aufgehoben. Deshalb wird der betrachtete Tupel  $(l^+, r^+)$  aus  $FIFO_k$  entfernt (vgl. Zeile 10). In Zeile 11 wird die verbleibende negative Nutzung  $r^-$  entsprechend aktualisiert. Der Sonst-Block der Fallunterscheidung tritt ein, wenn nicht alle verbleibenden positiven Nutzungen  $r^+$  von  $l^+$  durch  $l$  aufgehoben werden. Es werden folglich noch weitere relevante Nachfolger für  $l^+$  gesucht. Dazu wird  $r^+$  zusammen mit dem Eintrag in  $FIFO_k$  aktualisiert (vgl. Zeile 13). In Zeile 14 wird  $r^- := 0$  gesetzt, womit die Schleife in Zeile 6 abbricht.

Nachdem alle kumulativen Ressourcen  $k \in \mathcal{R}^{V^-}(l)$  durchlaufen wurden, werden in Zeile 18 die kumulativen Ressourcen betrachtet, die  $l$  positiv nutzt. Positive Nutzungen werden erst

**Algorithmus 10:** RelevanteNachfolgerArb

---

**Eingabe:** Instanz  $\mathcal{I}_{\text{RCPSP}/M}$ ; Vorgangsfolge  $\ell$

- 1 initialisiere  $\text{Succ}^-(l) := \emptyset$  für alle  $l \in V \cup V^0$ ;
- 2 initialisiere leere Liste  $\text{FIFO}_k$  für alle  $k \in \mathcal{R}^Y$ ;
- 3 **für jeden** Vorgang  $l$  in  $\ell$  **tue**
- 4     **für jede** kumulative Ressource  $k \in \mathcal{R}^{Y^-}(l)$  **tue**
- 5         setze  $r^- := r_{lk}^{Y^-}(\cdot, \infty)$ ;
- 6         **solange**  $r^- < 0$  **und**  $\text{FIFO}_k$  nichtleer ist **tue**
- 7             lese erstes  $(l^+, r^+)$  aus  $\text{FIFO}_k$  aus;
- 8             setze  $\text{Succ}^-(l^+) := \text{Succ}^-(l^+) \cup \{l\}$  hinzu;
- 9             **wenn**  $r^+ + r^- \leq 0$  **dann**
- 10                 entferne  $(l^+, r^+)$  aus  $\text{FIFO}_k$ ;
- 11                 setze  $r^- := r^- + r^+$ ;
- 12             **sonst**
- 13                 setze  $r^+ := r^+ + r^-$  und aktualisiere  $(l^+, r^+)$  in  $\text{FIFO}_k$ ;
- 14                 setze  $r^- := 0$ ;
- 15             **Ende**
- 16         **Ende**
- 17     **Ende**
- 18     **für jede** kumulative Ressource  $k \in \mathcal{R}^{Y^+}(l)$  **tue**
- 19         füge  $(l, f_{lk}^Y(\cdot, \infty))$  bzw.  $(l, r_{lk}^{Y^+}(\cdot, \infty))$  zu  $\text{FIFO}_k$  hinzu;
- 20     **Ende**
- 21 **Ende**

---

**Ausgabe:** Mengen der relevanten Nachfolger  $\text{Succ}^-$  und Listen  $\text{FIFO} := (\text{FIFO}_k)_{k \in \mathcal{R}^Y}$

---

nach den negativen Nutzungen betrachtet, um zu verhindern, dass ein Vorgang relevanter Nachfolger von sich selbst wird. In Zeile 19 wird für jede dieser kumulativen Ressourcen  $k \in \mathcal{R}^{Y^+}(l)$  ein Eintrag  $(l, f_{lk}^Y(\cdot, \infty))$  bzw.  $(l, r_{lk}^{Y^+}(\cdot, \infty))$  in  $\text{FIFO}_k$  erzeugt (abhängig davon, ob  $l$  ein Rückbau- oder ein Stoffstromvorgang ist). Das heißt, die positive Nutzung von  $k$  durch  $l$  wird festgehalten, um in den nachfolgenden Iterationen relevante Nachfolger zu deren Ausgleich zu finden.

RelevanteNachfolgerArb terminiert, nachdem die Vorgangsfolge  $\ell$  durchlaufen wurde, und gibt die relevanten Nachfolger sowie die Listen  $\text{FIFO}_k$  mit ihren verbleibenden Einträgen zurück. Letztere werden im Generierungsschema weiterverarbeitet. Beispiel 5.11 veranschaulicht den Ablauf von RelevanteNachfolgerArb.

**Beispiel 5.11:** Eine Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  besteht aus vier realen Stoffstromvorgängen  $1^0, \dots, 4^0$  und einer kumulativen Ressource L1. Es sind keine realen Rückbauvorgänge vorhanden. Die Stoffstromvorgänge dauern  $d_{1^0} = d_{2^0} = d_{4^0} = 1$  und  $d_{3^0} = 2$ . Die kumulativen Ressourcennutzungen sind wie folgt gegeben:  $r_{1^0, L1}^{Y, \text{start}} = -2$ ,  $r_{1^0, L1}^{Y, \text{end}} = 3$ ,  $r_{2^0, L1}^{Y, \text{start}} = 2$ ,  $r_{2^0, L1}^{Y, \text{end}} = -1$ ,  $r_{3^0, L1}^{Y, \text{end}} = 1$  und  $r_{4^0, L1}^{Y, \text{start}} = -2$ . Alle übrigen kumulativen Ressourcennutzungen sind null. In Abbildung 5.13 ist für die jeweiligen Stoffstromvorgänge  $j = 1^0, \dots, 4^0$  der Verlauf der

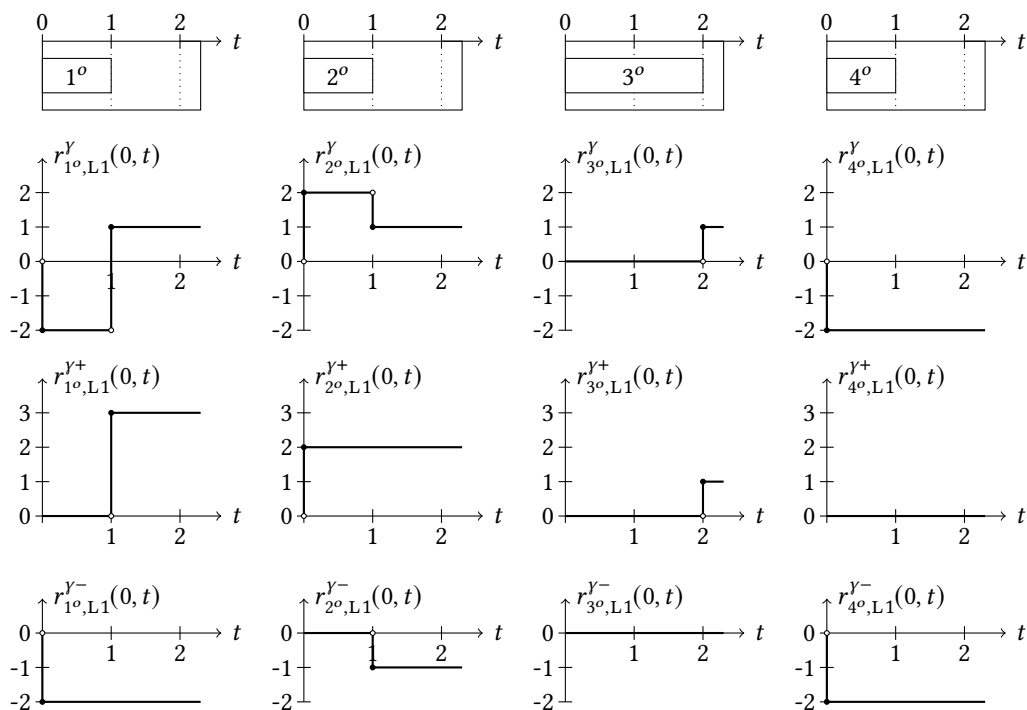
Funktionen  $r_{j,L_1}^Y(0, t)$ ,  $r_{j,L_1}^{Y^+}(0, t)$  und  $r_{j,L_1}^{Y^-}(0, t)$  abgebildet. Dabei ist deutlich zu erkennen, dass die Funktionen  $r_{j,L_1}^{Y^+}(0, t)$  und  $r_{j,L_1}^{Y^-}(0, t)$  die nach Vorzeichen gefilterten kumulativen Ressourcennutzungen darstellen.

Nun wird für diese Instanz und Vorgangsfolge  $\ell = (0, 1^o, 2^o, 3^o, 4^o, 1, 5^o)$  (dabei sind 0, 1 und 5<sup>o</sup> fiktive Vorgänge) `RelevanteNachfolgerArb` aufgerufen. In der ersten Iteration wird Vorgang 1<sup>o</sup> betrachtet: Zunächst wird  $r^- := r_{1^o,L_1}^{Y^-}(\cdot, \infty) := -2$  gesetzt. Da  $FIFO_{L_1}$  jedoch leer ist, kann 1<sup>o</sup> keinem anderen Vorgang als relevanter Nachfolger zugeordnet werden. Der Tupel  $(1^o, r_{1^o,L_1}^{Y^+}(\cdot, \infty)) = (1^o, 3)$  wird zu  $FIFO_{L_1}$  hinzugefügt. In der zweiten Iteration wird Vorgang 2<sup>o</sup> betrachtet: Zunächst wird  $r^- := r_{2^o,L_1}^{Y^-}(\cdot, \infty) := -1$  gesetzt. Aus  $FIFO_{L_1}$  wird der Tupel  $(1^o, 3)$  ausgelesen (d. h.  $r^+ = 3$ ). Vorgang 2<sup>o</sup> wird zu  $Succ^-(1^o)$  hinzugefügt. Da  $r^+ + r^- = 3 - 1 = 2 \not\leq 0$  wird  $r^+ := 2$  gesetzt und der aus  $FIFO_{L_1}$  ausgelesene Tupel auf  $(1^o, 2)$  aktualisiert. Das bedeutet, es verbleiben noch positive Nutzungen durch 1<sup>o</sup>, zu deren Ausgleich relevante Nachfolger gefunden werden sollen. Da 2<sup>o</sup> auch positive Nutzungen hat, wird Tupel  $(2^o, r_{2^o,L_1}^{Y^+}(\cdot, \infty)) = (2^o, 2)$  zu  $FIFO_{L_1}$  hinzugefügt. In der dritten Iteration wird Vorgang 3<sup>o</sup> betrachtet: Er nutzt nur positiv, weshalb lediglich Tupel  $(3^o, r_{3^o,L_1}^{Y^+}(\cdot, \infty)) = (3^o, 1)$  zu  $FIFO_{L_1}$  hinzugefügt wird. In der vierten Iteration wird Vorgang 4<sup>o</sup> betrachtet: Zunächst wird  $r^- := r_{4^o,L_1}^{Y^-}(\cdot, \infty) := -2$  gesetzt. Aus  $FIFO_{L_1}$  wird der Tupel  $(1^o, 2)$  ausgelesen (d. h.  $r^+ = 2$ ). Vorgang 4<sup>o</sup> wird zu  $Succ^-(1^o)$  hinzugefügt. Da  $r^+ + r^- = 2 - 2 = 0 \leq 0$  wird der ausgelesene Tupel aus  $FIFO_{L_1}$  entfernt. Das bedeutet, es verbleiben keine positiven Nutzungen durch 1<sup>o</sup> mehr, zu deren Ausgleich relevante Nachfolger gefunden werden sollen. Anschließend wird  $r^- := 0$  gesetzt, weshalb auch 4<sup>o</sup> nicht mehr als relevanter Nachfolger dienen kann und keine weiteren Einträge aus  $FIFO_{L_1}$  ausgelesen werden sollen. Vorgang 4<sup>o</sup> nutzt nur negativ, weshalb kein neuer Tupel zu  $FIFO_{L_1}$  hinzugefügt wird. Der Algorithmus terminiert und gibt  $Succ^-(1^o) = (2^o, 4^o)$ ,  $Succ^-(2^o) = Succ^-(3^o) = Succ^-(4^o) = \emptyset$  und  $FIFO = ((2^o, 2), (3^o, 1))$  zurück.  $\square$

Nach Ausführung von `RelevanteNachfolgerArb` ist es möglich, dass ein Stoffstromvorgang mehrere relevante Vorgänger hat. Umgekehrt bedeutet dies, dass ein Stoffstromvorgang in mehreren Mengen der relevanten Nachfolger enthalten sein kann. Stoffstromvorgänge, die weder positiv noch negativ nutzend sind, sind in keiner Menge der relevanten Nachfolger enthalten. Das heißt, die Menge der Stoffstromvorgänge wird nicht mehr wie bei den Instanzen  $\mathcal{I}_{RCPSP/M-agg}$  vollständig in disjunkte Teilmengen  $Succ^-(l)$  zerlegt (vgl. Abschnitt 5.4.1). Dies ist hier problemlos möglich, da die relevanten Nachfolger nicht mehr zum Zusammenstellen einer Stoffstromvorgangsfolge  $\mu$  verwendet werden. Stattdessen wird über die Vorgangsfolge  $\ell$  iteriert, die per Definition alle Stoffstromvorgänge enthalten muss.

## 5.5.2 Generierungsschema nach einem Integrationsansatz

Da in beliebigen Instanzen  $\mathcal{I}_{RCPSP/M}$  generell keine stoffstrominduzierte Zeitstruktur vorliegt, lassen sich auch keine Stoffstromstrukturen  $M_i$  auslesen. Und da die Stoffstromstrukturen Grundlage für den Dekompositionsansatz waren, kann dieser Ansatz hier nicht



**Abbildung 5.13:** Gantt-Diagramme und Verlauf der kumulativen Ressourcennutzungen zu Beispiel 5.11.

weiter verfolgt werden. Stattdessen lässt sich lediglich ein Generierungsschema nach einem Integrationsansatz entwickeln, womit sich dieser Abschnitt befasst. Algorithmus 11 beschreibt dieses Generierungsschema, welches als `ScheduleArb` bezeichnet wird. Die einzelnen Schritte sind teilweise identisch zu `ScheduleAgg`. Daher wird der Verfahrensablauf hier nicht mehr im Detail erläutert. Stattdessen werden lediglich die Unterschiede zu `ScheduleAgg` hervorgehoben. Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 11.

In Zeile 2 wird `RelevanteNachfolgerArb` zur Berechnung der relevanten Nachfolger aufgerufen. Die dabei zurückgelieferten Listen  $FIFO_k$  werden in Zeile 3 für alle  $k \in \mathcal{R}^Y$  durchlaufen. Der Eintrag  $r^+$  für jeden in der Liste gespeicherten Tupel  $(l, r^+)$  wird als Hilfsvariable  $ur_{lk}^{Y+}$  festgehalten. Die Bezeichnung „ur“ steht für „unassigned requirement“ und umfasst diejenige positive kumulative Ressourcennutzung durch  $l$ , zu deren Ausgleich in `RelevanteNachfolgerArb` kein relevanter Nachfolger gefunden werden konnte. In Zeile 5 wird eine als `SHIFT` bezeichnete Hilfsmenge initialisiert, deren Notwendigkeit weiter unten begründet wird.

Das Konzept der zeitzulässigen Mindestauslastung kann nach einer Anpassung auf die nach Vorzeichen gefilterten kumulativen Ressourcennutzungen weiterverwendet werden. Formal berechnet sich die zeitzulässige Mindestauslastung einer kumulativen Ressource

---

**Algorithmus 11:** ScheduleArb
 

---

**Eingabe:** beliebige Instanz  $\mathcal{I}_{\text{RCPSP/M}}$ ; Vorgangsfolge  $\ell$   
 1 initialisiere partiellen Schedule  $(S, S^o)$  mit  $S_i := -\infty$  für alle  $i \in V$  und  $S_j^o := -\infty$  für alle  $j \in V^o$ ;  
 2 berechne  $(\text{Succ}^-, \text{FIFO}) := \text{RelevanteNachfolgerArb}(\mathcal{I}_{\text{RCPSP/M}}, \ell)$ ;  
 3 initialisiere  $ur_{lk}^{y+} := r^+$  für alle Tupel  $(l, r^+)$  in  $\text{FIFO}_k$  und alle  $k \in \mathcal{R}^y$ ;  
 4 initialisiere  $ES_l := -\infty$  für alle  $l \in V \cup V^o$ ;  
 5 initialisiere  $\text{SHIFT} := \emptyset$ ;  
 6 setze  $l^* := 0$ ;  
 7 **wiederhole**  
 8     setze  $S_{l^*}$  auf das frühestmögliche  $t \geq ES_{l^*}$ , sodass (3.1), (4.1), (4.2), (4.11) eingehalten sind und Kapazität für zeitzulässige Mindestauslastung existiert;  
 9     **für jeden relevanten Vorgänger  $l^+$  von  $l^*$ , für den Restriktionen (5.21) verletzt werden tue**  
 10         setze  $\text{SHIFT} = \text{SHIFT} \cup \{l^+\}$ ;  
 11         erhöhe  $S_{l^+}$  bis Restriktionen (5.21) eingehalten sind;  
 12     **Ende**  
 13     setze  $ES_l := -\infty$  und  $S_l := -\infty$  für alle  $l \in (V \cup V^o) \setminus \text{SHIFT}$ , die in  $\ell$  nach einem  $l^+ \in \text{SHIFT}$  kommen;  
 14     setze  $ES_{l^+} := S_{l^+}$  und  $S_{l^+} := -\infty$  für alle  $l^+ \in \text{SHIFT}$ ;  
 15     setze  $\text{SHIFT} := \emptyset$ ;  
 16     **wenn ein Vorgang  $l$  in  $\ell$  mit  $S_l = -\infty$  existiert dann**  
 17         setze  $l^*$  auf den ersten Vorgang  $l$  in  $\ell$  mit  $S_l = -\infty$ ;  
 18     **Ende**  
 19 **bis**  $S_{(J+1)^o}^o \geq 0$ ;  
**Ausgabe:** zulässiger Schedule  $(S, S^o)$  für  $\mathcal{I}_{\text{RCPSP/M}}$

---

$k \in \mathcal{R}^{y+}(i)$  bzw.  $k \in \mathcal{R}^{y+}(j)$  für einen Rückbauvorgang  $i \in V$  bzw. einen Stoffstromvorgang  $j \in V^o$  zu einem Zeitpunkt  $t \geq S_i$  bzw.  $t \geq S_j^o$  zu

$$f_{ik}^y(S_i, t) + \sum_{j' \in \text{Succ}^-(i) | S_i + d_{ij'} \leq t} r_{j'k}^{y-}(S_i + d_{ij'}, t) \text{ bzw.} \quad (5.19)$$

$$r_{jk}^{y+}(S_j^o, t) + \sum_{j' \in \text{Succ}^-(j) | S_j^o + d_{jj'} \leq t} r_{j'k}^{y-}(S_j^o + d_{jj'}, t). \quad (5.20)$$

Diese zeitzulässige Mindestauslastung wird Zeile 8 berücksichtigt.

Da die Beschaffenheit beliebiger Instanzen  $\mathcal{I}_{\text{RCPSP/M}}$  unbekannt ist, wird in ScheduleArb dazu übergegangen, nach jeder Einplanung eines Vorgangs  $l^*$  (und nicht erst nach Einplanung aller relevanten Nachfolger) für jeden relevanten Vorgänger  $l^+$  von  $l^*$  die relevanten kumulativen Ressourcenrestriktionen zu prüfen (vgl. Zeile 9). Die relevanten Vorgänger werden dabei entsprechend ihrer Reihenfolge in der Vorgangsfolge  $\ell$  durchlaufen.

Die relevanten kumulativen Ressourcenrestriktionen werden neu definiert als

$$\underbrace{\sum_{i \in \mathcal{AV}(S,t)} \min\{f_{ik}^Y(S_i, t), \lambda_{ik}(S^o)\}}_{\text{I}} + \underbrace{\sum_{j \in \mathcal{AV}(S^o,t)} \min\{r_{jk}^{Y+}(S_j^o, t), \lambda_{jk}(S^o)\}}_{\text{II}}$$

$$+ \underbrace{\sum_{j \in \mathcal{AV}(S^o,t)} r_{jk}^{Y-}(S_j^o, t)}_{\text{III}} \leq R_k^Y \quad (\forall k \in \mathcal{R}^{Y+}(l^+), t = S_{l^+}, \dots, S_{l^*} + d_{l^*}) \quad (5.21)$$

mit

$$\lambda_{lk}(S^o) := \left| \sum_{j' \in \text{Succ}^-(l) | S_{j'}^o \geq 0} r_{j'k}^{Y-}(\cdot, \infty) \right| + ur_{lk}^{Y+}.$$

Summand I umfasst die Stofffreisetzungen aller Rückbauvorgänge zum Zeitpunkt  $t$ . Summand II umfasst die positiven kumulativen Ressourcennutzungen aller Stoffstromvorgänge zum Zeitpunkt  $t$ . Sowohl in Summand I als auch in Summand II führt der Minimumsausdruck in Kombination mit der Funktion  $\lambda_{lk}(S^o)$  dazu, dass positive Nutzungen nur in der Höhe berücksichtigt werden, wie sie auch durch eingeplante relevante Nachfolger negativ genutzt werden. In jedem Fall muss jedoch die Nutzung  $ur_{lk}^{Y+}$  berücksichtigt werden, da für ihren Ausgleich keine relevanten Nachfolger gefunden werden konnte. Sie wird deshalb in  $\lambda_{lk}(S^o)$  separat hinzuaddiert. Summand III umfasst die negativen kumulativen Ressourcennutzungen aller Stoffstromvorgänge zum Zeitpunkt  $t$ . Die Restriktionen (5.21) beschränken sich vergleichbar zum Integrationsansatz für aggregierte Instanzen nur auf  $k \in \mathcal{R}^{Y+}(l^+)$  und auf Zeitpunkte ab dem Start von  $l^+$  bis zum spätesten Ende von  $l^*$  (= relevanter Nachfolger).

Für einen partiellen Schedule  $(S, S^o)$  und alle Zeitpunkte  $t \geq 0$  beschreibt die linke Seite von (5.21) das kumulative Ressourcenprofil für eine kumulative Ressource  $k$ . Dieses wird in Zeile 8 herangezogen, um zu prüfen, ob die Kapazität für die zeitzulässige Mindestauslastung ausreicht.

Falls in Zeile 9 gemäß (5.21) ein Konflikt existiert, wird in Zeile 10  $l^+$  zu *SHIFT* hinzugefügt. Die Hilfsmenge *SHIFT* speichert somit alle Vorgänge, die zur Auflösung des Konflikts nach rechts verschoben werden müssen. In Zeile 11 wird  $S_{l^+}$  solange erhöht, bis die relevanten kumulativen Ressourcenrestriktionen (5.21) eingehalten sind.

In Zeile 13 werden alle Vorgänge  $l \in (V \cup V^o) \setminus \text{SHIFT}$  ausgeplant, die sich in der Vorgangsfolge  $\ell$  auf einer Position nach einem  $l^+ \in \text{SHIFT}$  befinden. Im Gegensatz zum Integrationsansatz für aggregierte Instanzen (vgl. Zeile 12 in *ScheduleAgg*) ist es hier nicht möglich, die Ausplanungen von einem Startzeitpunkt  $S_{old}$  abhängig zu machen. Der Grund dafür liegt in den relevanten Vorgänger-Nachfolger-Beziehungen. Damit die Prüfung der relevanten kumulativen Ressourcenrestriktionen in Zeile 9 korrekt funktioniert, muss stets sichergestellt sein, dass relevante Nachfolger nach ihren relevanten Vorgängern eingeplant werden. Beim erstmaligen Durchlaufen von  $\ell$  ist dies sichergestellt, da die

relevanten Vorgänger-Nachfolger-Beziehungen basierend auf  $\ell$  erstellt wurden. Somit ist ein relevanter Nachfolger in  $\ell$  immer auf einer Position hinter seinen relevanten Vorgängern. Falls in einem darauffolgenden Ausplanungsschritt ein relevanter Vorgänger ausgeplant wird, müssen auch seine relevanten Nachfolger ausgeplant werden. Eine relevante Vorgänger-Nachfolger-Beziehung begründet jedoch keine Zeitbeziehung, sondern lediglich eine Beziehung hinsichtlich der positiven und negativen Nutzungen von kumulativen Ressourcen. Das heißt, ein relevanter Nachfolger kann zu einem früheren Zeitpunkt als sein relevanter Vorgänger starten. Würde in einem solchen Fall in Zeile 13 die Ausplanung wie in `ScheduleAgg` erfolgen, könnte es vorkommen, dass relevante Vorgänger ausgeplant werden, ihre relevanten Nachfolger jedoch nicht. Daher werden pauschal alle Vorgänge ausgeplant, die sich in  $\ell$  auf einer Position nach einem  $l^+ \in SHIFT$  befinden. Vorgänge, die sich in  $\ell$  vor einem  $l^+ \in SHIFT$  befinden, werden nicht ausgeplant, um den Rechenaufwand zu verringern.

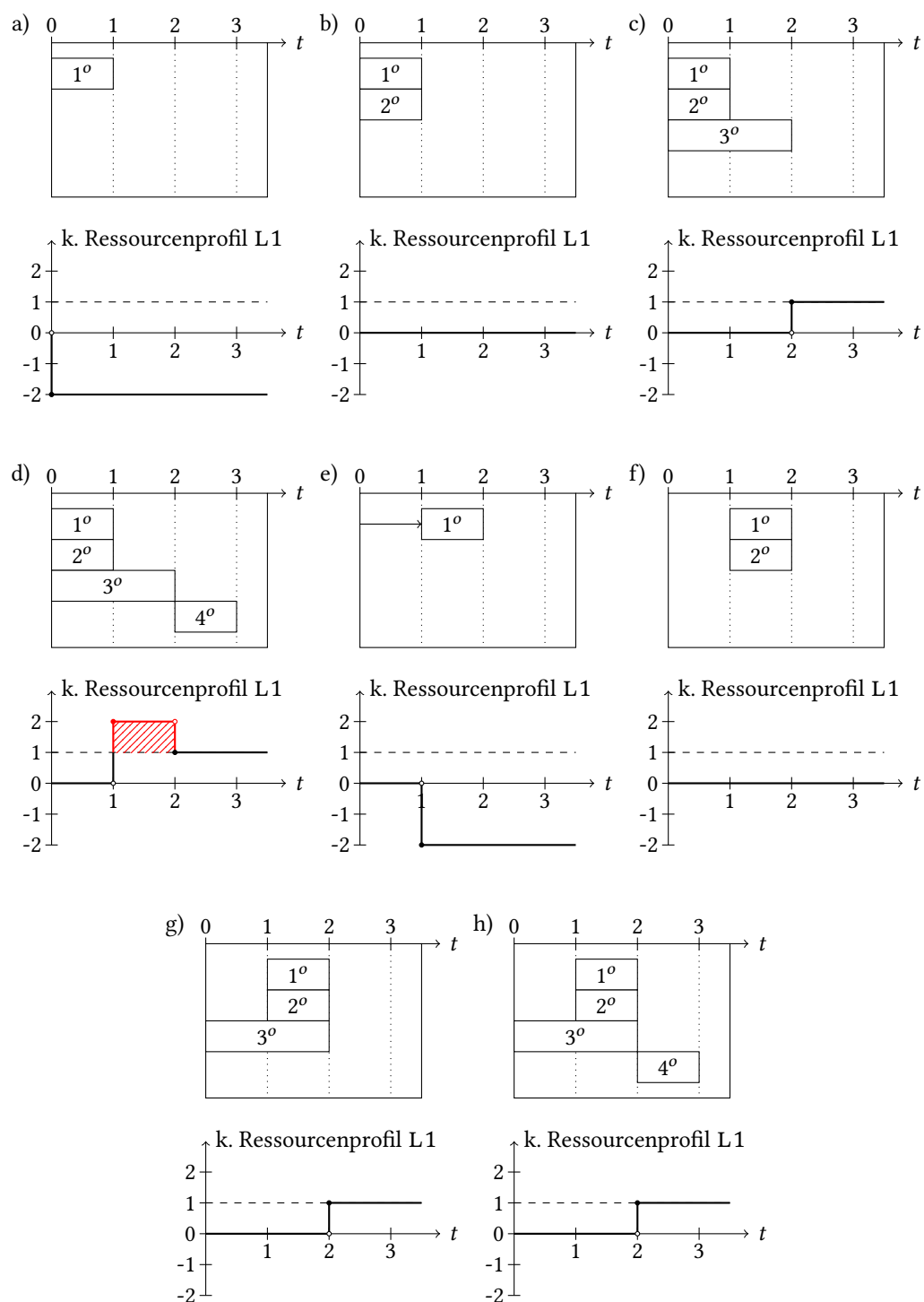
Für jeden zu verschiebenden relevanten Vorgänger  $l^+ \in SHIFT$  wird in Zeile 14 sein neu berechneter Startzeitpunkt in der Hilfsvariable  $ES_{l^+}$  gespeichert, woraufhin er ebenfalls ausgeplant wird. Diese Ausplanungen erfolgen außerhalb der Schleife von Zeile 9 bis 12, da dort die Informationen über die Startzeitpunkte noch benötigt werden. Dadurch wird die Notwendigkeit der Hilfsmenge  $SHIFT$  begründet. In Zeile 15 wird  $SHIFT$  geleert. In `ScheduleAgg` wurde  $SHIFT$  nicht benötigt, da jeder Stoffstromvorgang nur einen relevanten Vorgänger hatte und dieser daher direkt verschoben werden kann.

In Beispiel 5.12 wird der Ablauf von `ScheduleArb` anhand einer exemplarischen Instanz erläutert.

**Beispiel 5.12 (Fortsetzung des Beispiels 5.11):** Für die in Beispiel 5.11 vorgestellte Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  und die Vorgangsfolge  $\ell = (0, 1^0, 2^0, 3^0, 4^0, 1, 5^0)$  (dabei sind  $0, 1$  und  $5^0$  fiktive Vorgänge) wird nun mittels `ScheduleArb` ein zulässiger `Schedule` berechnet. Zusätzlich zu den bereits gegebenen Problemparametern wird angenommen, es existiere eine stoffstrominduzierte Zeitbeziehung  $(3^0, 4^0) \in E^{\text{flow}}$  mit zeitlichem Mindestabstand  $d_{3^0, 4^0}^{\text{flow}} = 2$ , d. h., Stoffstromvorgang  $4^0$  kann erst nach dem Ende von  $3^0$  starten. Die Kapazität der kumulativen Ressource  $L1$  beträgt  $R_{L1}^Y = 1$ .

Der in Beispiel 5.11 erläuterte Aufruf von `RelevanteNachfolgerArb` liefert  $Succ^-(1^0) = (2^0, 4^0)$ ,  $Succ^-(2^0) = Succ^-(3^0) = Succ^-(4^0) = \emptyset$  und  $FIFO = ((2^0, 2), (3^0, 1))$  an `ScheduleArb` zurück. Daher werden  $ur_{1^0, L1}^{Y+} = ur_{4^0, L1}^{Y+} := 0$ ,  $ur_{2^0, L1}^{Y+} := 2$  und  $ur_{3^0, L1}^{Y+} := 1$  initialisiert.

Tabelle 5.3 fasst die nachfolgend beschriebenen Iterationen von `ScheduleArb` zusammen. In der ersten Iteration wird der fiktive Rückbauvorgang  $0$  zum Zeitpunkt  $S_0 := 0$  eingeplant. In der zweiten Iteration wird Vorgang  $1^0$  zum Zeitpunkt  $S_{1^0}^0 := 0$  eingeplant. Gemäß Restriktionen (5.21), nach deren linker Seite sich das kumulative Ressourcenprofil berechnet, werden nur die negativen Nutzungen von  $1^0$  berücksichtigt (vgl. Abbildung 5.14a). Die positiven Nutzungen werden noch nicht berücksichtigt, da keine relevanten Nachfolger von  $1^0$  eingeplant sind und außerdem  $ur_{1^0, L1}^{Y+} = 0$ , weshalb  $\lambda_{1^0, L1}(S^0) = 0$ . Es wird nicht geprüft, ob kumulative Ressourcenkonflikte vorliegen, da  $1^0$  keine relevanten Vorgänger hat.



**Abbildung 5.14:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.12.



It.	$l^*$	$S_{j^*}^o$	Konflikte	Abb.
1	0	0	nicht geprüft, da keine relevanten Vorgänger	
2	$1^o$	0	nicht geprüft, da keine relevanten Vorgänger	5.14a
3	$2^o$	0	geprüft für $l^+ := 1^o$ : kein Konflikt in $\mathcal{R}^{Y^+}(1^o) = \{L1\}$	5.14b
4	$3^o$	0	nicht geprüft, da keine relevanten Vorgänger	5.14c
5	$4^o$	2	geprüft für $l^+ := 1^o$ : Konflikt in $\mathcal{R}^{Y^+}(1^o) = \{L1\}$ ; Ausplanung von $2^o, 3^o$ u. $4^o$ ; $ES_{1^o} := 1$ ; Ausplanung von $1^o$	5.14d
6	$1^o$	1	nicht geprüft, da keine relevanten Vorgänger	5.14e
7	$2^o$	1	geprüft für $l^+ := 1^o$ : kein Konflikt in $\mathcal{R}^{Y^+}(1^o) = \{L1\}$	5.14f
8	$3^o$	0	nicht geprüft, da keine relevanten Vorgänger	5.14g
9	$4^o$	2	geprüft für $l^+ := 1^o$ : kein Konflikt in $\mathcal{R}^{Y^+}(1^o) = \{L1\}$	5.14h
10	1	0	nicht geprüft, da keine relevanten Vorgänger	
11	$5^o$	3	nicht geprüft, da keine relevanten Vorgänger	

**Tabelle 5.3:** Iterationen in Beispiel 5.12 (It.: Iteration).

In der dritten Iteration wird Vorgang  $2^o$  zum Zeitpunkt  $S_{2^o}^o := 0$  eingeplant. Seine positiven Nutzungen werden aufgrund von  $ur_{2^o, L1}^{Y^+} = 2$  sofort im kumulativen Ressourcenprofil berücksichtigt. Da  $2^o$  ein relevanter Nachfolger von  $1^o$  ist und  $r_{2^o, L1}^{Y^-}(\cdot, \infty) = -1$ , ist nun  $\lambda_{1^o, L1}(S^o) = 1$ . Somit wird auch die positive Nutzung einer Ressourceneinheit zum Ende von  $1^o$  in (5.21) einbezogen. Aufsummiert ergibt sich ein konstantes Ressourcenprofil von null (vgl. Abbildung 5.14b). Es wird anschließend für  $l^+ := 1^o$  geprüft, ob kumulative Ressourcenkonflikte in  $\mathcal{R}^{Y^+}(1^o) = \{L1\}$  vorliegen. Das ist nicht der Fall.

In der vierten Iteration wird Vorgang  $3^o$  zum Zeitpunkt  $S_{3^o}^o := 0$  eingeplant. Seine positiven Nutzungen werden aufgrund von  $ur_{3^o, L1}^{Y^+} = 1$  sofort im kumulativen Ressourcenprofil berücksichtigt (vgl. Abbildung 5.14c). Er hat keine negativen Nutzungen und somit auch keine relevanten Vorgänger. Daher wird nicht auf kumulative Ressourcenkonflikte geprüft.

In der fünften Iteration wird Vorgang  $4^o$  aufgrund der Zeitbeziehung zum Zeitpunkt  $S_{4^o}^o := 2$  eingeplant. Er hat keine positiven Nutzungen. Da  $4^o$  ein relevanter Nachfolger von  $1^o$  ist und  $r_{4^o, L1}^{Y^-}(\cdot, \infty) = -2$ , ist nun  $\lambda_{1^o, L1}(S^o) = 3$ . Somit wird die gesamte positive Nutzung von  $1^o$  in (5.21) einbezogen. Aufsummiert ergibt sich das in Abbildung 5.14d dargestellte Ressourcenprofil. Es wird anschließend für  $l^+ := 1^o$  geprüft, ob kumulative Ressourcenkonflikte in  $\mathcal{R}^{Y^+}(1^o) = \{L1\}$  vorliegen. Das ist der Fall. Gemäß relevanter kumulativer Ressourcenrestriktionen (5.21) muss der Startzeitpunkt von  $1^o$  auf  $S_{1^o}^o = 1$  erhöht werden. Dann wird  $SHIFT := \{1^o\}$  gesetzt. Die Vorgänge  $2^o, 3^o$  und  $4^o$  werden ausgeplant, da sie sich in  $\ell$  auf einer Position nach  $1^o$  befinden. Es wird  $ES_{1^o} := 1$  festgehalten und auch Vorgang  $1^o$  ausgeplant. Ohne die über die relevanten Nachfolger formalisierte Zuordnung von negativ nutzenden Vorgängen zu positiven nutzenden Vorgängen wäre in dieser fünften Iteration nicht eindeutig bestimmt gewesen, welcher Vorgang verschoben werden muss, um den kumulativen Ressourcenkonflikt aufzuheben.

In der sechsten Iteration wird Vorgang  $1^o$  zum Zeitpunkt  $S_{1^o}^o := ES_{1^o} = 1$  eingeplant (vgl. Abbildung 5.14e). In der siebten Iteration wird Vorgang  $2^o$  ebenfalls zum Zeitpunkt  $S_{2^o}^o := 1$

eingepplant. Eine frühere Einplanung ist aufgrund der zeitzulässigen Mindestauslastung von zwei Ressourceneinheiten ab dem Start von  $2^o$  nicht möglich. Es wird anschließend für  $l^+ := 1^o$  geprüft, ob kumulative Ressourcenkonflikte in  $\mathcal{R}^{Y^+}(1^o) = \{L1\}$  vorliegen. Das ist nicht der Fall (vgl. Abbildung 5.14f), In der achten Iteration wird Vorgang  $3^o$  zum Zeitpunkt  $S_{3^o}^o := 0$  eingepplant (vgl. Abbildung 5.14g). In der neunten Iteration wird Vorgang  $4^o$  aufgrund der Vorrangbeziehung zum Zeitpunkt  $S_{4^o}^o := 2$  eingepplant. Es wird anschließend für  $l^+ := 1^o$  geprüft, ob kumulative Ressourcenkonflikte in  $\mathcal{R}^{Y^+}(1^o) = \{L1\}$  vorliegen. Das ist nicht der Fall (vgl. Abbildung 5.14h). Danach können in der zehnten und elften Iteration der fiktive Rückbauvorgang 1 und der fiktive Stoffstromvorgang  $5^o$  eingepplant werden, woraufhin `ScheduleArb` terminiert.

Wird alternativ Vorgangsfolge  $\ell = (0, 3^o, 4^o, 1^o, 2^o, 1, 5^o)$  verwendet, kann `ScheduleArb` bereits in sieben Iterationen (d. h. ohne Ausplanungsschritte) einen zulässigen `Schedule` bestimmen.  $\square$

Wie `ScheduleAgg` kann sich auch `ScheduleArb` in einer Endlosschleife aus Ein- und Ausplanungsschritten verfangen. Es ist daher sinnvoll, ein zusätzliches Kriterium zu definieren, ab welchem das Generierungsschema als fehlgeschlagen gilt (vgl. Abschnitt 5.4.3). Außerdem kann der Fall auftreten, dass in einer Iteration von `ScheduleArb` die Kapazität der kumulativen Ressourcen nicht mehr ausreicht, um den Vorgang  $l^*$  einem Zeitpunkt  $t \geq ES_{l^*}$  einzuplanen. Dann ist das Generierungsschema für die gegebene Vorgangsfolge ebenfalls fehlgeschlagen. Ein vergleichbarer Fehlschlag tritt in `ScheduleAgg` nicht auf, da in Instanzen  $\mathcal{I}_{\text{RCPSM}/M\text{-agg}}$  die kumulativen Ressourcen nach Einplanung aller relevanten Nachfolger eines Vorgangs stets leer sind. Bei beliebigen Instanzen  $\mathcal{I}_{\text{RCPSM}/M}$  ist es jedoch aufgrund eines  $ur_{lk}^{Y^+} > 0$  möglich, dass die Auslastung der kumulativen Ressource  $k$  bis zum Ende des Zeithorizonts erhalten bleibt.

Wie eingangs erwähnt, ist `ScheduleArb` auch dazu geeignet, Instanzen des (RCPSM/c) zu lösen. Es sei darauf hingewiesen, dass in `ScheduleArb` nur Höchstbestände für kumulative Ressourcen berücksichtigt werden. Eine Berücksichtigung von Mindestbeständen ist jedoch mit der in Abschnitt 4.5.3 beschriebenen Transformation der Problemparameter möglich.

## 5.6 Vergleich der Lösungsverfahren

Tabelle 5.4 zeigt einen Vergleich der vorgestellten Lösungsverfahren.

Die Generierungsschemas für granulare und aggregierte Instanzen nutzen die strukturellen Eigenschaften der Instanzen  $\mathcal{I}_{\text{RCPSM}/M\text{-gra}}$  und  $\mathcal{I}_{\text{RCPSM}/M\text{-agg}}$  aus, um möglichst effizient zulässige und gute Lösungen zu erzeugen. Sie sind, abgesehen von der Veröffentlichung in Gehring, Volk u. a. (2022), neuartig. Die Dekompositionsansätze zeichnen sich dadurch aus, dass Stoffstromstrukturen jeweils gesondert von den restlichen Vorgängen einer Instanz eingepplant werden. Das bedeutet, es werden keine Startzeitpunkte von Vorgängen außerhalb der Stoffstromstruktur verändert. Dies soll es ermöglichen, eine große Anzahl an Stoffstromvorgängen zu beherrschen. Im granularen Generierungsschema wird eine

granulare Instanzen	aggregierte Instanzen		beliebige Instanzen
Dekompositionsansatz	Dekompositionsansatz	Integrationsansatz	Integrationsansatz
Schedule wird mit einem Aufruf von SuperSchedule und $I+1$ Aufrufen von SubScheduleGra generiert	Schedule wird mit einem Aufruf von SuperSchedule und $I+1$ Aufrufen von SubScheduleAgg generiert	Schedule wird mit einem Aufruf von ScheduleAgg generiert	Schedule wird mit einem Aufruf von ScheduleArb generiert
Stoffstromvorgänge werden nach fester Stoffstromvorgangsfolge $\mu$ eingeplant $\Rightarrow$ Lösungsraum kann nicht vollständig abgesucht werden	Stoffstromvorgänge werden nach fester Stoffstromvorgangsfolge $\mu$ eingeplant $\Rightarrow$ Lösungsraum kann nicht vollständig abgesucht werden	Stoffstromvorgänge werden nach variabler Vorgangsfolge $\ell$ eingeplant $\Rightarrow$ Lösungsraum kann vollständig abgesucht werden	Stoffstromvorgänge werden nach variabler Vorgangsfolge $\ell$ eingeplant $\Rightarrow$ Lösungsraum kann vollständig abgesucht werden
Konflikte werden jeweils nach Einplanung aller Stoffstromvorgänge in $M_i$ geprüft und ggf. aufgelöst	Konflikte werden jeweils nach Einplanung aller relevanten Nachfolger geprüft und ggf. aufgelöst	Konflikte werden jeweils nach Einplanung aller relevanten Nachfolger geprüft und ggf. aufgelöst	Konflikte werden jeweils nach Einplanung eines Vorgangs geprüft und ggf. aufgelöst
zur Auflösung von Konflikten können mehrere Vorgänge gleichzeitig verschoben werden	zur Auflösung von Konflikten wird immer nur ein Vorgang verschoben	zur Auflösung von Konflikten wird immer nur ein Vorgang verschoben	zur Auflösung von Konflikten wird immer nur ein Vorgang verschoben
Ausplanungen erfolgen nur innerhalb von $M_i$	Ausplanungen erfolgen nur innerhalb von $M_i$	Ausplanungen können für alle Vorgänge erfolgen	Ausplanungen können für alle Vorgänge erfolgen

**Tabelle 5.4:** Vergleich der vorgestellten Lösungsverfahren.

zusätzliche Beschleunigung erzielt, indem kumulative Ressourcenkonflikte erst nach Einplanung aller Stoffstromvorgänge in einer Stoffstromstruktur  $M_i$  geprüft und ggf. durch Ausplanung und Verschiebung mehrerer Stoffstromvorgänge gleichzeitig aufgelöst werden. Das in der Literatur für kumulative Ressourcen existierende Generierungsschema von Schwindt, Fink u. a. (2007) (vgl. Abschnitt 5.2.3) enthält hingegen keinen Mechanismus zur Ausnutzung von Stoffstromstrukturen. Außerdem wird zur Auflösung von Konflikten je Iteration nur ein einzelner Vorgang verschoben (vgl. Schwindt, Fink u. a. 2007, S. 1349).

Das Generierungsschema für beliebige Instanzen ist nicht auf eine spezielle Instanzstruktur zugeschnitten und kann deshalb, wie bereits erwähnt, auch zur Lösung von Instanzen des (RCPSP/c) eingesetzt werden. Daher weist es Berührungspunkte zum Verfahren von Schwindt, Fink u. a. (2007) auf. So sind sowohl beim vorgestellten Verfahren als auch bei Schwindt, Fink u. a. (2007) Ausplanungsschritte von zentraler Bedeutung, um die Zulässigkeit hinsichtlich kumulativer Ressourcen herzustellen. Es handelt sich bei beiden Verfahren um prioritätsregelbasierte Verfahren. Schwindt, Fink u. a. (2007) stellen mit dem Konzept der sogenannten „capacity-driven latest start times“ sicher, dass Vorgänge mit positiven kumulativen Ressourcennutzungen vom Generierungsschema überhaupt zulässig eingeplant werden können. Dies wird in der vorliegenden Arbeit durch das Konzept der relevanten Nachfolger in Verbindung mit den relevanten kumulativen Ressourcenrestriktionen umgesetzt. In den relevanten kumulativen Ressourcenrestriktionen werden positive kumulative Ressourcennutzungen erst einberechnet, sobald entsprechende relevante Nachfolger mit negativen Nutzungen eingeplant sind. Falls es dann noch zu Überschreitungen der kumulativen Ressourcenkapazitäten kommt, wird ein Ausplanungsschritt für den relevanten Vorgänger durchgeführt. Damit ist in dieser Arbeit im Gegensatz zu Schwindt, Fink u. a. (2007) auch algorithmisch definiert, welche Vorgänge für Ausplanungsschritte herangezogen werden. Die relevanten Vorgänger-Nachfolger-Beziehungen werden basierend auf der Vorgangsfolge  $\ell$  berechnet (vgl. `RelevanteNachfolgerArb`). Die Vorgangsfolge  $\ell$  kann wiederum auf Ebene einer Metaheuristik probabilistisch abgeändert werden, wodurch die zum Absuchen des Lösungsraums erwünschte Variabilität bei den Ausplanungsschritten erzielt wird.

Bei sämtlichen vorgestellten Generierungsschemas ist zu beachten, dass der Rechenaufwand wesentlich von der Auflösung des Zeithorizonts abhängt. Insbesondere bei den Prüfungen, ob die zeitzulässige Mindestauslastung und die relevanten kumulativen Ressourcenrestriktionen eingehalten werden, ist keine effizientere Implementierung bekannt, als ein Durchlaufen aller entsprechenden Zeitpunkte. Für `ScheduleArb` beispielsweise bedeutet dies, dass bei jeder Prüfung über alle Zeitpunkte  $t = S_{l^*}, \dots, S_{l^*} + d_{l^*}$ , für die die relevanten kumulativen Ressourcenrestriktionen (5.21) definiert sind, iteriert werden muss. Eine derart rechenaufwendige Implementierung ist aufgrund der gleichmäßigen kumulativen Ressourcennutzungen über die Vorgangsausführung erforderlich. Diese Nutzungsart verursacht lineare Veränderungen des kumulativen Ressourcenprofils. Falls sich mehrere lineare Veränderungen bei paralleler Ausführung von Vorgängen überlagern, wäre es sehr aufwendig, analytisch zu bestimmen, zu welchen Zeitpunkten die kumulativen Ressourcenrestriktionen eingehalten sind. Dieser Aufwand hat auch Neumann, Schwindt und Trautmann (2005) dazu veranlasst, zu einem hybriden Verfahren überzuge-

hen und auf einen kommerziellen Solver zurückzugreifen (vgl. Abschnitt 5.2.2). Wären ausschließlich kumulative Ressourcennutzungen zum Start und zum Ende von Vorgängen erlaubt, könnte die Prüfung von kumulativen Ressourcenrestriktionen stattdessen auf die Start- und Endzeitpunkte von eingeplanten Vorgängen beschränkt werden. Eine solche Einschränkung des Modells wurde für diese Arbeit jedoch nicht gewählt, da hierzu keine hinreichende Evidenz seitens der praktischen Problemstellung vorliegt. Es ist sogar im Gegenteil eher davon auszugehen, dass in der Stoffverarbeitung bei Rückbauprojekten ein konstanter Arbeitsfortschritt zu beobachten ist, welcher die gleichmäßige Nutzung über die Vorgangsausführung sinnvoll erscheinen lässt.

Darüber hinaus wird der Rechenaufwand von der Granularität der erzeugten Stoffstromvorgänge beeinflusst. Da in den Generierungsschemas über Stoffstromvorgänge iteriert wird, ist offensichtlich, dass eine höhere Anzahl an Stoffstromvorgängen zu einem höheren Rechenaufwand führt. Die Größenordnung der Ressourcennutzungen hat hingegen keinen Einfluss auf den Rechenaufwand, da die Ressourcenrestriktionen lediglich in Additionsoperationen berücksichtigt werden. Ob beispielsweise die Auslastung einer Maschine (d. h. einer erneuerbaren Ressource) auf einer Skala von eins bis zehn oder auf einer Skala von eins bis 100 modelliert wird, hat keine Auswirkungen auf den Rechenaufwand. Dasselbe gilt für Lager (d. h. kumulative Ressourcen).

Abschließend sei angemerkt, dass das Generierungsschema für aggregierte Instanzen auch zur Lösung von granularen Instanzen verwendet werden kann, da granulare Instanzen ein Spezialfall von aggregierten Instanzen sind. Das Generierungsschema für beliebige Instanzen kann auch zur Lösung von aggregierten und granularen Instanzen verwendet werden. Diese Kompatibilitäten gelten jedoch nur theoretisch. Inwiefern Instanzen mit den jeweiligen Generierungsschemas auch praktisch lösbar sind, wird im nachfolgenden Kapitel 6 experimentell untersucht.

## 5.7 Erweiterungen der Lösungsverfahren

Anknüpfend an Abschnitt 4.5 werden nachfolgend Erweiterungen der vorgestellten Lösungsverfahren besprochen. Auf eine Unterteilung in entsprechende Unterabschnitte wird verzichtet.

Um zeitliche Höchstabstände (vgl. Abschnitt 4.5.1) zu berücksichtigen, sind die entwickelten Generierungsschemas so zu erweitern, wie bereits in Abschnitt 2.3.1 für das Generierungsschema zum (RCPSp/max) skizziert. Das bedeutet, es muss nach jeder Einplanung eines Vorgangs zusätzlich geprüft werden, ob die zeitlichen Höchstabstände eingehalten sind. Falls zeitliche Höchstabstände verletzt sind, müssen Ausplanungsschritte erfolgen, im Zuge derer die frühestmöglichen Startzeitpunkte anderer Vorgänge so weit erhöht werden, dass die zeitlichen Höchstabstände eingehalten sind. Nach einer solchen Erweiterung sind in den Generierungsschemas zum (RCPSp/M) zwei Arten von Ausplanungsschritten möglich: Ausplanungsschritte aufgrund kumulativer Ressourcenkonflikte und Auspla-

nungsschritte aufgrund verletzter zeitlicher Höchstabstände. Sowohl die kumulativen Ressourcen als auch die zeitlichen Höchstabstände induzieren eine NP-Vollständigkeit des Entscheidungsproblems zur Existenz einer zulässigen Lösung. Somit kann nicht garantiert werden kann, dass das Generierungsschema einen zulässigen Schedule findet. Analog zu den Ausplanungsschritten sind zwei Arten von Fehlschlägen möglich: Fehlschläge aufgrund kumulativer Ressourcenkonflikte und Fehlschläge aufgrund verletzter zeitlicher Höchstabstände. Da Generierungsschemas für Schedulingprobleme mit zeitlichen Höchstabständen in der Literatur etabliert sind, wird hier nicht weiter darauf eingegangen. Stattdessen wird auf Neumann, Schwindt und Zimmermann (2003, S. 85ff) oder Zimmermann u. a. (2006, S. 249ff) verwiesen.

Zur Modellierung zeitlich variabler Ressourcenkapazitäten wurde in Abschnitt 4.5.2 zwischen modellexogenen und modellendogenen zeitlichen Variabilitäten unterschieden. Bei modellexogenen Variabilitäten, d. h., wenn die Veränderungen der Ressourcenkapazitäten entscheidungsunabhängig sind, wurde vorgeschlagen, mittels fixierter Dummy-Vorgänge Ressourcenkapazitäten gezielt zu blockieren. Als Alternative dazu wurde die Erweiterung der Ressourcenkapazitäten um eine zeitliche Dimension genannt. Formal werden dazu  $R_k^\alpha$  bzw.  $R_k^\gamma$  durch  $R_{kt}^\alpha$  und  $R_{kt}^\gamma$  für alle  $k \in \mathcal{R}^\alpha$  bzw.  $k \in \mathcal{R}^\gamma$  und alle  $t \geq 0$  ersetzt. Die vorgestellten Generierungsschemas könnten ohne Weiteres mit solchen zeitabhängigen Ressourcenkapazitäten für kumulativen Ressourcen umgehen. Grund dafür ist, dass zur Prüfung kumulativer Ressourcenrestriktionen und zeitzulässiger Mindestauslastungen ohnehin über Zeitpunkte iteriert werden muss. Dabei kann statt einer konstanten Ressourcenkapazität  $R_k^\gamma$  auch eine zeitabhängige Ressourcenkapazität  $R_{kt}^\gamma$  herangezogen werden. Modellendogene Variabilitäten können gemäß Abschnitt 4.5.2 über Vorgänge abgebildet werden, wozu die Modellbestandteile des (RCPSM) ausreichen und keine Erweiterungen erforderlich sind.

Zur Berücksichtigung von Mindestbeständen (vgl. Abschnitt 4.5.3) sind die Modellbestandteile des (RCPSM) ebenfalls ausreichend, da Mindestbestände nach der Idee von Carlier und Moukrim (2015, S. 185) auch über die vorhandenen kumulativen Ressourcenrestriktionen zu Höchstbeständen abgebildet werden können. Somit ist keine Erweiterung der Generierungsschemas erforderlich.

Abschließend wird noch auf eine mögliche Erweiterung hingewiesen, die sich lediglich auf die Ausgestaltung des Generierungsschemas bezieht und sich nicht auf eine Modellerweiterung stützt. Für granulare Instanzen könnte grundsätzlich auch ein Generierungsschema nach einem Integrationsansatz entwickelt werden. Dies würde bedeuten, dass Rückbau- und granulare Stoffstromvorgänge nicht mehr auf einer über- und einer untergeordneten, sondern gleichermaßen auf einer Ebene eingeplant werden. Da alle Vorgänge dann in beliebiger (vorrangzulässiger) Folge eingeplant werden können, könnte der Lösungsraum vollständig abgesucht werden. Damit könnte dem Nachteil begegnet werden, dass SubScheduleGra aufgrund der festen, mittels Tiefensuche berechneten Stoffstromvorgangsfolge, die optimale Lösung ausschließen kann. Auf eine ausführliche Herleitung eines solchen Generierungsschemas nach einem Integrationsansatz wurde in dieser Arbeit aus zwei Gründen verzichtet:

- (1) Beim Integrationsansatz ist es möglich, dass sich ein- und auslagernde Vorgänge einer Stoffstromstruktur in der Vorgangsfolge voneinander entfernen. Dies zieht nach sich, dass zur Prüfung kumulativer Ressourcenrestriktionen über größere Zeiträume iteriert werden muss, sowie dass zusätzliche Ausplanungsschritte und größere Rechtsverschiebungen anfallen. Der dafür erforderliche Rechenaufwand kombiniert mit einer hohen Anzahl an granularen Stoffstromvorgängen lässt erwarten, dass das Generierungsschema ineffizient wird.
- (2) Die für aggregierte bzw. beliebige Instanzen entwickelten Generierungsschemas nach einem Integrationsansatz sind auch für granulare Instanzen geeignet. Damit steht mit diesen Generierungsschemas eine Möglichkeit zur Verfügung, auch für granulare Instanzen den vollständigen Lösungsraum abzusuchen.

## 5.8 Lösung mittels Constraintprogrammierung

Die *Constraintprogrammierung* ist gemäß Frühwirth und Abdennadher (2003, S. 1) ein Programmierparadigma, das sich dadurch auszeichnet, dass die Problemlösung auf die Formulierung der Restriktionen reduziert wird. Entscheidungsträger sollen sich möglichst wenig mit der Funktionsweise des Lösungsverfahrens, sondern vordergründig mit der Logik des Problems befassen (Frühwirth und Abdennadher 2003, S. 7). Die Restriktionen in diesem Zusammenhang werden dem englischsprachigen Begriff entsprechend als *Constraints* bezeichnet. Constraints beschränken sich im Vergleich zur linearen Programmierung nicht nur auf lineare Gleichungen und Ungleichungen, sondern umfassen auch prädikatenlogische Formeln (Hofstedt und Wolf 2007, S. 53ff). Das so formulierte Problem wird als *Constraintprogramm* bezeichnet. Zur Lösung eines Constraintprogramms werden (*Constraint-*)*Solver* eingesetzt, die prüfen, ob und unter welchen Belegungen der Entscheidungsvariablen die Constraints erfüllbar sind (Hofstedt und Wolf 2007, S. 53).

Gemäß IBM (2019) ist einer der Vorteile der Constraintprogrammierung, dass die Problemformulierung gleichzeitig als Problemspezifikation für den Constraintsolver dient, d. h. vorgibt, wie das betrachtete Problem definiert ist. Aufgrund dieser Übereinstimmung gibt es keine einheitliche Syntax zur Formulierung von Constraintprogrammen. Stattdessen orientiert sich die Syntax häufig an den Konzepten und Rahmenbedingungen, die der verwendete Constraintsolver vorgibt. Beispielsweise stellen Schnell und Hartl (2017, S. 3) ein Constraintprogramm für das (MRCPSp) vor, welches auf den Solver SCIP zugeschnitten ist. Gerhards (2020, S. 931) und Bofill u. a. (2022, S. 5) formulieren das Constraintprogramm für ihre Schedulingprobleme in Anlehnung an den Solver CP Optimizer. In allen in diesem Abschnitt zitierten Beiträgen werden manche Constraints auch unter Verwendung natürlichsprachiger Bezeichnungen für Funktionen, die der Solver zur Verfügung stellt, formuliert. Beispielsweise stellt der CP Optimizer zur Formulierung von Vorrangrestriktionen die Funktion `endBeforeStart` zur Verfügung, die Gerhards (2020) in seinem Constraintprogramm gleichlautend übernimmt. Hinter solchen Funktionen stehen arithmetische oder logische Bedingungen, die jedoch vorimplementiert und von der Benutzeroberfläche abgekapselt sind, sodass das Problem möglichst intuitiv formuliert werden kann.

Das in dieser Arbeit vorgestellte (RCPSP/M) lässt sich ebenfalls per Constraintprogrammierung lösen. Als Solver wird dafür der CP Optimizer gewählt. Damit wird an Arbeiten, die sich mit vergleichbaren Schedulingproblemen befassen, wie Gerhards (2020) und Boffill u. a. (2022) angeknüpft. Weitere Gründe für den CP Optimizer sind seine einfache Bedienbarkeit, die Schnittstellen zu verschiedenen Programmiersprachen, sowie die ausführliche Dokumentation. Entwickler IBM bewirbt die Eignung des Solvers für industrielle Anwendungen, da einfach zu erlernende Modellierungskonzepte und ein „robuster und automatischer“ Lösungsalgorithmus eingesetzt werden (Laborie u. a. 2018, S. 211). Außerdem wird die hohe Leistungsfähigkeit des CP Optimizers zur Lösung von Schedulingproblemen hervorgehoben, die anhand von Rechenexperimenten gezeigt werden konnte (Vilím u. a. 2015, S. 445ff). Die grundlegenden Modellierungskonzepte und Anwendungsgebiete des CP Optimizers werden anhand ausführlicher Beispiele in Laborie (2009) und Laborie u. a. (2018) vorgestellt. Laut Laborie (2009, S. 148f) basiert der im CP Optimizer verwendete Lösungsalgorithmus auf einem „Self-Adapting Large Neighbourhood Search that iteratively unfreezes and reoptimizes a selected fragment of the current solution“. Laborie u. a. (2018, S. 238ff) stellt detailliertere Informationen zur Funktionsweise des Lösungsalgorithmus bereit und verweist auf weiterführende Literatur.

Das Constraintprogramm zum (RCPSP/M) unter Verwendung der Modellierungskonzepte des CP Optimizers ist von (5.22a) bis (5.22h) angegeben.

$$\min_{act} \text{endOf}(act[I + 1]) \quad (5.22a)$$

$$\text{u. d. R. } \text{endBeforeStart}(act[i], act[i']) \quad (\forall (i, i') \in E); \quad (5.22b)$$

$$\text{endBeforeStart}(act[i], act[j], d_{ij}^{rel}) \quad (\forall (i, j) \in E^{rel}); \quad (5.22c)$$

$$\text{endBeforeStart}(act[j], act[j'], d_{jj'}^{flow}) \quad (\forall (j, j') \in E^{flow}); \quad (5.22d)$$

$$\begin{aligned} & \sum_{i \in V} \text{pulse}(act[i], r_{ik}^\alpha) \\ & + \sum_{j \in V^o} \text{pulse}(act[j], r_{jk}^\alpha) \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha); \end{aligned} \quad (5.22e)$$

$$\begin{aligned} & \sum_{i \in V} \sum_{\substack{w \in \mathcal{W} \\ k_1(w)=k}} \left( \text{stepAtStart}(act[i], f_{iw}^{y,start}) \right. \\ & \quad \left. + \text{stepAtEnd}(act[i], f_{iw}^{y,end}) \right) \\ & + \sum_{j \in V^o} \left( \text{stepAtStart}(act[j], r_{jk}^{y,start}) \right. \\ & \quad \left. + \text{stepAtEnd}(act[j], r_{jk}^{y,end}) \right) \leq R_k^Y \quad (\forall k \in \mathcal{R}^Y); \end{aligned} \quad (5.22f)$$

$$act[i] = \text{IntervalVar}(i, d_i) \quad (\forall i \in V); \quad (5.22g)$$

$$act[j] = \text{IntervalVar}(j, d_j) \quad (\forall j \in V^o). \quad (5.22h)$$



Sogenannte *Intervallvariablen* dienen der Modellierung von Rückbau- und Stoffstromvorgängen. Es handelt sich dabei um eine spezielle Art von Entscheidungsvariable mit einer zeitlichen Dimension. Das bedeutet, der Solver entscheidet über den Start- und den Endzeitpunkt einer Intervallvariable, wobei sich Start- und Endzeitpunkt um eine vorgegebene Länge (= die Vorgangsdauer) unterscheiden müssen (Laborie 2009, S. 150). Intervallvariablen werden im Constraintprogramm mit der Funktion `IntervalVar` in (5.22g) und (5.22h) unter Angabe der Vorgangsdauer definiert.

Die Vorrang-, Freisetzungs- und stoffstrominduzierten Zeitbeziehungen sind in (5.22b), (5.22c) und (5.22d) formuliert. Der Funktion `endBeforeStart` werden zwei Intervallvariablen und optional ein Zeitabstand als Argumente übergeben. Um die erneuerbaren und kumulativen Ressourcenprofile zu modellieren, bietet der CP Optimizer das Konzept der *Cumul Functions* an. Mit den Funktionen `pulse`, `stepAtStart` und `stepAtEnd` können die Veränderungen einer Cumul Function durch Intervallvariablen spezifiziert werden. Für erneuerbare Ressourcen ist die Verwendung des `pulse`-Befehls geeignet, da jeder Vorgang, wie die Bezeichnung impliziert, eine pulsartige Veränderung des Ressourcenprofils verursacht. Aufsummiert und nach oben durch die Ressourcenkapazitäten beschränkt, ergeben sich die erneuerbaren Ressourcenrestriktionen, wie in (5.22e) formuliert. Die kumulativen Ressourcenrestriktionen setzen sich analog dazu zusammen, wobei hier die Funktionen `stepAtStart` und `stepAtEnd` für die jeweiligen Freisetzungs- bzw. Nutzungsarten verwendet werden (vgl. (5.22f)).

Es ist zu beachten, dass das angegebene Constraintprogramm keine vollständig äquivalente Formulierung zum (RCPSP/M) ist, da die Freisetzungen bzw. kumulativen Ressourcennutzungen während der Vorgangsausführung  $f_{iw}^{y,lin}$  bzw.  $r_{ik}^{y,lin}$  nicht enthalten sind. Grund dafür ist, dass der CP Optimizer (in der verwendeten Version 12.9.0) neben `stepAtStart` und `stepAtEnd` keine weiteren Funktionen zur Verfügung stellt, die geeignet wären, andere Nutzungsarten von kumulativen Ressourcen zu modellieren. Gilt für eine Instanz  $\mathcal{I}_{\text{RCPSP/M}}$ , dass mindestens ein  $f_{iw}^{y,lin} > 0$  oder ein  $r_{ik}^{y,lin} \neq 0$ , kann sie mit dem angegebenen Constraintprogramm nicht äquivalent abgebildet werden. Um den CP Optimizer dennoch als Lösungsverfahren verwenden zu können, müssen die gleichmäßigen Freisetzungen bzw. Nutzungen während der Vorgangsausführung nach folgendem Schema in Freisetzungen bzw. Nutzungen zum Vorgangstart und -ende überführt werden: Freisetzungen  $f_{iw}^{y,lin} > 0$  werden für alle  $i \in V$  und  $w \in \mathcal{W}$  zu  $f_{iw}^{y,start}$  hinzuaddiert. Nutzungen  $r_{jk}^{y,lin} > 0$  werden für alle  $j \in V^o$  und  $k \in \mathcal{R}^y$  zu  $r_{jk}^{y,start}$  hinzuaddiert. Nutzungen  $r_{jk}^{y,lin} < 0$  werden für alle  $j \in V^o$  und  $k \in \mathcal{R}^y$  zu  $r_{jk}^{y,end}$  hinzuaddiert. Dieses Schema bewirkt, dass der durch gleichmäßige Freisetzungen bzw. Nutzungen während der Vorgangsausführung benötigte Lagerplatz bereits zu Beginn jedes Vorgangs blockiert und erst am Ende wieder freigegeben wird. Dadurch ist der mit dem CP Optimizer berechnete Schedule auch für die originale Instanz ein zulässiger Schedule. Folglich stellt der mit dem CP Optimizer berechnete Zielfunktionswert eine obere Schranke für den Zielfunktionswert der originalen Instanz dar. Zur besseren sprachlichen Unterscheidung wird die mit dem beschriebenen Schema umgewandelte Instanz, welche nur Freisetzungen bzw. Nutzungen zum Vorgangstart und -ende enthält als *Stufenvariante* (in Anlehnung an die Stufenform des kumulativen Ressourcenprofils)

der originalen Instanz  $\mathcal{I}_{\text{RCPSP}/M}$  bezeichnet. Beispiel 5.13 veranschaulicht die Unterschiede im kumulativen Ressourcenprofil der originalen Instanz und der Stufenvariante.

**Beispiel 5.13:** Eine Instanz  $\mathcal{I}_{\text{RMCPSP}}$  besteht aus einem realen Rückbauvorgang 1, der zehn Perioden dauert und gleichmäßig während seiner Ausführung fünf Stoffeinheiten mit einem Stoffstrompfad  $W1 = (L1, P1, s)$  freisetzt. Es gilt somit  $d_1 = 10$  und  $f_{1,W1}^{y,lin} = 5$ . Die Verarbeitung einer Stoffeinheit in P1 dauert eine Periode, d. h.  $d_{P1} = 1$ . Es existieren keine erneuerbaren Ressourcen.

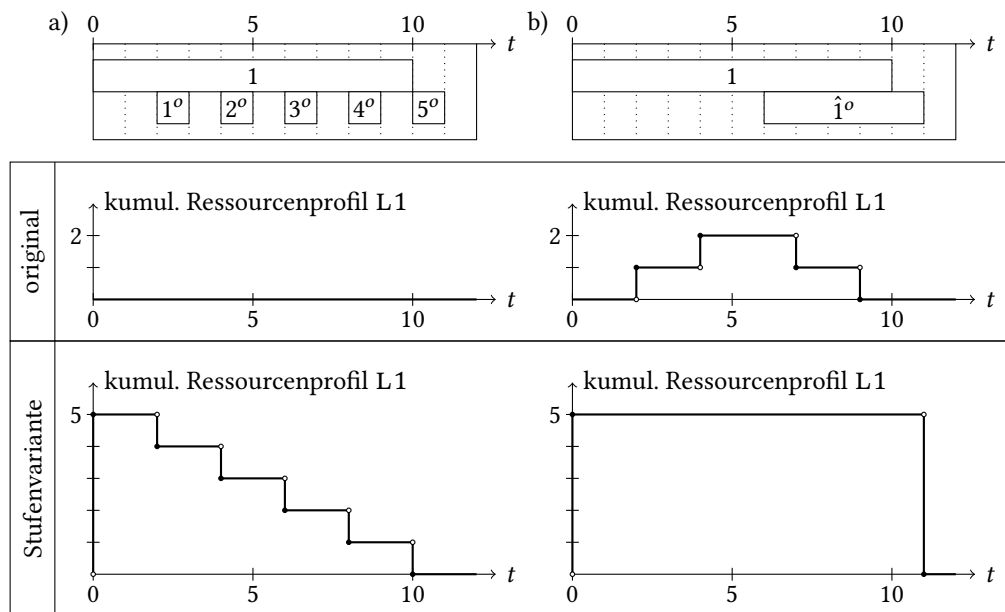
Die mit dem granularen Transformationsschema erzeugte Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  besteht aus fünf granularen Stoffstromvorgängen  $j = 1^o, \dots, 5^o$ . Jeder granulare Stoffstromvorgang  $j$  dauert eine Periode und lagert zu seinem Beginn eine Stoffeinheit aus L1 aus. Rückbauvorgang 1 setzt alle zwei Perioden eine Stoffeinheit frei, weshalb die zeitlichen Mindestabstände  $d_{1j}^{rel}$  der granularen Stoffstromvorgänge zwei, vier, sechs, acht und zehn Perioden betragen. Die mit dem aggregierten Transformationsschema erzeugte Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$  besteht aus einem aggregierten Stoffstromvorgang  $\hat{1}^o$ . Er dauert fünf Perioden und lagert gleichmäßig während seiner Ausführung fünf Stoffeinheiten aus L1 aus. Um für L1 sicherzustellen, dass ein Mindestbestand von null eingehalten wird, beträgt der zeitliche Mindestabstand  $d_{1,\hat{1}^o}^{rel} = 6$ .

Abbildung 5.15 zeigt, wie sich das kumulative Ressourcenprofil der Stufenvariante von der originalen granularen bzw. aggregierten Instanz unterscheidet. Es ist deutlich zu erkennen, dass der benötigte Lagerplatz in der Stufenvariante größer ist, da bereits zu Beginn jedes Vorgangs Lagerplatz blockiert und erst am Ende wieder freigegeben wird.  $\square$

## 5.9 Lösung mittels linearer Programmierung

Das (RCPSP/M) kann auch als gemischt-ganzzahliges lineares Programm formuliert werden. Solche Programme können mit entsprechenden Solvern, wie CPLEX, gelöst werden. Wie bei der Constraintprogrammierung dient auch hier die Problemformulierung gleichzeitig als Problemspezifikation für den Solver. In der Literatur existieren unterschiedliche Ansätze für gemischt-ganzzahlige Formulierungen von Schedulingproblemen. Diese Ansätze unterscheiden sich in der Wahl der Entscheidungsvariablen und damit auch in der Ausformulierung der Restriktionen. Für lineare Programme zum (RCPSP) wird auf Artigues u. a. (2015) verwiesen. Für lineare Programme zum (RCPSP/c) mit der Einschränkung, dass lediglich Einlagerungen zum Vorgangsende und Auslagerungen zum Vorgangstart modelliert werden, wird auf Koné u. a. (2013) verwiesen.

Eine Klasse von Ansätzen zur linearen Formulierung von Schedulingproblemen sind die *zeitindizierten Formulierungen* (Artigues u. a. 2015, S. 18). Diese Formulierungen zeichnen sich dadurch aus, dass die Entscheidungsvariablen im linearen Programm für diskrete Zeitpunkte definiert werden. Da die Anzahl der Entscheidungsvariablen endlich sein muss, wird eine obere Schranke  $T$  für die Projektdauer benötigt. Eine einfache obere Schranke



**Abbildung 5.15:** Gantt-Diagramme und kumulative Ressourcenprofile zu Beispiel 5.13 (Veranschaulichung der Unterschiede im kumulativen Ressourcenprofil der originalen Instanz und der Stufenvariante): a) granulare Instanz; b) aggregierte Instanz.

für das (RCPSP) ist die Summe der Dauern aller Vorgänge (Kolisch 2015, S. 5). Diese obere Schranke kann auch für das (RCPSP/M) verwendet werden, sofern die zeitlichen Mindestabstände  $d_{ij}^{rel}$  und  $d_{jj'}^{flow}$  nicht größer als die Dauern der jeweiligen Vorgänger  $i$  bzw.  $j$  sind. Mit gegebenem  $T$  kann der Zeithorizont auf  $t = 0, \dots, T$  beschränkt werden, wodurch auch die Anzahl der Entscheidungsvariablen beschränkt ist.

Ein Ansatz innerhalb der Klasse der zeitindizierten Formulierungen verwendet sogenannte *Puls-Startvariablen*  $x_{it} \in \{0, 1\}$ , die immer genau dann den Wert eins annehmen, wenn Rückbauvorgang  $i \in V$  zum Zeitpunkt  $t \in \{0, \dots, T\}$  startet, und sonst null sind (Artigues u. a. 2015, S. 19). Selbiges gilt für Stoffstromvorgänge, wobei dann von  $x_{jt} \in \{0, 1\}$  gesprochen wird. Die Entscheidungsvariablen  $S_i$  bzw.  $S_j^o$  aus der konzeptionellen Formulierung in Abschnitt 4.1 sind in der linearen Formulierung nicht enthalten. Die Belegungen der verschiedenen Entscheidungsvariablen können jedoch einfach ineinander transformiert werden: Wenn für ein bestimmtes  $t$  gilt, dass  $x_{it} = 1$ , wird  $S_i := t$  gesetzt und umgekehrt (analog für Stoffstromvorgänge). Die lineare Formulierung des (RCPSP/M) mit Puls-Startvariablen ist von (5.23a) bis (5.23l) angegeben.

Restriktionen (5.23b), (5.23c) und (5.23d) drücken die Vorrang-, Freisetzungs- und stoffstrominduzierten Zeitrestriktionen aus. Aufgrund des Zusammenhangs  $S_i = \sum_{t=0}^T tx_{it}$  und  $S_j^o = \sum_{t=0}^T tx_{jt}$  lässt sich leicht die Äquivalenz zu den entsprechenden Ungleichungen in der konzeptionellen Formulierung des (RCPSP/M) erkennen (vgl. Artigues u. a. 2015, S. 20). Restriktionen (5.23e) sind die erneuerbaren Ressourcenrestriktionen, welche unverändert aus Artigues u. a. (2015, S. 20) übernommen wurden. Darin wird im zweiten

$$\min_x \sum_{t=0}^T tx_{I+1,t} \quad (5.23a)$$

$$\text{u. d. R. } \sum_{t=0}^T tx_{i't} \geq \sum_{t=0}^T tx_{it} + d_i \quad (\forall (i, i') \in E); \quad (5.23b)$$

$$\sum_{t=0}^T tx_{jt} \geq \sum_{t=0}^T tx_{it} + d_{ij}^{rel} \quad (\forall (i, j) \in E^{rel}); \quad (5.23c)$$

$$\sum_{t=0}^T tx_{j't} \geq \sum_{t=0}^T tx_{jt} + d_{jj'}^{flow} \quad (\forall (j, j') \in E^{flow}); \quad (5.23d)$$

$$\begin{aligned} & \sum_{i \in V} \sum_{\tau=t-d_i+1}^t r_{ik}^\alpha x_{i\tau} \\ & + \sum_{j \in V^0} \sum_{\tau=t-d_j+1}^t r_{jk}^\alpha x_{j\tau} \leq R_k^\alpha \quad (\forall k \in \mathcal{R}^\alpha, t = 0, \dots, T); \end{aligned} \quad (5.23e)$$

$$\begin{aligned} & \sum_{i \in V} \sum_{\tau=0}^t f_{ik}^\gamma(\tau, t) x_{i\tau} \\ & + \sum_{j \in V^0} \sum_{\tau=0}^t r_{jk}^\gamma(\tau, t) x_{j\tau} \leq R_k^\gamma \quad (\forall k \in \mathcal{R}^\gamma, t = 0, \dots, T); \end{aligned} \quad (5.23f)$$

$$\sum_{t=0}^T x_{it} = 1 \quad (\forall i \in V); \quad (5.23g)$$

$$\sum_{t=0}^T x_{jt} = 1 \quad (\forall j \in V^0); \quad (5.23h)$$

$$x_{it} = 0 \quad (\forall i \in V, t = 0, \dots, ES_i - 1, LS_i + 1, \dots, T); \quad (5.23i)$$

$$x_{jt} = 0 \quad (\forall j \in V^0, t = 0, \dots, ES_j - 1, LS_j + 1, \dots, T); \quad (5.23j)$$

$$x_{it} \in \{0, 1\} \quad (\forall i \in V, t = ES_i, \dots, LS_i); \quad (5.23k)$$

$$x_{jt} \in \{0, 1\} \quad (\forall j \in V^0, t = ES_j, \dots, LS_j). \quad (5.23l)$$

Summenausdruck über alle Zeitpunkte  $\tau = t - d_i + 1, \dots, t$  aufsummiert. Es handelt sich hier um diejenigen Zeitpunkte, für die Folgendes gilt: Wenn Rückbauvorgang  $i$  zu einem dieser Zeitpunkte startet, befindet er sich zum Zeitpunkt  $t$  noch in Ausführung und nutzt daher die erneuerbare Ressource  $k$ . Mathematisch wird dies mit dem Produkt  $r_{ik}^\alpha x_{i\tau}$  ausgedrückt: Wenn  $x_{i\tau} = 1$  für ein  $\tau = t - d_i + 1, \dots, t$ , wird die erneuerbare Ressourcennutzung  $r_{ik}^\alpha$  in die Summe aufgenommen.

Nach ähnlichem Muster wie die erneuerbaren Ressourcenrestriktionen sind in (5.23f) die kumulativen Ressourcenrestriktionen formuliert. Da kumulative Ressourcennutzungen auch über den Endzeitpunkt eines Vorgangs hinaus Bestand haben, muss im zweiten Summenausdruck über alle  $\tau = 0, \dots, t$  aufsummiert werden. Falls ein Rückbauvorgang zu einem dieser Zeitpunkte startet, ist  $x_{i\tau} = 1$ . Damit wird  $f_{ik}^Y(\tau, t)$  in die Summe aufgenommen. Die Verwendung der Funktion  $f_{ik}^Y(\tau, t)$ , welche in (3.5) definiert wurde und die zu einem Zeitpunkt  $t$  durch den Rückbauvorgang  $i$  freigesetzten Stoffeinheiten angibt, vereinfacht die Formulierung. Alternativ hätte auch die hinter  $f_{ik}^Y(\tau, t)$  stehende Funktionsgleichung in die lineare Formulierung eingebaut werden können. Der Wert von  $f_{ik}^Y(\tau, t)$  ist nämlich entscheidungsunabhängig. Dies ist daran ersichtlich, dass im Funktionsargument keine Entscheidungsvariable enthalten ist. Somit ist beim Zusammenstellen des Programms für eine bestimmte Instanz  $\mathcal{I}_{\text{RCPS}/M}$  der Funktionswert von  $f_{ik}^Y(\tau, t)$  eine ganze Zahl, womit die Linearität gewährleistet ist. Analog zu Rückbauvorgängen wird für Stoffstromvorgänge der Wert der Funktion  $r_{jk}^Y(\tau, t)$  in der Formulierung verwendet. Zum besseren Verständnis der kumulativen Ressourcenrestriktionen in der linearen Formulierung wird auf Beispiel 5.14 verwiesen.

Restriktionen (5.23g) und (5.23h) stellen sicher, dass jeder Rückbau- und Stoffstromvorgang genau einmal zu einem Zeitpunkt innerhalb des Zeithorizonts startet. Restriktionen (5.23i), (5.23j), (5.23k) und (5.23l) spezifizieren den Definitionsbereich der Entscheidungsvariablen. Dazu müssen in einem vorgelagerten Schritt die frühest- bzw. spätestmöglichen Startzeitpunkte  $ES_i, ES_j$ , bzw.  $LS_i$  und  $LS_j$  für alle Rückbau- und Stoffstromvorgänge bestimmt werden. Außerhalb des Zeitraums zwischen dem frühest- und spätestmöglichen Startzeitpunkt werden die Entscheidungsvariablen  $x_{it}$  bzw.  $x_{jt}$  durch (5.23i) und (5.23j) auf null gesetzt. Damit wird der Lösungsraum verkleinert und dem Solver die Suche erleichtert. Der frühestmögliche Startzeitpunkt  $ES_i$  entspricht der Länge des längsten Pfades vom fiktiven Startvorgang zu  $i$ ,  $d_{0i}$  (vgl. Abschnitt 2.2.3). Der spätestmögliche Startzeitpunkt wird berechnet, indem von  $T$  die Länge des längsten Pfades zwischen  $i$  und dem fiktiven Endvorgang,  $d_{i,I+1}$ , abgezogen wird (Artigues u. a. 2015, S. 19).

Das angegebene lineare Programm ist eine vollständig äquivalente Formulierung zum (RCPS/M), da alle Freisetzungs- und Nutzungsarten korrekt berücksichtigt werden.

**Beispiel 5.14:** Eine Instanz  $\mathcal{I}_{\text{RCPS}/M}$  besteht unter anderem aus einem Stoffstromvorgang  $1^o$  mit Dauer  $d_{1^o} = 5$ . Außerdem ist eine kumulative Ressource L1 mit Kapazität  $R_{L1}^Y = 15$  gegeben. Stoffstromvorgang  $1^o$  lagert gleichmäßig während seiner Ausführung zehn Stoffeinheiten in L1 ein, d. h.  $r_{1^o, L1}^{Y, lin} = 10$ .

Exemplarisch wird die kumulative Ressourcenrestriktion zum Zeitpunkt  $t = 2$  betrachtet. Falls Stoffstromvorgang  $1^o$  zum Zeitpunkt  $S_{1^o}^o = 0$  startet, sind durch ihn zum Zeitpunkt  $t = 2$  gemäß Gleichung (4.12) vier Stoffeinheiten in L1 eingelagert. Falls Stoffstromvorgang  $1^o$  zum Zeitpunkt  $S_{1^o}^o = 1$  startet, sind durch ihn zum Zeitpunkt  $t = 2$  gemäß Gleichung (4.12) zwei Stoffeinheiten in L1 eingelagert. Falls Stoffstromvorgang  $1^o$  zu einem Zeitpunkt  $S_{1^o}^o \geq 2$  startet, sind durch ihn zum Zeitpunkt  $t = 2$  null Stoffeinheiten eingelagert.

Für den Zeitpunkt  $t = 2$  setzt sich der Summenausdruck  $\sum_{\tau=0}^t r_{jk}^Y(\tau, t) x_{j\tau}$  in Restriktion (5.23f) wie folgt zusammen:

$$\begin{aligned} & \sum_{\tau=0}^2 r_{1^o, L1}^Y(\tau, 3) \cdot x_{1^o, \tau} \\ &= r_{1^o, L1}^Y(0, 2) \cdot x_{1^o, 0} + r_{1^o, L1}^Y(1, 2) \cdot x_{1^o, 1} + r_{1^o, L1}^Y(2, 2) \cdot x_{1^o, 2} \\ &= \lfloor (2 - 0) \cdot 10/5 \rfloor \cdot x_{1^o, 0} + \lfloor (2 - 1) \cdot 10/5 \rfloor \cdot x_{1^o, 1} + 0 \cdot x_{1^o, 2} \\ &= 4x_{1^o, 0} + 2x_{1^o, 1} \end{aligned}$$

Werden andere Rückbau- und Stoffstromvorgänge außer Acht gelassen, besagt die kumulative Ressourcenrestriktion zum Zeitpunkt  $t = 2$ , dass  $4x_{1^o, 0} + 2x_{1^o, 1} \leq 15$ . Dies deckt sich exakt mit den oben beschriebenen Einlagerungen durch Stoffstromvorgang  $1^o$ : Falls  $x_{1^o, 0} = 1$ , sind vier Stoffeinheiten in L1 eingelagert und falls  $x_{1^o, 1} = 1$ , sind zwei Stoffeinheiten in L1 eingelagert.  $\square$

## 5.10 Überblick und Zusammenfassung

Abbildung 5.16 zeigt anknüpfend an Abbildung 4.19 einen Überblick über die Inhalte der Kapitel 3, 4 und 5. Sie fasst damit die wesentlichen Bestandteile des in dieser Arbeit entwickelten Optimierungswerkzeugs und deren Verknüpfungen zusammen. Pfeile stellen dabei Modellierungsprozesse bzw. Lösungsverfahren dar. Der Pfeil zur Constraintprogrammierung ist gestrichelt eingezeichnet, da mit dem vorgestellten Constraintprogramm nur Stufenvarianten der Instanzen  $\mathcal{I}_{\text{RCPSP/M}}$  gelöst werden können. Rechtecke stellen Probleminstanzen bzw. die Lösung dar. Der Rechtspfeil „ $\rightarrow$ “ symbolisiert einen Verweis auf den jeweiligen Abschnitt, in dem entsprechende Inhalte behandelt werden.

Zusammengefasst wurden in diesem Kapitel neuartige Lösungsverfahren entwickelt, die jeweils für bestimmte Arten von Instanzen des (RCPSP/M) geeignet sind. Kern dieser Lösungsverfahren bilden Generierungsschemas, die abhängig von einer gegebenen Prioritätsregel bzw. Vorgangsfolge einen Schedule durch iteratives Einplanen der Rückbau- und Stoffstromvorgänge erzeugen. Von existierenden Generierungsschemas in der Literatur unterscheiden sich die Generierungsschemas dieser Arbeit insbesondere durch die Fähigkeit, kumulative Ressourcenrestriktionen zu berücksichtigen. Für Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  und  $\mathcal{I}_{\text{RCPSP/M-agg}}$  wurde je ein Generierungsschema nach einem Dekompositionsansatz vorgestellt. Hier werden Rückbau- und Stoffstromvorgänge getrennt voneinander auf zwei algorithmischen Ebenen eingeplant. Dies ermöglicht die Ausnutzung der stoffstrominduzierten Zeitstruktur, um effizienter zulässige Schedules zu erzeugen. Für Instanzen

$\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  und beliebige Instanzen  $\mathcal{I}_{\text{RCPSP}/M}$  wurde außerdem je ein Generierungsschema nach einem Integrationsansatz vorgestellt, bei welchem Rückbau- und Stoffstromvorgänge gleichrangig auf derselben algorithmischen Ebene eingeplant werden. Dies bietet den Vorteil, den vollständigen Lösungsraum absuchen zu können. Insgesamt wurden somit vier Generierungsschemas vorgestellt, woraus nach einer Einbettung in eine Metaheuristik vier heuristische Lösungsverfahren resultieren. Darüber hinaus wurden die Constraintprogrammierung und die lineare Programmierung als exakte Lösungsverfahren für beliebige Instanzen  $\mathcal{I}_{\text{RCPSP}/M}$  vorgestellt. Hier ist es aufgrund kommerziell verfügbarer Solver ausreichend, lediglich die entsprechende Problemformulierung anzugeben.

Das Lösungsverfahren für granulare Instanzen wird auch in Gehring, Volk u. a. (2022) ausführlich vorgestellt. Für die darin beschriebenen Rechenexperimente wurden auch das Lösungsverfahren für aggregierte Instanzen (nach einem Dekompositionsansatz) und die Constraintprogrammierung eingesetzt. Die Funktionsweise des Lösungsverfahrens für aggregierte Instanzen wird im Beitrag jedoch nicht im Detail dargelegt und das Lösungsverfahren für beliebige Instanzen sowie die lineare Programmierung sind darin nicht enthalten.

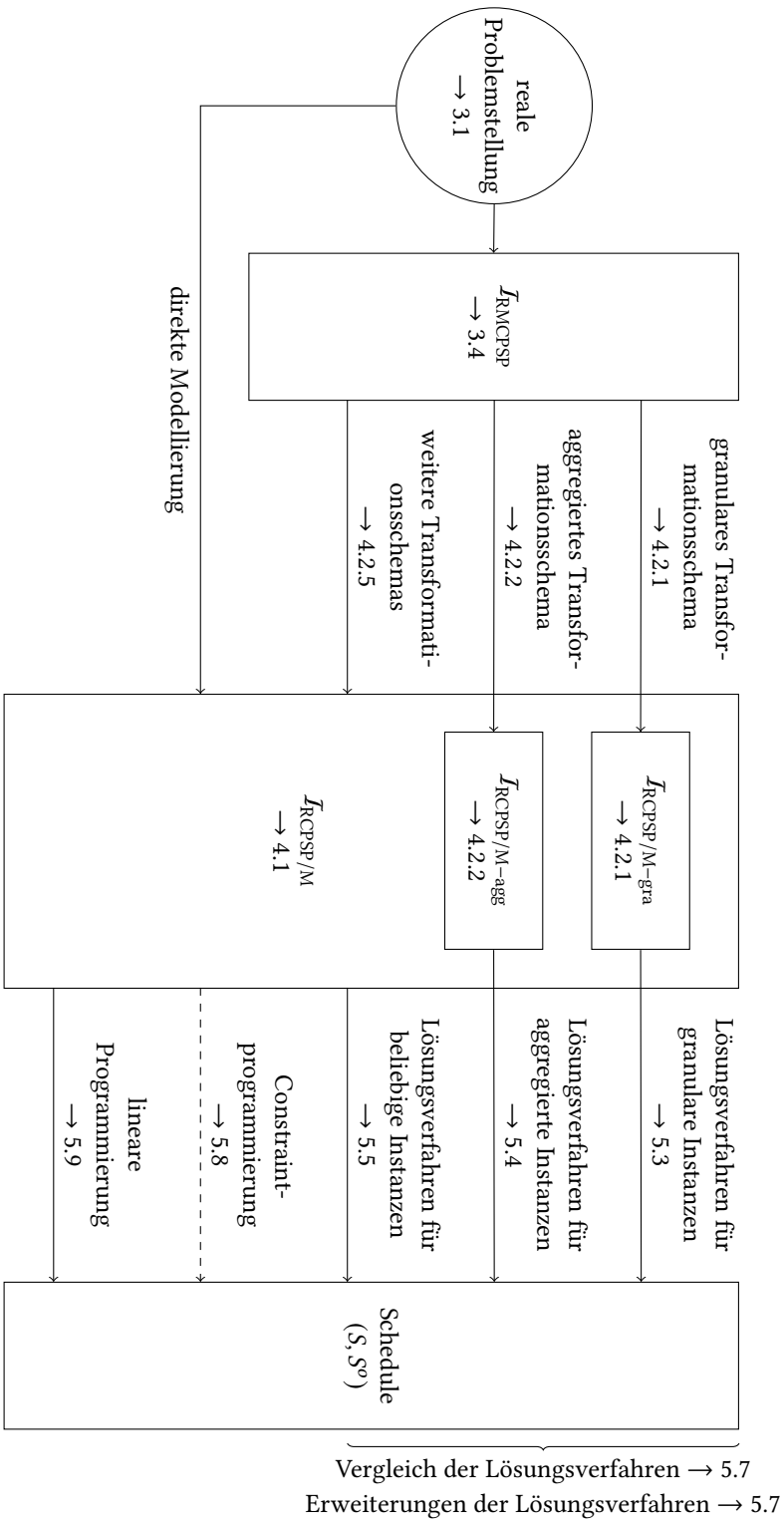


Abbildung 5.16: Überblick über die Inhalte der Kapitel 3, 4 und 5.



## 6 Rechenexperimente

Das vorliegende Kapitel behandelt Rechenexperimente, die der Anwendungsdemonstration und dem Test der vorgestellten Transformationsschemas und Lösungsverfahren dienen. Vorbereitend dazu erläutern die Abschnitte 6.1 und 6.2 die Vorgehensweise zur Erzeugung von Testinstanzen und den Testaufbau. Die darauffolgenden Abschnitte 6.3, 6.4 und 6.5 befassen sich mit der Durchführung und Auswertung der Rechenexperimente. Sie sind jeweils nach demselben Muster aufgebaut. Abschnitt 6.3 fokussiert sich darauf, zu analysieren, wie sich der Stoffstrombereich auf die mit den jeweiligen Transformationsschemas und Lösungsverfahren erzielte Lösungsqualität auswirkt. In Abschnitt 6.4 liegt der Fokus hingegen auf dem Rückbaubereich. Für die Experimente in Abschnitt 6.5 wird die Zielfunktion des (RMCPSP) abgeändert, um ihren Einfluss auf die Leistungsfähigkeit zu analysieren.

### 6.1 Testinstanzen

Zahlreiche wissenschaftliche Artikel, die sich mit Schedulingproblemen befassen, verwenden zum Testen der vorgestellten Modelle und Lösungsverfahren künstlich generierte *Testinstanzen* (z. B. Neumann und Schwindt 2002; Neumann, Schwindt und Trautmann 2005; Koné u. a. 2013; Briskorn, Jaehn u. a. 2013; Vanhoucke und Coelho 2016). Dies lässt sich mit einem Mangel an Praxisdaten oder mit Geheimhaltungspflichten im Zusammenhang mit vorhandenen Praxisdaten begründen. Allerdings gibt es auch methodische Gründe, um auf Testinstanzen zurückzugreifen: Falls die Testinstanzen einer öffentlich zugänglichen Bibliothek entstammen, ermöglichen sie es, die Effizienz verschiedener Lösungsverfahren publikationsübergreifend miteinander zu vergleichen. Solche Bibliotheken werden in der Literatur als *Benchmark Libraries* bezeichnet. Außerdem kann mit einer Menge systematisch erzeugter Testinstanzen eine größere Bandbreite an realen Problemstellungen abgedeckt werden. Wird hingegen nur ein bestimmter Praxisdatensatz verwendet, kann damit zwar gezeigt werden, dass ein entwickeltes Modell oder Lösungsverfahren für den betrachteten Fall geeignet ist. Die Ergebnisse können jedoch nur begrenzt auf andere Fälle, wie beispielsweise auf größere Projekte oder Projekte mit anderer Zeitstruktur, übertragen werden.

Auch in der vorliegenden Arbeit werden, aufgrund eingeschränkt verfügbarer Praxisdaten und um eine möglichst breite Abdeckung verschiedener Anwendungsfälle zu erreichen, Testinstanzen verwendet. Damit wird in den Rechenexperimenten die Effizienz der im vorangegangenen Kapitel vorgestellten Lösungsverfahren sowie die Auswirkung der Trans-

formationsschemas auf die Lösungsqualität untersucht. Ein Vergleich mit Ergebnissen aus anderen Publikationen ist jedoch nicht möglich, da für das neuartige (RMCPSP) keine Benchmark Library zur Verfügung steht. Daher wird ein Verfahren benötigt, mit dem systematisch neue Testinstanzen für das (RMCPSP) erzeugt werden können. Diese können dann mit den vorgestellten Transformationsschemas in Instanzen des (RCPSP/M) umgewandelt werden, so dass sie den Lösungsverfahren zugänglich sind. In Abschnitt 6.1.1 werden Parameter vorgestellt, in Abhängigkeit derer die Testinstanzen generiert werden. In Abschnitt 6.1.2 wird das Verfahren zur Erzeugung der Testinstanzen dargelegt. Um die Lösungsqualität bemessen zu können, werden in Abschnitt 6.1.3 zwei untere Schranken für den Zielfunktionswert vorgestellt. Die Inhalte dieses Abschnitts wurden teilweise bereits in Gehring, Volk u. a. (2022) veröffentlicht.

### 6.1.1 Instanzparameter

Um Testinstanzen systematisch zu erzeugen, werden zwei Gruppen sogenannter *Instanzparameter* definiert. Instanzparameter sind die Eingabe für das Verfahren zur Erzeugung der Testinstanzen. Sie unterscheiden sich damit von Problemparametern, welche die Eingabe für das Schedulingproblem sind.

Die erste Gruppe umfasst Instanzparameter, die Eigenschaften des Rückbauprojekts beschreiben:

- $I \in \mathbb{Z}_{\geq 0}$ : Die Anzahl der realen Rückbauvorgänge.
- $NC > 0$ : Die Komplexität der Vorrangstruktur, d. h. die durchschnittliche Anzahl an nichtredundanten Vorrangbeziehungen je realem oder fiktivem Rückbauvorgang (Kolisch, Sprecher und Drexl 1995, S. 1696).
- $RS \in [0, 1]$ : Ein Maß für die Knappheit der erneuerbaren Ressourcen (engl. renewable resource strength). Folgender Zusammenhang zur erneuerbaren Ressourcenkapazität gilt nach Kolisch, Sprecher und Drexl (1995, S. 1698f): Für jede erneuerbare Ressource  $k \in \mathcal{R}^\alpha$  ist

$$R_k^\alpha = r_k^{\alpha, \min} + \text{round}(RS (r_k^{\alpha, \max} - r_k^{\alpha, \min})),$$

wobei  $r_k^{\alpha, \min} := \max_{i \in V} r_{ik}^\alpha$  und  $r_k^{\alpha, \max}$  das Maximum des erneuerbaren Ressourcenprofils in einem Schedule ist, in dem alle Rückbauvorgänge zu ihren frühestmöglichen Startzeitpunkten starten. Bei  $\text{round}(x)$  handelt es sich um die Rundungsfunktion, die  $x$  auf eine ganze Zahl rundet.

- $RF \in [0, 1]$ : Das Verhältnis der Anzahl echt positiver erneuerbarer Ressourcennutzungen durch Rückbauvorgänge zur Anzahl aller Kombinationen aus Rückbauvorgängen und erneuerbaren Ressourcen (engl. renewable resource factor). Formal gilt nach Kolisch, Sprecher und Drexl (1995, S. 1697)

$$RF = \frac{1}{I \cdot |\mathcal{R}^\alpha|} \sum_{i=1, \dots, I} \sum_{k \in \mathcal{R}^\alpha} \begin{cases} 1 & \text{wenn } r_{ik}^\alpha > 0, \\ 0 & \text{sonst.} \end{cases}$$

Die zweite Gruppe umfasst Instanzparameter, die die Eigenschaften des Stoffstrombereichs beschreiben:

- $INV \in \mathbb{Z}_{\geq 0}$ : Die Lagerkapazität jedes Lagers  $k \in \mathcal{R}^Y$  (engl. inventory). Da alle Lager dieselbe Lagerkapazität  $INV$  zugewiesen bekommen, werden hier nicht die Problemparameter  $R_k^Y$  verwendet.
- $NREL \in \mathbb{Z}_{\geq 0}$ : Die Stofffreisetzung jedes realen Rückbauvorgangs  $i = 1, \dots, I$  (engl. number of released material units).
- $PREL \in [0, 1]$ : Das Verhältnis der Anzahl freisetzender Rückbauvorgänge zur Gesamtzahl an Rückbauvorgängen (engl. portion of releasing activities). Formal gilt

$$PREL = \frac{1}{I} \sum_{i=1, \dots, I} \begin{cases} 1 & \text{wenn } f_{ik}^Y(\cdot, \infty) > 0 \text{ für mindestens ein } k \in \mathcal{R}^Y, \\ 0 & \text{sonst.} \end{cases}$$

- $RFP \in [0, 1]$ : Das Verhältnis der Anzahl echt positiver erneuerbarer Ressourcennutzungen durch Verarbeitungsschritte zur Anzahl aller Kombinationen aus Verarbeitungsschritten und erneuerbaren Ressourcen (engl. resource factor of processing steps).

$$RFP = \frac{1}{|\mathcal{P}| \cdot |\mathcal{R}^\alpha|} \sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{R}^\alpha} \begin{cases} 1 & \text{wenn } r_{pk}^\alpha > 0, \\ 0 & \text{sonst.} \end{cases}$$

- $DUR \in \mathbb{R}_{\geq 0}^{|\mathcal{P}|}$ : Der Vektor der Dauern aller Verarbeitungsschritte  $p \in \mathcal{P}$  (engl. duration vector). Jede Vektorkomponente  $DUR_p$  gibt die Dauer an, die benötigt wird, um Verarbeitungsschritt  $p$  auf eine Stoffeinheit anzuwenden.

An der Verwendung der Begriffe „Lager“ und „Verarbeitungsschritt“ ist erkennbar, dass sich die Instanzparameter zur Beschreibung des Stoffstrombereichs auf der Ebene des (RMCPSP) bewegen.

### 6.1.2 Verfahren zur Erzeugung von Testinstanzen

In der Literatur gibt es bereits zahlreiche Benchmark Libraries zum (RCPSP). Da das (RMCPSP) als Erweiterung des (RCPSP) konzipiert wurde, enthält es auch sämtliche Problemparameter des (RCPSP). Konkret handelt es sich hierbei um die Menge der Rückbauvorgänge, die Dauern der Rückbauvorgänge, die Vorrangbeziehungen, die Menge der erneuerbaren Ressourcen mit den entsprechenden Ressourcenkapazitäten und die erneuerbaren Ressourcennutzungen. Zur Belegung dieser Problemparameter in einer Testinstanz  $\mathcal{I}_{\text{RMCPSP}}$  bietet es sich daher an, eine existierende Testinstanz  $\mathcal{I}_{\text{RCPSP}}$  als Quelle zu verwenden. Dies bietet den Vorteil, dass kein existierendes Verfahren zur Erzeugung dieser Problemparameter reproduziert werden muss. Außerdem wird es dadurch möglich, die in den Benchmark Libraries hinterlegten optimalen oder, falls noch keine Optimalität bewiesen werden konnte,

besten bekannten Lösungen als untere Schranke für die Testinstanzen zum (RMCPSP) zu verwenden (siehe Abschnitt 6.1.3).

Eine bekannte Benchmark Library zum (RCPSP) ist die PSPLIB (Kolisch und Sprecher 2005a). Für ausführliche Erläuterungen dazu, wie die Instanzen in dieser Library (kurz: die PSPLIB-Instanzen) erzeugt wurden, wird auf Kolisch und Sprecher (1996) verwiesen. Da die PSPLIB-Instanzen basierend auf den in Abschnitt 6.1.1 aufgelisteten Instanzparametern  $I$ ,  $NC$ ,  $RF$  und  $RS$  erzeugt wurden, eignen sie sich ideal als Quelle für die dem (RCPSP) entstammenden Problemparameter im (RMCPSP).

Die Werte der Instanzparameter sind eine Eingabe für das Verfahren zur Erzeugung von Testinstanzen. Nach dem Prinzip der Rückwärtssuche wird im ersten Verfahrensschritt jeweils diejenige PSPLIB-Instanz gewählt, die mit den gegebenen Instanzparametern erzeugt wurde. Dabei ist zu beachten, dass die Instanzparameter lediglich Werte annehmen dürfen, die bei der Erzeugung der PSPLIB verwendet wurden. Gemäß Kolisch und Sprecher (2005b) bedeutet dies:  $I \in \{30, 60, 90, 120\}$ ,  $NC = \{1,5; 1,8; 2,1\}$  und  $RF = \{0,25; 0,5; 0,75; 1\}$ . Für  $RS$  wurden keine einheitlichen Werte verwendet, sondern  $RS \in \{0,2; 0,5; 0,7; 1\}$  für  $I \in \{30, 60, 90\}$  und  $RS \in \{0,1; 0,2; \dots; 0,5\}$  für  $I = 120$ . Die PSPLIB-Instanzen sind je nach Wert des Instanzparameters  $I$  in sogenannte Sets gruppiert, die als J30-, J60-, J90- und J120-Set bezeichnet werden. Innerhalb dieser Sets sind die PSPLIB-Instanzen fortlaufend nummeriert. In einer separaten Dokumentationsdatei (Kolisch und Sprecher 2005b) ist für jede Nummer angegeben, welche Kombination der Instanzparameter  $NC$ ,  $RS$  und  $RF$  bei der Erzeugung der entsprechenden PSPLIB-Instanz verwendet wurde. Soll beispielsweise eine Testinstanz mit  $I = 30$ ,  $NC = 1,5$ ,  $RS = 0,2$  und  $RF = 0,5$  erzeugt werden, muss das J30-Set herangezogen werden. Mit Hilfe der Dokumentationsdatei wird identifiziert, dass die PSPLIB-Instanz mit der Nummer fünf zu wählen ist. Zu jeder Nummer gibt es in der PSPLIB zehn sogenannte Replikationen, d. h. Instanzen, die mit denselben Parameterwerten erzeugt wurden und sich lediglich aufgrund der probabilistischen Vorgehensweise bei der Erzeugung unterscheiden. Für diese Arbeit wird jeweils die erste Replikation gewählt. Die Problemparameter der gewählten PSPLIB-Instanz werden in eine neu erzeugte Instanz  $\mathcal{I}_{\text{RMCPSP}}$  kopiert. Vorgänge in der PSPLIB-Instanz werden dabei als Rückbauvorgänge betrachtet.

Im nächsten Schritt zur Vervollständigung der Instanz  $\mathcal{I}_{\text{RMCPSP}}$  werden acht Verarbeitungsschritte  $P1, \dots, P8$  erzeugt und zur (bisher leeren) Menge  $\mathcal{P}$  hinzugefügt, d. h.  $\mathcal{P} = \{P1, \dots, P8\}$ . Außerdem werden fünf Lager  $L1, \dots, L5$  erzeugt und zur (bisher leeren) Menge  $\mathcal{R}^Y$  hinzugefügt, d. h.  $\mathcal{R}^Y = \{L1, \dots, L5\}$ . Damit decken sich die Verarbeitungsschritte und Lager mit denen in Beispiel 3.1. Es wird davon ausgegangen, dass diese exemplarische Kombination aus acht Verarbeitungsschritten und fünf Lagern stellvertretend für einen realistischen Anwendungsfall ist. Außerdem wird damit der Test ermöglicht, ob das Planungswerkzeug mit verzweigten Stoffströmen umgehen kann. Die Kapazität jedes Lagers wird auf  $INV$  gesetzt, d. h.  $R_k^Y := INV$  für alle  $k \in \mathcal{R}^Y$ .

Jede PSPLIB-Instanz enthält vier erneuerbare Ressourcen mit gegebener Kapazität, die als  $R1, \dots, R4$  oder zusammengefasst als PSPLIB-Ressourcen bezeichnet werden. Die Nutzungen der PSPLIB-Ressourcen durch die Rückbauvorgänge sind der PSPLIB entnommen

und liegen gemäß Kolisch und Sprecher (1996) im Intervall  $\{0, \dots, 10\}$ . Für die Verarbeitungsschritte gilt initial  $r_{pk}^\alpha = 0$  für alle  $p \in \mathcal{P}$  und alle  $k \in R1, \dots, R4$ . In realen Problemstellungen ist nicht auszuschließen, dass erneuerbare Ressourcen zwischen Rückbau- und Stoffstrombereich geteilt werden (vgl. Modellanforderung (3)). Um solche Situationen auch in den Testinstanzen abzubilden, wurde der Instanzparameter  $RFP$  eingeführt. Darüber wird gesteuert, wie stark die vier PSPLIB-Ressourcen auch durch Verarbeitungsschritte genutzt werden. Bei der Erzeugung einer Testinstanz werden zufällig  $\lfloor RFP \cdot |\mathcal{P}| \cdot 4 \rfloor$  Tupel  $(p, k)$  aus allen Kombinationen aus den Verarbeitungsschritten  $P1, \dots, P8$  und den Ressourcen  $R1, \dots, R4$  ausgewählt. Für jeden ausgewählten Tupel  $(p, k)$  wird die erneuerbare Ressourcennutzung  $r_{pk}^\alpha$  zufällig mit gleichen Wahrscheinlichkeiten auf einen Wert in der Menge  $\{1, 2, 3\}$  gesetzt. Durch Verwendung dieser relativ niedrigen Werte wird sichergestellt, dass der Stoffstrombereich die Nutzungen der Ressourcen  $R1, \dots, R4$  nicht gegenüber dem Rückbaubereich dominiert. Falls  $RFP = 0$ , werden keine Tupel ausgewählt, womit die Verarbeitungsschritte dann keine der PSPLIB-Ressourcen nutzen. In einem solchen Fall teilen sich Rückbau- und Stoffstrombereich keine erneuerbaren Ressourcen.

Um die begrenzten Verarbeitungskapazitäten zu modellieren, werden sechs zusätzliche erneuerbare Ressourcen, die zur besseren Unterscheidung von den PSPLIB-Ressourcen als  $MA1, \dots, MA6$  (MA = Maschine) bezeichnet werden, erzeugt und zu  $\mathcal{R}^\alpha$  hinzugefügt. Somit ist  $\mathcal{R}^\alpha = \{R1, \dots, R4, MA1, \dots, MA6\}$ . Dann werden die Nutzungen dieser Ressourcen durch die Verarbeitungsschritte auf folgende Werte gesetzt:

$$\begin{aligned} r_{P3,MA1}^\alpha &= r_{P4,MA1}^\alpha = r_{P6,MA2}^\alpha = r_{P7,MA2}^\alpha \\ &= r_{P1,MA3}^\alpha = r_{P2,MA4}^\alpha = r_{P5,MA5}^\alpha = r_{P8,MA6}^\alpha := 1 \end{aligned} \quad (6.1)$$

Das heißt, die Verarbeitungsschritte  $P3$  und  $P4$  sowie  $P6$  und  $P7$  teilen sich paarweise eine Ressource und die übrigen Verarbeitungsschritte verwenden jeweils ihre eigene Ressource. Rückbauvorgänge nutzen keine der Ressourcen  $MA1, \dots, MA6$ . Als Kapazität wird  $R_k^\alpha := 1$  für  $k = MA1, \dots, MA6$  festgelegt. Damit stellen die Ressourcen  $MA1, \dots, MA6$  eine über alle Testinstanzen fixe, begrenzte Verarbeitungskapazität im Stoffstrombereich dar.

Im nächsten Schritt werden  $\lfloor PREL \cdot I \rfloor$  Rückbauvorgänge ausgewählt. Für jeden ausgewählten Rückbauvorgang  $i$  werden  $NREL$  zufällige Stoffstrompfade durch das in Abbildung 3.4.2 gezeigte Flussnetzwerk simuliert. Die simulierten Pfade werden zu  $\mathcal{W}$  hinzugefügt, wobei Duplikate zu entfernen sind. Für alle  $w \in \mathcal{W}$  wird  $f_{iw}^{y,lin}$  jeweils auf die Anzahl, wie oft ein Pfad  $w$  für Rückbauvorgang  $i$  simuliert wurde, gesetzt. Das bedeutet, die Stofffreisetzungen in den Testinstanzen erfolgen stets gleichmäßig über die Vorgangsausführung. Dem liegt die Annahme zugrunde, dass sich die Rückbauvorgänge durch einen konstanten Arbeitsfortschritt auszeichnen.

Algorithmus 12 fasst die Schritte des beschriebenen Verfahrens zur Erzeugung von Testinstanzen, welches als `GenerateTestInstance` bezeichnet wird, zusammen. Die Ausgabe umfasst die erzeugte Testinstanz  $\mathcal{I}_{RMCPSP}$  sowie die Dauern  $d_p$  und die erneuerbaren Ressourcennutzungen  $r_{pk}^\alpha$  für alle  $p \in \mathcal{P}$  und  $k \in \mathcal{R}^\alpha$ . Damit entspricht die Ausgabe von

GenerateTestInstance exakt der Eingabe der Transformationsschemas, womit Instanzen des (RCPSP/M) erzeugt werden können.

---

**Algorithmus 12:** GenerateTestInstance
 

---

- Eingabe:** PSPLIB; Instanzparameter  $I$ ,  $NC$ ,  $RS$ ,  $RF$ ,  $INV$ ,  $NREL$ ,  $PREL$ ,  $RFP$  und  $DUR$
- 1 erzeuge eine neue Instanz  $\mathcal{I}_{\text{RMCPSP}}$ ;
  - 2 identifiziere diejenige PSPLIB-Instanz, die mit den gegebenen Instanzparametern  $I$ ,  $NC$ ,  $RS$ ,  $RF$  erzeugt wurde;
  - 3 kopiere die Problemparameter des (RCPSP) aus der PSPLIB-Instanz in  $\mathcal{I}_{\text{RMCPSP}}$ ;
  - 4 erzeuge acht Verarbeitungsschritte  $P1, \dots, P8$  und füge sie zu  $\mathcal{P}$  hinzu;
  - 5 erzeuge fünf Lager  $L1, \dots, L5$  und füge sie zu  $\mathcal{R}^Y$  hinzu;
  - 6 setze die Lagerkapazität  $R_k^Y := INV$  für alle  $k \in \mathcal{R}^Y$ ;
  - 7 setze die Nutzungen der PSPLIB-Ressourcen  $R1, \dots, R4$  in Abhängigkeit von  $RFP$  auf einen Wert aus der Menge  $\{1, 2, 3\}$ ;
  - 8 erzeuge sechs erneuerbare Ressourcen  $MA1, \dots, MA6$  und füge sie zu  $\mathcal{R}^\alpha$  hinzu;
  - 9 setze die erneuerbare Ressourcenkapazität  $R_k^\alpha := 1$  für alle  $k = MA1, \dots, MA6$ ;
  - 10 setze die Nutzungen von  $MA1, \dots, MA6$  durch die Verarbeitungsschritte wie in Gleichung (6.1) gegeben;
  - 11 wähle  $\lfloor PREL \cdot I \rfloor$  zufällige Rückbauvorgänge und simuliere  $NREL$  Pfade durch das Flussnetzwerk;
  - 12 setze für jeden gewählten Rückbauvorgang seine Freisetzungen  $f_{iw}^{Y,lin}$  auf die Anzahl, wie oft ein Pfad simuliert wurde;
- Ausgabe:** Instanz  $\mathcal{I}_{\text{RMCPSP}}$ ; Dauern  $d_p$  ( $= DUR_p$ ) und erneuerbare Ressourcennutzungen  $r_{pk}^\alpha$  für alle  $p \in \mathcal{P}$  und  $k \in \mathcal{R}^\alpha$
- 

Das Ausmaß, in welchem die Engpässe im Stoffstrombereich einen Rückkopplungseffekt auf das Rückbauprojekt haben, ergibt sich aus dem Zusammenspiel aller stoffstrombezogenen Problemparameter in einer Instanz  $\mathcal{I}_{\text{RMCPSP}}$ . Werden die Problemparameter wie mit GenerateTestInstance künstlich generiert, bestehen zahlreiche Freiheitsgrade. Beispielsweise kann die Verarbeitungskapazität in einer Testinstanz über eine Erhöhung der erneuerbaren Ressourcenkapazitäten erhöht werden. Eine Verringerung der Verarbeitungsdauern führt zu einem analogen Effekt. Das vorgestellte Verfahren zur Erzeugung von Testinstanzen wurde gezielt darauf ausgelegt, diese Freiheitsgrade zu reduzieren. Dadurch lassen sich die Einflüsse einzelner Instanzparameter in den Rechenexperimenten besser untersuchen: Es ist sichergestellt, dass keine anderen Instanzparameter einen gegenläufigen Einfluss ausüben. Auf dieser Argumentation basierend wurden folgende Entscheidungen getroffen:

1. Der Instanzparameter  $NREL$  wurde gezielt als Skalar definiert, welcher für alle Rückbauvorgänge eine deterministische und gleich große Stofffreisetzung vorgibt. Formal gilt in den erzeugten Testinstanzen  $\sum_{w \in \mathcal{W}} f_{iw}^{Y,lin} \in \{NREL, 0\}$  für alle  $i = 1, \dots, I$ . Ob Stoffe gleichmäßig von allen Rückbauvorgängen oder überwiegend punktuell von wenigen Rückbauvorgängen freigesetzt werden, wird ausschließlich über  $PREL$  gesteuert. Ein hoher Wert für  $PREL$  beschreibt eine hohe Wahrscheinlichkeit, mit

der ein Rückbauvorgang Stoffe freisetzt, woraus eine eher gleichmäßige Verteilung der Stofffreisetzungen über alle Rückbauvorgänge resultiert, und umgekehrt.

2. Der Instanzparameter  $INV$  wurde gezielt als Skalar definiert, welcher für alle Lager eine deterministische und gleich große Lagerkapazität vorgibt. Somit wird die im Rückbauprojekt verfügbare Lagerkapazität über  $INV$  gesteuert. Inwiefern einzelne Lager einen Engpass bei der Stoffverarbeitung darstellen, wird ausschließlich über  $DUR$  gesteuert. Da  $DUR$  ein Vektor ist, ermöglicht er es, für jeden Verarbeitungsschritt eine individuelle Verarbeitungsdauer festzulegen. Die Verarbeitungsdauern bestimmen, wie schnell die Lager entleert werden, aus denen die zugehörigen Verarbeitungsschritte ihre Stoffe beziehen. Damit bestimmen sie auch, inwiefern die Lager einen Engpass darstellen.
3. Die erneuerbaren Ressourcen  $MA1, \dots, MA6$  mit ihren Kapazitäten sowie ihre Nutzungen durch die Verarbeitungsschritte wurden gezielt deterministisch und über alle Testinstanzen unveränderlich vorgegeben. Dadurch werden die verfügbaren Verarbeitungskapazitäten ausschließlich über den Instanzparameter  $DUR$  gesteuert.

### 6.1.3 Untere Schranken

Wie bereits erläutert, wurde die Benchmark Library PSPLIB unter anderem als Grundlage zur Erzeugung von Testinstanzen für das (RMCPSP) verwendet, da sich damit die Projektdauern der besten bekannten Lösungen aus der PSPLIB als untere Schranken für die Zielfunktionswerte des (RMCPSP) betrachten lassen. Dies ist möglich, da im (RMCPSP) auch sämtliche Restriktionen des (RCPS) enthalten sind, für welches die PSPLIB-Instanzen generiert wurden. Dadurch kann die optimale Lösung des (RMCPSP) zu einer Instanz  $I_{\text{RMCPSP}}$  nie besser als die optimale Lösung des (RCPS) zur selben Instanz werden. Es sei angemerkt, dass eine beste bekannte Lösung aus der PSPLIB strenggenommen keine korrekte untere Schranke ist, da ihre Optimalität nicht bewiesen wurde. Theoretisch könnte daher eine noch bessere Lösung berechnet werden. Da die besten bekannten Lösungen in der PSPLIB allerdings mit hohem Rechenaufwand und einer Vielzahl unterschiedlicher Verfahren berechnet wurden, wird davon ausgegangen, dass sie durch die Lösungsverfahren dieser Arbeit nicht verbessert werden können. Daher wird angenommen, dass sie eine untere Schranke bilden. Für PSPLIB-Instanzen des J30-Sets sind alle optimalen Lösungen bekannt, für die übrigen Sets jedoch nicht. Der beste bekannte Zielfunktionswert der zugehörigen PSPLIB-Instanz wird formal als untere Schranke  $LB_{\text{PSPLIB}}$  für den Zielfunktionswert einer Testinstanz  $I_{\text{RMCPSP}}$  bezeichnet.

Eine große Anzahl an freigesetzten Stoffeinheiten und dadurch auftretende Engpässe im Stoffstrombereich können das Rückbauprojekt aufgrund des Rückkopplungseffekts stark verzögern. Da  $LB_{\text{PSPLIB}}$  nur auf den erneuerbaren Ressourcennutzungen und Vorrangbeziehungen im Rückbaubereich basiert, würde dann die Projektdauer auch stark von  $LB_{\text{PSPLIB}}$  abweichen. Um auch in solchen Fällen eine möglichst eng bindende untere Schranke berechnen zu können, wird nachfolgend die stoffstrombasierte untere Schranke  $LB_{\text{material}}$

hergeleitet. Sie ist auf `GenerateTestInstance` zugeschnitten und kann daher auch nur für Testinstanzen berechnet werden, die damit erzeugt wurden.

Die Verarbeitungsschritte  $P1, \dots, P8$  nutzen bei ihrer Ausführung jeweils eine Einheit einer erneuerbaren Ressource  $k \in \{MA1, \dots, MA6\}$ . Diese Nutzungen sind deterministisch und über alle Testinstanzen fix vorgegeben. Die Kapazitäten von  $MA1, \dots, MA6$  wurden ebenfalls deterministisch und fix auf eins festgelegt. Die Dauern für die Verarbeitung einer Stoffeinheit in den jeweiligen Verarbeitungsschritten werden über den Instanzparameter  $DUR$  vorgegeben. Somit bestimmen die Kapazitäten und Nutzungen der Ressourcen  $MA1, \dots, MA6$  im Zusammenspiel mit den Dauern  $DUR$  eine Verarbeitungsrate für die Stoffeinheiten. Basierend auf dieser Verarbeitungsrate wird  $LB_{material}$  berechnet. Zunächst wird exemplarisch Ressource  $MA5$  betrachtet. Sie wird ausschließlich von Verarbeitungsschritt  $P5$  genutzt. Aufgrund ihrer Kapazität  $R_{MA5}^\alpha = 1$  können Stoffeinheiten von  $P5$  nicht parallel verarbeitet werden. Sei  $units(p)$  die Anzahl der Stoffeinheiten, die Verarbeitungsschritt  $p$  während des gesamten Rückbauprojekts durchlaufen. Dann ergibt sich die gesamte Dauer zur Verarbeitung aller Stoffeinheiten in  $P5$  zu  $units(P5) \cdot DUR_{P5}$ . Diese Dauer hat keinen direkten Einfluss auf die Projektdauer, da Lager zur Pufferung der Stoffeinheiten vorhanden sein können. Dem Verarbeitungsschritt  $P5$  sind im Flussnetzwerk in Abbildung 3.5 die Lager  $L1, L3, L4$  und  $L5$  vorgeschaltet. Die Lagerkapazität zur Pufferung aller Stoffeinheiten, die  $P5$  durchlaufen, entspricht daher maximal  $R_{L1}^Y + R_{L3}^Y + R_{L4}^Y + R_{L5}^Y$ . Umgekehrt müssen mindestens  $units(P5) - R_{L1}^Y - R_{L3}^Y - R_{L4}^Y - R_{L5}^Y$  Stoffeinheiten ohne Pufferung in  $P5$  verarbeitet werden, d. h. synchron zu ihrer Freisetzung. Die gesamte Dauer zur Verarbeitung dieser Stoffeinheiten ergibt sich zu

$$\left( units(P5) - R_{L1}^Y - R_{L3}^Y - R_{L4}^Y - R_{L5}^Y \right) \cdot DUR_{P5}.$$

Während dieses Zeitraums können Stoffeinheiten durch die Rückbauvorgänge nur genauso schnell freigesetzt werden, wie die Verarbeitung in  $P5$  erfolgt. Andernfalls würde die Lagerkapazität überschritten werden. Um daraus eine untere Schranke für die Projektdauer abzuleiten, muss noch eine Verarbeitungsdauer  $DUR_{P5}$  abgezogen werden. Grund dafür ist, dass Stoffeinheiten jeweils zu Beginn ihrer Verarbeitung aus dem Lager genommen werden. Formal entspricht daher

$$\begin{aligned} & \left( units(P5) - R_{L1}^Y - R_{L3}^Y - R_{L4}^Y - R_{L5}^Y \right) \cdot DUR_{P5} - DUR_{P5} \\ & \equiv \left( units(P5) - R_{L1}^Y - R_{L3}^Y - R_{L4}^Y - R_{L5}^Y - 1 \right) \cdot DUR_{P5} \end{aligned}$$

demjenigen Zeitpunkt, zu dem die letzte, nicht gepufferte Stoffeinheit freigesetzt werden darf. Dies ist eine untere Schranke für die Projektdauer.

Nach derselben Logik induzieren alle erneuerbaren Ressourcen  $MA1, \dots, MA6$  eine untere Schranke. Dabei ist zu beachten, dass sich die Verarbeitungsschritte  $P3$  und  $P4$  sowie  $P6$



und P7 paarweise eine erneuerbare Ressource teilen. Daher wird die folgende generelle Formel zur Bestimmung von  $LB_{material}$  verwendet:

$$LB_{material} = \max_{\substack{k=MA1, \\ \dots, MA6}} \left( \left( \sum_{p \in \mathcal{P} | r_{pk}^\alpha = 1} \text{units}(p) - \sum_{k \in \bigcup_{p \in \mathcal{P} | r_{pk}^\alpha = 1} \text{upstream}(p)} R_k^Y - 1 \right) \cdot \min_{p \in \mathcal{P} | r_{pk}^\alpha = 1} DUR_p \right) \quad (6.2)$$

Diese Formel setzt sich nach demselben Prinzip zusammen, wie oben exemplarisch für MA5 beschrieben. Allerdings wird mit  $p \in \mathcal{P} | r_{pk}^\alpha = 1$  im ersten Summenausdruck über alle Verarbeitungsschritte aufsummiert, die die erneuerbare Ressource  $k$  nutzen. Die Funktion  $\text{upstream}(p)$  gibt die Menge aller einem Verarbeitungsschritt  $p \in \mathcal{P}$  vorgeschalteten Lager zurück (vgl. Abbildung 3.5). Beispielsweise ist  $\text{upstream}(P5) = \{L1, L3, L4, L5\}$ . Für alle Verarbeitungsschritte  $p \in \mathcal{P} | r_{pk}^\alpha = 1$  wird die Vereinigungsmenge von  $\text{upstream}(p)$  gebildet. Über die zur Vereinigungsmenge gehörenden Lagerkapazitäten wird aufsummiert. Dies ergibt die gesamte Lagerkapazität zur Pufferung der Stoffeinheiten, die die Verarbeitungsschritte  $p \in \mathcal{P} | r_{pk}^\alpha = 1$  durchlaufen. Die Ausführungsreihenfolge der Verarbeitungsschritte  $p \in \mathcal{P} | r_{pk}^\alpha = 1$  ist bei der Bestimmung der unteren Schranke unbekannt. Das heißt, es ist auch unbekannt, welche Stoffeinheiten synchron verarbeitet und welche gepuffert werden. Für eine untere Schranke muss daher die Dauer nach unten abgeschätzt werden. Das bedeutet, es wird angenommen, dass alle Stoffeinheiten im schnellsten Verarbeitungsschritt  $p \in \mathcal{P} | r_{pk}^\alpha = 1$  verarbeitet werden. Deshalb wird in Formel (6.2) mit  $\min_{p \in \mathcal{P} | r_{pk}^\alpha = 1} DUR_p$  multipliziert. Es sei angemerkt, dass  $LB_{material}$  negativ wird, wenn für jede Ressource MA1, . . . , MA6 die Lagerkapazität zur Pufferung größer als die Anzahl der verarbeiteten Stoffeinheiten ist. Dann wird  $LB_{material}$  auf null gesetzt.

Formel (6.2) funktioniert nur, da es gemäß Abbildung 3.5 nicht möglich ist, dass eine Stoffeinheit zwei- oder mehrmals einen Verarbeitungsschritt durchläuft, der dieselbe Ressource MA1, . . . , MA6 nutzt. Konkret können Stoffeinheiten nur entweder P3 oder P4 bzw. entweder P6 oder P7 durchlaufen. Wenn dies nicht der Fall wäre, müsste der erste Summand in Formel (6.2) angepasst werden, um keine Stoffeinheit mehrfach zu zählen.

## 6.2 Testaufbau

In diesem Abschnitt wird erläutert, wie die durchgeführten Rechenexperimente zum Testen der Transformationsschemas und Lösungsverfahren aufgebaut sind. In Abschnitt 6.2.1 werden die beiden verwendeten Metaheuristiken vorgestellt. Abschnitt 6.2.2 zeigt auf, welche Testinstanzen mit welchen Instanzparametern erzeugt wurden. In Abschnitt 6.2.3 werden die verwendeten Lösungsstrategien vorgestellt. Unter dem Begriff der Lösungsstrategie wird eine bestimmte Festlegung aller variablen Bestandteile eines Rechenexperiments verstanden.

### 6.2.1 Verwendete Metaheuristiken

Um vollwertige Lösungsverfahren zu erhalten, müssen die in den Abschnitten 5.3, 5.4 und 5.5 entwickelten Generierungsschemas in Metaheuristiken eingebettet werden. Für die Rechenexperimente in dieser Arbeit werden zwei Metaheuristiken verwendet: Eine an Hübner u. a. (2021) angelehnte Multi-Start-Metaheuristik und eine Simulated-Annealing-Metaheuristik nach Bouleimen und Lecocq (2003). Beide Metaheuristiken werden im Folgenden kurz erläutert.

Hübner u. a. (2021) stellen ein Lösungsverfahren für eine spezielle Variante eines Ressourceninvestitionsproblems vor, welches gezielt auf die Lösung großer Probleminstanzen ausgelegt wurde. Es läuft in zwei Phasen ab, deren anteilige Rechenzeit von außen vorgegeben werden muss. Die erste Phase besteht aus einer *Multi-Start-Local-Search-Metaheuristik* (MLS-Metaheuristik), die zur Erzeugung eines zulässigen Schedules auf ein problemspezifisches paralleles Generierungsschema zurückgreift. In der zweiten Phase läuft ein sogenanntes Adaptive-Large-Neighbourhood-Search-Verfahren, welches versucht, die durch die MLS-Metaheuristik gefundene Lösung unter Verwendung eines Solvers für gemischt-ganzzahlige lineare Programme zu verbessern. Die MLS-Metaheuristik kann auch in dieser Arbeit eingesetzt werden, sofern das parallele Generierungsschema in Hübner u. a. (2021) durch ein Generierungsschema für das (RCPSP/M) ersetzt wird. Der Local-Search-Teil der MLS-Metaheuristik besteht darin, für eine betrachtete Vorgangsfolge alle direkt aufeinanderfolgenden Paare von Vorgängen zu durchlaufen und jeweils, falls vorrangzulässig möglich, zu vertauschen. Für jeden Tausch wird mit dem Generierungsschema ein neuer Schedule berechnet. Hierfür wird ein Großteil der Rechenzeit innerhalb der MLS-Metaheuristik verwendet. Daher ist der Local-Search-Teil nur sinnvoll, wenn entweder sehr viel Rechenzeit zur Verfügung steht oder die einzelnen Durchläufe der Generierungsschemas vergleichsweise wenig Rechenzeit in Anspruch nehmen. Ansonsten dominiert der Local-Search-Teil die Lösungssuche, d. h., der Lösungsraum wird nur lokal abgesucht. Generierungsschemas für das (RCPSP/M) sind komplexer als das parallele Generierungsschema in Hübner u. a. (2021), da in den Generierungsschemas für das (RCPSP/M) der Rechenaufwand mit der Auflösung des Zeithorizonts skaliert und umfangreichere Ausplanungsschritte erforderlich sein können. Daher ist zu erwarten, dass sie mehr Rechenzeit in Anspruch nehmen. Aus diesem Grund wird der Local-Search-Teil der MLS-Metaheuristik nach Hübner u. a. (2021) für die Rechenexperimente dieser Arbeit nicht übernommen.

Ohne den Local-Search-Teil verbleibt von der MLS-Metaheuristik nach Hübner u. a. (2021) nur der Multi-Start-Teil. Dieser Teil bildet die Multi-Start-Metaheuristik, welche in Algorithmus 13 formal beschrieben ist und als `MultiStart` bezeichnet wird. Eine Instanz  $\mathcal{I}_{\text{RCPSP/M}}$ , eine Prioritätsregel  $\pi$ , ein Zeitlimit  $timeLimit$ , ein Faktor  $perturbFactor \in [0, 1]$  und ein für die Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  geeignetes Generierungsschema bilden die Eingabe für `MultiStart`. Die Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  ist die Zusammenfassung aller Problemparameter des (RCPSP/M). Bei den weiteren Eingabedaten, wie  $timeLimit$  oder  $perturbFactor$ , handelt es sich um *Verfahrensparameter*. Im Gegensatz zu Problemparametern sind sie spezifisch

für das jeweilige Lösungsverfahren (bzw. die Metaheuristik) definiert und dienen dem Anwender dazu, systematisch Einfluss auf dessen Ablauf zu nehmen.

---

**Algorithmus 13:** MultiStart
 

---

**Eingabe:**  $\mathcal{I}_{\text{RCPSP/M}}$ ; Prioritätsregel  $\pi$ ;  $timeLimit$ ;  $perturbFactor$ ; ein für  $\mathcal{I}_{\text{RCPSP/M}}$  geeignetes Generierungsschema

- 1 berechne basierend auf  $\pi$  eine vorrangzulässige Vorgangsfolge  $\ell_{best}$ ;
- 2 berechne basierend auf  $\ell_{best}$  mit dem Generierungsschema einen Schedule  $(S_{best}, S_{best}^o)$ ;
- 3 **solange**  $timeLimit$  nicht erreicht **tue**
- 4     dupliziere  $\ell_{best}$  zu  $\ell$ ;
- 5     ändere  $\ell$  in Abhängigkeit von  $perturbFactor$  zufällig ab;
- 6     berechne basierend auf  $\ell$  mit dem Generierungsschema einen Schedule  $(S, S^o)$ ;
- 7     **wenn**  $z(S, S^o) < z(S_{best}, S_{best}^o)$  **dann**
- 8         setze  $(S_{best}, S_{best}^o) := (S, S^o)$ ;
- 9         setze  $\ell_{best} := \ell$ ;
- 10    **Ende**
- 11 **Ende**

**Ausgabe:** bester gefundener Schedule  $(S_{best}, S_{best}^o)$

---

Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 13. In Zeile 1 wird eine vorrangzulässige Vorgangsfolge berechnet und als beste gefundene Vorgangsfolge  $\ell_{best}$  gespeichert. Basierend darauf wird mit dem in der Eingabe festgelegten Generierungsschema ein Schedule berechnet und als bester gefundener Schedule  $(S_{best}, S_{best}^o)$  gespeichert (vgl. Zeile 2). Es ist zu beachten, dass abhängig vom Generierungsschema die Vorgangsfolge entweder nur aus den Rückbauvorgängen (beim Dekompositionsansatz) oder aus Rückbau- und Stoffstromvorgängen (beim Integrationsansatz) besteht. Solange das  $timeLimit$  noch nicht erreicht ist (vgl. Zeile 3), wird  $\ell_{best}$  in Zeile 4 zu  $\ell$  dupliziert. Anschließend wird  $\ell$  in Zeile 5 abgeändert. Dazu wird eine Zufallszahl zwischen null und  $perturbFactor \cdot \text{length}(\ell)$  gezogen (wobei  $\text{length}(\ell)$  der Anzahl der Vorgänge in  $\ell$  entspricht). Diese Zufallszahl gibt an, wie viele zufällig ausgewählte Vorgänge auf zufällige, vorrangzulässige Positionen in  $\ell$  verschoben werden. Somit beschreibt der  $perturbFactor$  eine Obergrenze für die Anzahl der möglichen Verschiebungen. Je größer der  $perturbFactor$ , desto höher ist die Wahrscheinlichkeit, dass umfangreichere Abänderungen vorgenommen werden. Basierend auf der abgeänderten Vorgangsfolge  $\ell$  wird in Zeile 6 mit Hilfe des in der Eingabe festgelegten Generierungsschemas ein neuer Schedule berechnet. Falls der neue Schedule besser als der beste gefundene Schedule ist (vgl. Zeile 7), werden in den Zeilen 8 und 9 die abgeänderte Vorgangsfolge und der dazu berechnete Schedule als  $\ell_{best}$  bzw.  $(S_{best}, S_{best}^o)$  festgehalten. Anschließend startet die nächste Iteration, falls das  $timeLimit$  noch nicht überschritten wurde.

Bouleimen und Lecocq (2003, S. 271-273) stellen in ihrem Beitrag eine Simulated-Annealing-Metaheuristik für das (RCPSP) vor. Diese Metaheuristik wird ebenfalls für die Rechenexperimente in dieser Arbeit verwendet. Sie wurde als Alternative zur Multi-Start-Metaheuristik

gewählt, da sie einen stärkeren Fokus auf die lokale Suche legt (vgl. Agarwal u. a. 2015, S. 64) und auch Verschlechterungen beim Durchsuchen des Lösungsraums zulässt. Die Multi-Start-Metaheuristik erfüllt bei nicht allzu kleinem *perturbFactor* hingegen eher die Merkmale einer globalen Suche und akzeptiert neue Lösungen nur, wenn diese echt besser als die bisher gefundenen sind. Im Vergleich zu anderen bekannten Metaheuristiken ist die Simulated-Annealing-Metaheuristik dennoch vergleichsweise einfach zu implementieren, da sie keine aufwendigen Speicherstrukturen (wie beispielsweise bei der Tabu-Search-Metaheuristik) und keine aufwendigen Mechanismen zum Abändern der Vorgangsfolge (wie bei genetischen Metaheuristiken) benötigt. Es gilt jedoch weiterhin, dass die entwickelten Generierungsschemas auch in beliebige andere Metaheuristiken eingebettet werden können.

Die eingesetzte Simulated-Annealing-Metaheuristik nach Bouleimen und Lecocq (2003) ist formal in Algorithmus 14 beschrieben und wird als `SimulatedAnnealing` bezeichnet. Eine Instanz  $\mathcal{I}_{\text{RCSP/M}}$ , eine Prioritätsregel  $\pi$ , ein Zeitlimit *timeLimit*, die Anzahl an Abkühlketten *nbChains*, die Anzahl an Abkühlritten *nbSteps*, die initiale Anzahl zu betrachtender Nachbarn *nbNeighbInit*, ein Faktor *tempReductionFactor*  $\in [0, 1]$  und ein für die Instanz  $\mathcal{I}_{\text{RCSP/M}}$  geeignetes Generierungsschema bilden die Eingabe für `SimulatedAnnealing`. Die nachfolgenden Zeilenreferenzen beziehen sich auf Algorithmus 14. In Zeile 1 wird eine vorrangzulässige Vorgangsfolge berechnet und als bisher beste bekannte Vorgangsfolge  $\ell_{\text{best}}$  gespeichert. Basierend darauf wird mit dem in der Eingabe festgelegten Generierungsschema ein Schedule berechnet und als bester gefundener Schedule  $(S_{\text{best}}, S_{\text{best}}^o)$  gespeichert (vgl. Zeile 2). Auch hier ist zu beachten, dass abhängig vom Generierungsschema die Vorgangsfolge entweder nur aus den Rückbauvorgängen (beim Dekompositionsansatz) oder aus Rückbau- und Stoffstromvorgängen (beim Integrationsansatz) besteht. Beim Simulated Annealing werden auch Variablen zum Speichern der in der jeweiligen Iteration betrachteten Vorgangsfolge bzw. des betrachteten Schedules benötigt. Sie werden in Zeile 3 als  $\ell_{\text{current}}$  bzw.  $(S_{\text{current}}, S_{\text{current}}^o)$  erzeugt und initial auf die beste gefundene Vorgangsfolge bzw. den besten gefundenen Schedule gesetzt. In Zeile 4 wird die initiale Temperatur berechnet. Die verwendete Formel ist Bouleimen und Lecocq (2003, S. 272) entnommen. Mit ihr wird die initiale Temperatur so berechnet, dass die Wahrscheinlichkeit, eine 20-prozentige Verschlechterung des Zielfunktionswertes zu akzeptieren, 1 % beträgt (vgl. Formel in Zeile 19). Solange das *timeLimit* noch nicht erreicht ist (vgl. Zeile 5), werden *nbChains* Kühlketten durchlaufen. Beim Durchlaufen einer Kühlkette wird die Temperatur schrittweise abgesenkt, wodurch zu Beginn der Kühlkette Verschlechterungen des betrachteten Schedules eher zugelassen werden, als zum Ende der Kühlkette. Mit Beginn einer Kühlkette wird in Zeile 7 die Temperatur auf die initiale Temperatur gesetzt. Anschließend werden *nbSteps* Abkühlritte durchlaufen (vgl. Zeile 8). Zu Beginn jedes Abkühlritts wird der *neighbFactor* berechnet, welcher sich in Zeile 10 auf die Anzahl der zu betrachtenden Nachbarn auswirkt. Die dazu verwendete Formel in Zeile 9 sorgt dafür, dass sich diese Anzahl in jedem Abkühlritt verdoppelt. Mit der Schleife in Zeile 10 werden die einzelnen zu betrachtenden Nachbarn durchlaufen. Für jeden Nachbarn wird  $\ell_{\text{best}}$  in Zeile 11 zu  $\ell$  dupliziert. Anschließend wird in Zeile 12 ein zufällig ausgewählter Vorgang in  $\ell$  auf eine andere zufällig ausgewählte, vorrangzulässige Position verschoben. Im Vergleich zur Multi-Start-Metaheuristik fallen die Veränderungen in  $\ell$  demnach deutlich geringfügiger

aus. Basierend auf der abgeänderten Vorgangsfolge  $\ell$  wird in Zeile 13 mit Hilfe des in der Eingabe festgelegten Generierungsschemas ein neuer Schedule berechnet. Falls der neue Schedule besser als  $(S_{current}, S_{current}^o)$  ist (vgl. Zeile 14), werden in Zeile 15 die abgeänderte Vorgangsfolge und der dazu berechnete Schedule als  $\ell_{current}$  bzw.  $(S_{current}, S_{current}^o)$  gesetzt. Falls der neue Schedule auch besser als der beste gefundene Schedule ist, werden in Zeile 17 die entsprechenden Variablen aktualisiert. Falls der neue Schedule schlechter als  $(S_{current}, S_{current}^o)$  ist, werden er und die zugehörige Vorgangsfolge mit einer gewissen Wahrscheinlichkeit dennoch für das weitere Durchsuchen der Nachbarschaft akzeptiert (vgl. Zeile 20). Diese Wahrscheinlichkeit hängt von der aktuellen Temperatur  $temp$  ab und wird in Zeile 19 berechnet. Zur Durchführung eines Zufallsexperiments wird mit  $\text{rnd}()$  eine Zufallszahl im Intervall  $[0, 1)$  abgerufen. Nachdem alle Nachbarn durchlaufen wurden, wird in Zeile 23 die Temperatur durch Multiplikation dem  $tempReductionFactor$  reduziert. Außerdem wird, nachdem alle  $nbChains$  Kühlketten durchlaufen wurden, die Anzahl der initial zu betrachtenden Nachbarn um den Faktor 1,2 erhöht (vgl. Zeile 26).

Die Simulated-Annealing-Metaheuristik von Bouleimen und Lecocq (2003) endet mit einem weiteren Schritt zur „vollständigen Durchsuchung der Nachbarschaft des besten gefundenen Schedules“ (Bouleimen und Lecocq 2003, S. 272). Details werden nicht genannt. Die Formulierung lässt vermuten, dass alle vorrangulässigen Verschiebungen aller Vorgänge in  $\ell_{best}$  ausprobiert werden. Ein solcher Schritt würde für große Instanzen sehr viel Rechenzeit in Anspruch nehmen und wird daher in `SimulatedAnnealing` nicht durchgeführt.

Im Vergleich zum Verfahren von Bouleimen und Lecocq (2003) wurde `SimulatedAnnealing` um die Schleife in Zeile 5, welche die Einhaltung der Rechenzeit prüft, ergänzt. Diese Schleife ist erforderlich, weil in Bouleimen und Lecocq (2003, S. 272) die Anzahl der berechneten Schedules als Abbruchkriterium dient, wohingegen in dieser Arbeit die Rechenzeit als Abbruchkriterium gewählt wurde. Die Anzahl der berechneten Schedules wäre hier kein geeignetes Abbruchkriterium, da neben den Metaheuristiken auch die Constraintprogrammierung und die lineare Programmierung getestet werden sollen. Aufgrund der unterschiedlichen Funktionsprinzipien der einzelnen Lösungsverfahren ist die Rechenzeit dasjenige Abbruchkriterium, das eine bestmögliche Vergleichbarkeit ermöglicht. Voraussetzung ist lediglich, dass sämtliche Rechenexperimente in derselben Testumgebung durchgeführt werden.

Beim Verfahrensparameter  $timeLimit$  ist offensichtlich, dass mit einem höheren Wert bessere Lösungen zu erwarten sind. Für die Prioritätsregel  $\pi$ , welche zur Berechnung der initialen Lösung dient, können die in der Literatur als effizient anerkannten Prioritätsregeln verwendet werden (vgl. Kolisch 1996a). Die übrigen Verfahrensparameter, welche den Ablauf der Simulated-Annealing-Metaheuristik steuern, sind  $nbChains$ ,  $nbSteps$ ,  $nbNeighbInit$  und  $tempReductionFactor$ . Um sie zu bestimmen, ist es üblich, im Rahmen einer sogenannten *Kalibrierung* (in der Literatur auch *Tuning* genannt) vor Beginn der Rechenexperimente geeignete Werte zu ermitteln (vgl. z. B. Bouleimen und Lecocq 2003, S. 275f; Hübner u. a. 2021, S. 284f). Dabei wird zunächst je Verfahrensparameter ein Wertebereich festgelegt. Nach einem gewissen Prinzip werden Kalibrierungsexperimente durchgeführt, um zu testen, welche Werte der Verfahrensparameter die besten Ergebnisse liefern. Wird nach

**Algorithmus 14:** SimulatedAnnealing

---

**Eingabe:**  $\mathcal{I}_{\text{RCPSP/M}}$ ; Prioritätsregel  $\pi$ ;  $timeLimit$ ;  $nbChains$ ;  $nbSteps$ ;  $nbNeighbInit$ ;  $tempReductionFactor$ ; ein für  $\mathcal{I}_{\text{RCPSP/M}}$  geeignetes Generierungsschema

- 1 berechne basierend auf  $\pi$  eine vorrangzulässige Vorgangsfolge  $\ell_{best}$ ;
- 2 berechne basierend auf  $\ell_{best}$  mit dem Generierungsschema einen Schedule  $(S_{best}, S_{best}^o)$ ;
- 3 setze  $\ell_{current} := \ell_{best}$  und  $(S_{current}, S_{current}^o) := (S_{best}, S_{best}^o)$ ;
- 4 berechne die initiale Temperatur  $tempInit := \frac{-z(S_{best}, S_{best}^o) \cdot 0,2}{\ln(0,01)}$ ;
- 5 **solange**  $timeLimit$  nicht erreicht **tue**
- 6     **für**  $coolingChain := 1$  **bis**  $nbChains$  **tue**
- 7         setze die Temperatur  $temp := tempInit$ ;
- 8         **für**  $step := 1$  **bis**  $nbSteps$  **tue**
- 9             berechne  $neighbFactor := 2^{step-1}$ ;
- 10            **für**  $neighb := 1$  **bis**  $nbNeighbInit \cdot neighbFactor$  **tue**
- 11                dupliziere  $\ell_{current}$  zu  $\ell$ ;
- 12                verschiebe einen Vorgang in  $\ell$  zufällig auf eine andere vorrangzulässige Position;
- 13                berechne basierend auf  $\ell$  mit dem Generierungsschema einen Schedule  $(S, S^o)$ ;
- 14                **wenn**  $z(S, S^o) < z(S_{current}, S_{current}^o)$  **dann**
- 15                    setze  $\ell_{current} := \ell$  und  $(S_{current}, S_{current}^o) := (S, S^o)$ ;
- 16                    **wenn**  $z(S, S^o) < z(S_{best}, S_{best}^o)$  **dann**
- 17                        setze  $\ell_{best} := \ell$  und  $(S_{best}, S_{best}^o) := (S, S^o)$ ;
- 18                    **Ende**
- 19                **sonst wenn**  $\exp\left(\frac{z(S, S^o) - z(S_{current}, S_{current}^o)}{temp}\right) > \text{rnd}()$  **dann**
- 20                    setze  $\ell_{current} := \ell$  und  $(S_{current}, S_{current}^o) := (S, S^o)$ ;
- 21                **Ende**
- 22            **Ende**
- 23            setze  $temp := temp \cdot tempReductionFactor$ ;
- 24         **Ende**
- 25     **Ende**
- 26     setze  $nbNeighbInit := nbNeighbInit \cdot 1,2$ ;
- 27 **Ende**

**Ausgabe:** bester gefundener Schedule  $(S_{best}, S_{best}^o)$

---

dem Prinzip des vollständigen Versuchsplans vorgegangen, müssen alle Kombinationen der möglichen Werte getestet werden. Aufgrund des exponentiellen Wachstums ist es daher erforderlich, die Anzahl der Verfahrensparameter sowie die Größe der Wertebereiche zu begrenzen.

Um die Anzahl der Verfahrensparameter zu begrenzen, enthält Algorithmus 14 in den Zeilen 4, 9 und 26 fixe Zahlenwerte. Die Werte in Zeile 4 wurden aus Bouleimen und Lecocq (2003, S. 272) übernommen und sind auch dort fixiert. In Zeile 9 wird der Wert zwei als Basis in der Formel zur Vergrößerung der Nachbarschaft verwendet. Bouleimen und Lecocq (2003, S. 272) verwenden hier stattdessen einen Verfahrensparameter. In ihren Kalibrierungsexperimenten stellen sie jedoch fest, dass eine Verdopplung der Nachbarschaft, d. h. die Basis zwei, die besten Ergebnisse liefert. Es wird angenommen, dass dies auch für die hier betrachteten Instanzen gilt. Zur Vergrößerung der Nachbarschaft in Zeile 26 wird der Wert 1,2 verwendet, woraus eine moderate Vergrößerung um 20 % resultiert. Dies verhindert, dass die Rechenzeit pro Kühlkette zu stark ansteigt und ermöglicht dennoch eine Ausdehnung der Suche. Es wurde kein zusätzlicher Verfahrensparameter anstelle dieses Wertes eingeführt, da Zeile 26 in der ursprünglichen Simulated-Annealing-Metaheuristik nach Bouleimen und Lecocq (2003) nicht enthalten ist. Diese Zeile dient lediglich dazu, über die Iterationen der Schleife in Zeile 5 den Ablauf der Metaheuristik in gewissem Ausmaß zu variieren.

Der Verfahrensparameter *nbNeighbInit* hat laut Bouleimen und Lecocq (2003, S. 275) einen besonders gewichtigen Einfluss auf das Verhalten der Simulated-Annealing-Metaheuristik. Davon hängt ab, wie viele Kühlketten in der gegebenen Rechenzeit durchlaufen werden können. Da die Rechenzeiten für die einzelnen Durchläufe der Generierungsschemas unbekannt sind und sich je nach Instanzgröße stark unterscheiden können, wird die Simulated-Annealing-Metaheuristik in dieser Arbeit um einen Mechanismus ergänzt, der den Wert von *nbNeighbInit* während des laufenden Verfahrens aktualisiert. Dieser Mechanismus wird als *Auto-Kalibrierung* bezeichnet und nach Ablauf ersten zehner Sekunden der Rechenzeit gestartet. Darin wird anhand der gemessenen und gemittelten Rechenzeit für die bis dahin ausgeführten Generierungsschemas berechnet, wie viele Abkühlketten voraussichtlich innerhalb von 60 Sekunden durchlaufen werden. Falls mindestens eine Abkühlkette durchlaufen wird, wird *nbNeighbInit* auf dem in der Eingabe gegebenen Wert belassen. Falls weniger als eine Abkühlkette durchlaufen wird, wird *nbNeighbInit* auf denjenigen Wert gesetzt, mit dem innerhalb von 60 Sekunden eine Abkühlkette durchlaufen wird. Beim Prüfen der Abbruchbedingung der Schleife in Zeile 10 wird nach Durchführung der Auto-Kalibrierung der aktualisierte Wert von *nbNeighbInit* herangezogen. Durch die Auto-Kalibrierung wird daher sichergestellt, dass selbst bei vergleichsweise hohen Rechenzeiten für die Generierungsschemas auch eine Abkühlung im Sinne des Simulated Annealing durchgeführt wird.

## 6.2.2 Verwendete Testinstanzen

Um eine endliche Anzahl an Testinstanzen zu erzeugen, werden für die Instanzparameter bestimmte Wertebereiche festgelegt. Tabelle 6.1 listet diese auf.

Instanzparameter	Wertebereich
<i>I</i>	{30; 120}
<i>NC</i>	{1,5; 2,1}
<i>RS</i>	{0,2; 0,5}
<i>RF</i>	{0,25; 0,5}
<i>INV</i>	{200; 1000}
<i>NREL</i>	{50; 200}
<i>PREL</i>	{0,25; 1}
<i>RFP</i>	{0; 0,5}
<i>DUR</i>	{ <i>dur</i> <sub>1</sub> ; <i>dur</i> <sub>2</sub> ; <i>dur</i> <sub>3</sub> }

**Tabelle 6.1:** Wertebereiche der Instanzparameter.

Die Wertebereiche für *I*, *NC*, *RS* und *RF* können maximal die in der PSPLIB verwendeten Wertebereiche umfassen (vgl. Abschnitt 6.1.2). Zur Reduktion der Anzahl an Testinstanzen werden hier jeweils nur zwei Werte aus den PSPLIB-Wertebereichen verwendet. Als größter Wert für *NREL* wurde 200 gewählt, sodass bei *PREL* = 1 und *I* = 120 insgesamt  $120 \cdot 200 = 24\,000$  Stoffeinheiten freigesetzt werden. Zum Vergleich: Beim Rückbau des Blocks 2 des Kernkraftwerks Philippsburg wird angegeben, dass 15 590 Mg Stoffe von der Strahlenschutzverordnung erfasst sind (vgl. Abschnitt 3.2.1). Wenn angenommen wird, dass 1 Mg (= eine Tonne) Stoffe als eine Stoffeinheit modelliert wird, ist die größte Testinstanz mit 24 000 Stoffeinheiten in dieser Hinsicht mit einem großen kerntechnischen Rückbauprojekt vergleichbar. Mit 50 steht ein weiterer Wert für *NREL* zur Verfügung, der die Erzeugung kleinerer Testinstanzen ermöglicht. Als kleinster Wert für *INV* wurde 200 gewählt, womit bei *NREL* = 200 gerade die Stofffreisetzung eines Rückbauvorgangs in jedes Lager passt. Mit *INV* = 1000 steht hingegen deutlich mehr Lagerkapazität zur Verfügung. Für *PREL* wurden die Werte 0,25 und eins gewählt, womit sowohl Testinstanzen erzeugt werden, in denen nur ein Teil der Rückbauvorgänge Stoffe freisetzen, als auch Testinstanzen, in denen alle Rückbauvorgänge Stoffe freisetzen. Für *RFP* wurden die Werte null und 0,5 gewählt. In Testinstanzen mit *RFP* = 0 gibt es keine erneuerbaren Ressourcen, die von Rückbau- und Stoffstromvorgängen genutzt werden, d. h., die Ressourcenpools sind zwischen Rückbau- und Stoffstrombereich getrennt. Mit *RFP* = 0,5 werden Testinstanzen erzeugt, in denen Stoffstromvorgänge auch die PSPLIB-Ressourcen *R*<sub>1</sub>, . . . , *R*<sub>4</sub> nutzen. In einer solchen Situation konkurrieren Stoffstromvorgänge mit Rückbauvorgängen um erneuerbare Ressourcen. Für *DUR* werden drei Vektoren verwendet, die kurz mit *dur*<sub>1</sub>, *dur*<sub>2</sub> und *dur*<sub>3</sub> bezeichnet sind. Dabei gilt *dur*<sub>1</sub> = (0,06; 0,06; . . . ; 0,06); *dur*<sub>2</sub> = (0,1; 0,08; 0,06; 0,1; 0,04; 0,1; 0,02; 0,02) und *dur*<sub>3</sub> = (0,02; 0,04; 0,06; 0,02; 0,08; 0,04; 0,1; 0,08). Diese Werte wurden gewählt, sodass die erwartete Dauer zur vollständigen Verarbeitung einer Stoffeinheit mit  $\approx 0,2$  jeweils ungefähr gleich groß ist. Mit *dur*<sub>1</sub> dauern alle Verarbeitungsschritte gleich lang. Mit *dur*<sub>2</sub>



wird eine Situation beschrieben, in der der Engpass bei der Stoffverarbeitung im oberen Teil des Flussnetzwerks liegt (vgl. Abbildung 3.5). Mit  $dur_3$  liegt der Engpass hingegen im unteren Teil.

Es ist zu beachten, dass aufgrund der Diskretisierung der Zeit nur ganze Zahlen für die Dauern zugelassen sind. Da die Vektoren für  $DUR$  Dezimalzahlen enthalten, müssen die Dauern aller Rückbauvorgänge und die Dauern aller Verarbeitungsschritte auf ein ganzzahliges Vielfaches ihres größten gemeinsamen Teiles hochskaliert werden. Der größte gemeinsame Teiler ist 0,02. Daher müssen alle Dauern mit dem Faktor 50 multipliziert werden. Dies führt dazu, dass sich die Auflösung der Zeithorizonte stark erhöht: Die besten gefundenen Projektdauern der PSPLIB-Instanzen aus dem J120-Set liegen in etwa zwischen 60 und 230 Perioden. Nach der Skalierung liegen sie zwischen 3000 und 11 500 Perioden. Da kerntechnische Rückbauprojekte mehrere Jahrzehnte dauern können, handelt es sich hierbei jedoch nicht um übermäßig große Zeithorizonte (unter der Annahme, dass tagesbasiert geplant wird). Die Hochskalierung der Zeithorizonte ist insbesondere für granulare Instanzen erforderlich, da für jeden Verarbeitungsschritt und jede Stoffeinheit ein granularer Stoffstromvorgang erzeugt wird. Bei aggregierten Instanzen sind die Dauern der Stoffstromvorgänge aufgrund der Aggregation in der Regel größer. Dann kann, falls unter den Dauern aller Vorgänge ein größter gemeinsamer Teiler echt größer als eins existiert, wieder herunterskaliert werden.

Würden die Testinstanzen nach dem Prinzip des vollständigen Versuchsplans erzeugt werden, d. h. mit allen möglichen Kombinationen der in Tabelle 6.1 gegebenen Werte, ergäben sich daraus  $2^8 \cdot 3 = 768$  Testinstanzen. Der Rechenaufwand wäre zu hoch, wenn jede Testinstanz mit den verfügbaren Lösungsverfahren und Rechenzeiten im Minutenbereich gelöst werden soll. Daher werden zwei Mengen von Testinstanzen erzeugt:

- Menge  $\mathcal{T}_1$ : Die Instanzparameter  $NC$ ,  $RS$  und  $RF$  werden auf  $NC = 2,1$ ,  $RS = 0,5$  und  $RF = 0,5$  fixiert. Die übrigen Instanzparameter werden variiert, sodass nach dem Prinzip des vollständigen Versuchsplans  $2^5 \cdot 3 = 96$  Testinstanzen  $\mathcal{I}_{\text{RMCPSP}}$  mittels `GenerateTestInstance` erzeugt und zu  $\mathcal{T}_1$  hinzugefügt werden. Da außer  $I$  alle Instanzparameter der PSPLIB-Instanzen fixiert sind, bauen die Testinstanzen der Menge  $\mathcal{T}_1$  allesamt auf zwei PSPLIB-Instanzen auf: `j3038_1` für  $I = 30$  und `j12050_1` für  $I = 120$ . Für beide PSPLIB-Instanzen sind mit 48 und 116 die optimalen Zielfunktionswerte bekannt. Aufgrund der Hochskalierung des Zeithorizonts mit dem Faktor 50 beträgt  $LB_{\text{PSPLIB}}$  für die Testinstanzen der Menge  $\mathcal{T}_1$  daher entweder 2400 (für  $I = 30$ ) oder 5800 (für  $I = 120$ ). Abhängig von der Kombination der Parameter  $I$ ,  $NREL$  und  $PREL$  werden je Testinstanz zwischen 350 und 24 000 Stoffeinheiten freigesetzt ( $= \sum_{i \in V} \sum_{w \in \mathcal{W}} f_{iw}^{y,lin}$ ).  $LB_{\text{material}}$  bewegt sich zwischen null und 69 265. Durch Anwendung des granularen Transformationsschemas auf die 96 Testinstanzen  $\mathcal{I}_{\text{RMCPSP}}$  werden 96 Instanzen  $\mathcal{I}_{\text{RCPSP/M-gra}}$  erzeugt. Diese Instanzen umfassen jeweils 30 oder 120 Rückbauvorgänge (abhängig von  $I$ ) und zwischen 1073 und 76 257 granulare Stoffstromvorgänge. Durch Anwendung des aggregierten Transformationsschemas werden darüber hinaus 96 Instanzen  $\mathcal{I}_{\text{RCPSP/M-agg}}$  erzeugt. Diese Instanzen umfassen jeweils ebenfalls 30 oder 120 Rückbauvorgänge und zwischen 106 und 2348 aggregierte Stoffstromvorgänge. Dadurch, dass in  $\mathcal{T}_1$  sämtliche Instanzparameter, die die

Eigenschaften des Stoffstrombereichs beschreiben, variiert werden, eignen sich die Testinstanzen in  $\mathcal{T}_1$  insbesondere dazu, den Einfluss des Stoffstrombereichs auf die Lösbarkeit zu untersuchen.

- Menge  $\mathcal{T}_2$ : Die Instanzparameter  $INV$ ,  $NREL$  und  $PREL$  werden auf  $INV = 200$ ,  $NREL = 50$  und  $PREL = 1$  fixiert. Die übrigen Instanzparameter werden variiert, sodass nach dem Prinzip des vollständigen Versuchsplans  $2^5 \cdot 3 = 96$  Testinstanzen  $\mathcal{I}_{\text{RMCPSP}}$  mittels `GenerateTestInstance` erzeugt und zu  $\mathcal{T}_2$  hinzugefügt werden. Da nun alle Instanzparameter der PSPLIB-Instanzen variiert werden, bauen die Testinstanzen der Menge  $\mathcal{T}_2$  auf insgesamt 16 PSPLIB-Instanzen auf. Aufgrund der Hochskalierung des Zeithorizonts mit dem Faktor 50 liegt  $LB_{\text{PSPLIB}}$  zwischen 1900 und 6850. Da  $NREL$  und  $PREL$  fixiert sind, werden in jeder Testinstanz entweder 1500 (für  $I = 30$ ) oder 6000 (für  $I = 120$ ) Stoffeinheiten freigesetzt ( $= \sum_{i \in V} \sum_{w \in \mathcal{W}} f_{iw}^{y,lin}$ ).  $LB_{\text{material}}$  bewegt sich zwischen 1416 und 15 350. Durch Anwendung des granularen Transformationsschemas auf die 96 Testinstanzen  $\mathcal{I}_{\text{RMCPSP}}$  werden 96 Instanzen  $\mathcal{I}_{\text{RCPSM-gra}}$  erzeugt. Diese Instanzen umfassen jeweils 30 oder 120 Rückbauvorgänge (abhängig von  $I$ ) und zwischen 4595 und 19 271 granulare Stoffstromvorgänge. Durch Anwendung des aggregierten Transformationsschemas werden darüber hinaus 96 Instanzen  $\mathcal{I}_{\text{RCPSM-agg}}$  erzeugt. Diese Instanzen umfassen jeweils ebenfalls 30 oder 120 Rückbauvorgänge und zwischen 485 und 2067 aggregierte Stoffstromvorgänge. Dadurch, dass in  $\mathcal{T}_2$  sämtliche Instanzparameter, die die Eigenschaften des Rückbaubereichs beschreiben, variiert werden, eignen sich die Testinstanzen in  $\mathcal{T}_2$  insbesondere dazu, den Einfluss des Rückbaubereichs auf die Lösbarkeit zu untersuchen.

### 6.2.3 Verwendete Lösungsstrategien

Eine *Lösungsstrategie* beschreibt eine Festlegung aller variablen Bestandteile eines Rechenexperiments. Nachfolgend sind die variablen Bestandteile der Rechenexperimente dieser Arbeit aufgelistet. Um eine Lösungsstrategie möglichst kurz und aussagekräftig benennen zu können, wird eine spezielle Notation eingeführt. Dazu werden die einzelnen Ausprägungen der Bestandteile jeweils wie nachfolgend angegeben mit in Schreibmaschienschrift gesetzten Abkürzungen bezeichnet. Durch eine über Senkrechtstriche getrennte Aneinanderreihung aller Ausprägungen kann dadurch jede Lösungsstrategie eindeutig beschrieben werden.

- Transformationsschema: Die erzeugten Testinstanzen sind Instanzen des (RMCPSP), wohingegen die Lösungsverfahren nur auf Instanzen des (RCPSM) anwendbar sind. Zur systematischen Umwandlung der Instanzen wurden in Kapitel 4 Transformationsschemas vorgestellt. Für die Rechenexperimente wird zwischen dem granularen Transformationsschema, abgekürzt mit *gra*, und dem aggregierten Transformationsschema, abgekürzt mit *agg*, unterschieden.
- Lösungsverfahren: Als Lösungsverfahren stehen die in Kapitel 5 vorgestellten Verfahren zur Verfügung. Das Lösungsverfahren für granulare Instanzen unter Ver-

wendung des Generierungsschemas nach einem Dekompositionsansatz wird mit SGS-gra (SGS = Schedule Generation Scheme) abgekürzt. Das Lösungsverfahren für aggregierte Instanzen unter Verwendung des Generierungsschemas nach einem Dekompositionsansatz wird mit SGS-agg-d abgekürzt. Das Lösungsverfahren für aggregierte Instanzen unter Verwendung des Generierungsschemas nach einem Integrationsansatz wird mit SGS-agg-i abgekürzt. Das Lösungsverfahren für beliebige Instanzen unter Verwendung des Generierungsschemas nach einem Integrationsansatz wird mit SGS-arb abgekürzt. Darüber hinaus lassen sich die Testinstanzen per Constraintprogrammierung oder linearer Programmierung lösen. Als Solver für die Constraintprogrammierung wird aus den in Abschnitt 5.8 dargelegten Gründen der CP Optimizer verwendet, welcher kurz als CP bezeichnet wird. Als Solver für die lineare Programmierung wird CPLEX verwendet, welcher als CPLEX bezeichnet wird.

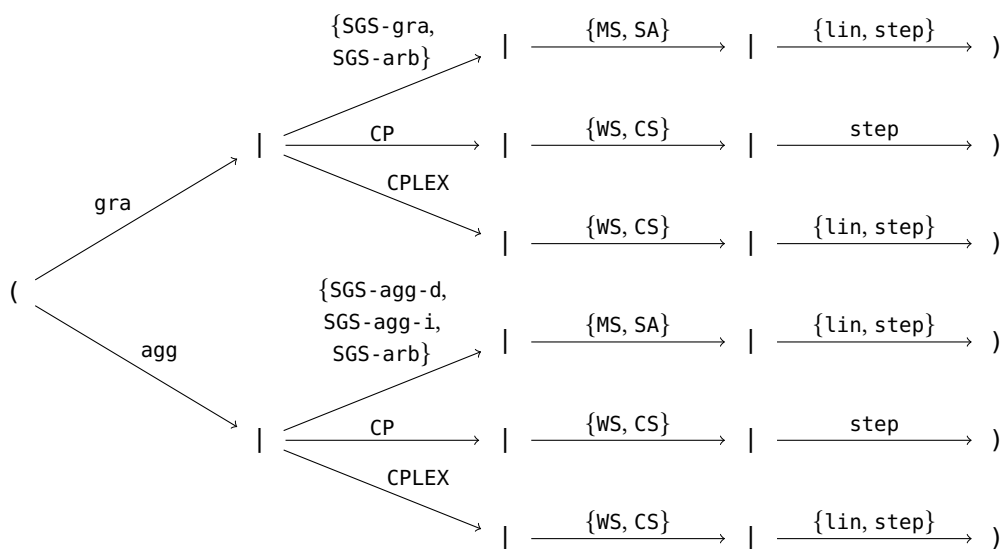
- Metaheuristik bzw. Startkonfiguration: Wird ein Generierungsschema als Lösungsverfahren verwendet, muss es in eine der beiden in Abschnitt 6.2.1 vorgestellten Metaheuristiken eingebettet werden. Für die Multi-Start-Metaheuristik wird die Abkürzung MS verwendet. Für die Simulated-Annealing-Metaheuristik wird die Abkürzung SA verwendet. Werden der CP Optimizer oder CPLEX als Lösungsverfahren verwendet, ist keine Metaheuristik anzugeben. Stattdessen wird jedoch zwischen zwei Startkonfigurationen unterschieden: Bei einem Kaltstart erhält der Solver keine Informationen über zulässige Lösungen. Bei einem Warmstart wird dem Solver eine mit einem passenden Generierungsschema und der Prioritätsregel Smallest Latest Start Time (vgl. Kolisch 1996b, S. 326ff) berechnete Lösung als Startlösung übergeben. Die entsprechenden Abkürzungen lauten CS für Kaltstart und WS für Warmstart.
- Freisetzungs- bzw. Nutzungsarten: Falls die Freisetzungs- bzw. Nutzungsarten bei der Lösung unverändert übernommen werden, wird dies in der Lösungsstrategie mit der Abkürzung lin ausgedrückt. Diese Abkürzung deutet auf den linearen Verlauf des kumulativen Ressourcenprofils hin. Ein solcher Verlauf liegt bei den Testinstanzen vor, da die Stofffreisetzungen stets gleichmäßig über die Ausführung der Rückbauvorgänge erfolgen (wie im Verfahren zur Erzeugung der Testinstanzen festgelegt). In aggregierten Instanzen erfolgen die kumulativen Ressourcennutzungen der Stoffstromvorgänge ebenfalls gleichmäßig während der Vorgangsausführung (wie im aggregierten Transformationsschema festgelegt). Wie in Abschnitt 5.8 erläutert, kann der CP Optimizer nur die Stufenvariante einer Instanz  $\mathcal{I}_{\text{RCPSP/M}}$  lösen. Darin werden die gleichmäßigen Freisetzungen bzw. Nutzungen während der Vorgangsausführung auf die Nutzungen zum Start oder Ende hinzuaddiert. Falls die Stufenvariante bei der Lösung verwendet wird, wird die Lösungsstrategie mit der Abkürzung step versehen.

Wird beispielsweise von der Lösungsstrategie (gra|SGS-gra|MS|lin) gesprochen, bedeutet dies: Die Testinstanzen wurden mit dem granularen Transformationsschema in Instanzen des (RCPSP/M) transformiert. Als Lösungsverfahren wurde das Generierungsschema für granulare Instanzen nach einem Dekompositionsansatz gewählt. Dieses wurde in die Multi-Start-Metaheuristik eingebettet. Die Freisetzungs- und Nutzungsarten wurden unverändert, wie in den Testinstanzen gegeben, übernommen. Das heißt im konkreten Fall, dass

die Stofffreisetzungen aller Rückbauvorgänge gleichmäßig über die Vorgangsausführung und die Nutzungen von kumulativen Ressourcen durch die granularen Stoffstromvorgänge zum Start (für Auslagerungen) oder zum Ende (für Einlagerungen) erfolgen.

Wird ein Punkt in der Bezeichnung der Lösungsstrategie verwendet, erfüllt er die Funktion eines Platzhalters. Eine Aussage, wie beispielsweise „die Lösungsstrategien (gra|SGS-arb|·|·) waren nicht in der Lage, sämtliche granuläre Instanzen zu lösen“ gilt daher für (gra|SGS-arb|MS|lin), (gra|SGS-arb|MS|step), (gra|SGS-arb|SA|lin) und (gra|SGS-arb|SA|step).

Abbildung 6.1 veranschaulicht in Form eines Entscheidungsbaums, welche Lösungsstrategien betrachtet werden. Um die Anzahl der Verzweigungen zu reduzieren, wurden gleichartige Zweige unter Verwendung der Mengenschreibweise zusammengefasst. Eine Anwendung der Lösungsverfahren für aggregierte Instanzen (SGS-agg-d und SGS-agg-i) wäre aufgrund der in Abschnitt 5.6 beschriebenen Kompatibilitäten auch auf granuläre Instanzen möglich. Darauf wird jedoch verzichtet, um die Anzahl der Kombinationen zu reduzieren. Stattdessen wird das Lösungsverfahren für beliebige Instanzen (SGS-arb) auf granuläre und aggregierte Instanzen angewendet. Damit kann getestet werden, wie sich die theoretischen Kompatibilitäten in der Praxis verhalten.



**Abbildung 6.1:** Veranschaulichung der betrachteten Lösungsstrategien.

Das aggregierte Transformationsschema wurde als Erweiterung des granularen Transformationsschemas konzipiert. Das heißt, ein zulässiger Schedule für eine aggregierte Instanz ist auch ein zulässiger Schedule für eine granuläre Instanz (sofern die aggregierten Vorgänge wieder in granuläre Vorgänge rücktransformiert werden; vgl. Abschnitt 4.2.2). Außerdem wurde die Stufenvariante einer Instanz so definiert, dass der benötigte Lagerplatz bereits zu Beginn jedes Vorgangs blockiert und erst am Ende wieder freigegeben wird. Das heißt, ein zulässiger Schedule für die Stufenvariante einer Instanz ist auch ein zulässiger Schedule für die originale Instanz (vgl. Abschnitt 5.8). Aufgrund dieser Zusammenhänge sind alle mit den jeweiligen Lösungsstrategien berechneten Zielfunktionswerte

zu einer bestimmten Testinstanz auf Basis der absoluten Zielfunktionswerte miteinander vergleichbar. Es handelt sich jeweils um dasselbe Rückbauprojekt, welches lediglich auf verschiedene Weise modelliert wurde. Findet beispielsweise (gra|SGS-gra|MS|lin) für eine bestimmte Testinstanz einen Schedule mit kleinerem Zielfunktionswert als (agg|SGS-arb|MS|step), kann trotz der Unterschiede in der Modellierung die Aussage getroffen werden, dass (gra|SGS-gra|MS|lin) in diesem Fall die bessere Lösungsstrategie ist. Sie findet für das betrachtete Rückbauprojekt einen Schedule mit kürzerer Projektdauer.

## 6.3 Experimente mit Fokus auf den Stoffstrombereich

Die nachfolgend vorgestellten Experimente fokussieren sich darauf, verschiedene Einflüsse auf die Lösbarkeit des (RMCPSP) unter Variation der Charakteristika des Stoffstrombereichs zu untersuchen. Daher werden die Testinstanzen der Menge  $\mathcal{T}_1$  verwendet, in denen die Instanzparameter des Stoffstrombereichs variabel gestaltet sind (vgl. Abschnitt 6.1.2).

Sämtliche Experimente wurden auf einem Rechner mit AMD Ryzen 9-Prozessor (4,0 Gigahertz Taktfrequenz, 12 Kerne) und 128 Gigabyte Arbeitsspeicher durchgeführt. Die neu entwickelten Lösungsverfahren wurden als Java-Programm implementiert und mit Eclipse 2019-12 kompiliert. Für die Berechnungen mit CPLEX und dem CP Optimizer wurde das IBM ILOG CPLEX Optimization Studio in der Version 12.9.0 verwendet. Der Zugriff auf CPLEX und den CP Optimizer erfolgte über die Concert-Technology-Bibliothek von IBM, welche eine Schnittstelle für Java-Programme zur Verfügung stellt.

Zunächst werden in Abschnitt 6.3.1 der Ablauf der Rechenexperimente und eine Ergebnisübersicht vorgestellt. In den darauffolgenden Abschnitten 6.3.2 bis 6.3.5 werden die Ergebnisse im Detail ausgewertet. Abschnitt 6.3.6 befasst sich mit einer Ergänzung der Rechenexperimente um Langzeitexperimente, in denen höhere Zeitlimits gewählt werden.

### 6.3.1 Ablauf und Ergebnisübersicht

Für MultiStart wird auf eine Kalibrierung verzichtet, da diese Metaheuristik den Lösungsraum ohnehin auf eine vergleichsweise primitive Art absucht. Die Prioritätsregel  $\pi$  wird auf Smallest Latest Start Time gesetzt. Der *perturbFactor* wird auf 0,4 gesetzt, womit deutlich größere Schritte im Lösungsraum möglich sind, als bei SimulatedAnnealing. Dennoch ist mit diesem Wert sichergestellt, dass nur etwas weniger als die Hälfte der besten gefundenen Vorgangsfolge abgeändert wird.

SimulatedAnnealing enthält eine größere Anzahl an Verfahrensparametern. Zur Belegung des Verfahrensparameters *nbNeighbInit* wurde, wie in Abschnitt 6.2.1 beschrieben, der Mechanismus einer Auto-Kalibrierung geschaffen. Die Prioritätsregel  $\pi$  wird auf Smallest Latest Start Time gesetzt. Die übrigen Verfahrensparameter werden in vorgelagerten Kalibrierungsexperimenten bestimmt. Als Wertebereiche werden *nbChains*  $\in \{5, 10, 20\}$ , *nbSteps*  $\in \{5, 10, 20\}$ , *nbNeighbInit*  $:= \{10, 50, 100\}$ , *tempReductionFactor*  $\in \{0,2; 0,4; 0,6\}$

gewählt. Anschließend werden zufällig zehn Testinstanzen aus den 96 Testinstanzen der Menge  $\mathcal{T}_1$  ausgewählt. Für jede der gewählten zehn Testinstanzen wird für jede Kombination aus möglichen Werten für die Verfahrensparameter eine 60-sekündige Berechnung ( $timeLimit = 60$  s) mit SimulatedAnnealing gestartet. Diejenige Kombination der Verfahrensparameter, bei welcher die 60-sekündigen Berechnungen zu den durchschnittlich besten Ergebnissen führen, wird gewählt. Im Einzelnen haben sich damit folgende Werte für die Verfahrensparameter ergeben:

- Für die Lösungsstrategien (gra|SGS-gra|SA|·):  $nbChains = 5$ ,  $nbSteps = 5$ ,  $nbNeighbInit = 50$  und  $tempReductionFactor = 0,2$ .
- Für die Lösungsstrategien (gra|SGS-arb|SA|·):  $nbChains = 10$ ,  $nbSteps = 5$ ,  $nbNeighbInit = 10$  und  $tempReductionFactor = 0,4$ .
- Für die Lösungsstrategien (agg|·|SA|·):  $nbChains = 10$ ,  $nbSteps = 20$ ,  $nbNeighbInit = 50$  und  $tempReductionFactor = 0,4$ .

Nach Festlegung der Verfahrensparameter wird für jede der 96 Testinstanzen aus der Menge  $\mathcal{T}_1$  und jede Lösungsstrategie eine zehnminütige Berechnung ( $timeLimit = 600$  s) durchgeführt. Dieses Zeitlimit wird als sinnvoller Kompromiss zwischen erzielbarer Lösungsqualität und Rechenaufwand erachtet und ist die Basis für die nachfolgenden Auswertungen. In Abschnitt 6.3.6 wird abschließend ein höheres Zeitlimit gewählt. Für die Lösungsstrategien mit Metaheuristiken werden jeweils sechs parallele Threads gestartet. Auf jedem Thread läuft eine Metaheuristik. Es erfolgt kein Austausch von Informationen zwischen den Threads. Da für die Metaheuristiken aller Threads dieselben Verfahrensparameter verwendet werden, ergibt sich der Unterschied im Verfahrensablauf einzelner Threads nur durch die Zufallskomponenten. Die Parallelisierung ermöglicht es somit, rein zufallsbasiert einen noch größeren Teil des Lösungsraums abzusuchen.

Die Ergebnisse der Rechenexperimente sind in Tabelle 6.2 dargestellt. Als Maß zur Beurteilung der Lösungsqualität wird darin die relative Abweichung von der besten unteren Schranke verwendet. Sei  $z$  der von einer Lösungsstrategie berechnete Zielfunktionswert für eine Testinstanz. Dann wird formal für jede Lösungsstrategie und jede Testinstanz ein Wert

$$= \frac{z - \max\{LB_{PSPLIB}, LB_{material}\}}{\max\{LB_{PSPLIB}, LB_{material}\}}$$

berechnet. Diese Werte werden anschließend für jede Lösungsstrategie über alle Testinstanzen gemittelt und sind in Zeile 1 der Tabelle 6.2 angegeben. In den übrigen Zeilen wird nach den verschiedenen Werten der für  $\mathcal{T}_1$  als variabel festgelegten Instanzparameter  $I$ ,  $INV$ ,  $NREL$ ,  $PREL$ ,  $RFP$  und  $DUR$  unterschieden. Hier werden die Testinstanzen nach dem jeweiligen Parameterwert gefiltert und anschließend wird der Durchschnitt nur über die gefilterten Testinstanzen gebildet. Beispielsweise enthalten 48 der insgesamt 96 Testinstanzen  $I = 30$  Rückbauvorgänge. Wenn in der Zeile mit  $I = 30$  ein Eintrag mit 10 % vorhanden ist, hat die entsprechende Lösungsstrategie für diese 48 Testinstanzen Zielfunktionswerte berechnet, die durchschnittlich um 10 % von der besten unteren Schranke abweichen. In

jeder Zeile ist der Eintrag zu derjenigen Lösungsstrategie, die am besten abgeschnitten hat, fettgedruckt.

Die durchschnittlichen relativen Abweichungen können nur berechnet werden, falls mit einer bestimmten Lösungsstrategie für alle Testinstanzen, die zu dem jeweils betrachteten Wert des Instanzparameters gehören, eine Lösung gefunden werden konnte. Falls für eine Testinstanz keine Lösung gefunden werden konnte, ist die relative Abweichung unbestimmt (oder unendlich). Die Aussagekraft der Durchschnittswerte wäre eingeschränkt, wenn der Durchschnitt nur über gelöste und nicht über ungelöste Testinstanzen gebildet wird. Daher ist in Tabelle 6.2 ein „-“ eingetragen, falls nicht für alle Instanzen, die zu dem jeweils betrachteten Wert des Instanzparameters gehören, eine zulässige Lösung gefunden wurde.

Neben der Problematik mit ungelösten Instanzen ist auch kritisch anzumerken, dass Durchschnittswerte empfindlich gegenüber Ausreißern sind. Daher wird in Tabelle 6.3 ein weiteres Maß zur Beurteilung der Lösungsqualität verwendet. Es handelt sich dabei um ein binäres Maß, welches aussagt, ob eine Lösungsstrategie für eine Testinstanz die beste bekannte Lösung gefunden hat oder nicht. Damit können auch ungelöste Instanzen miteinbezogen werden. Sei  $z$  der von einer Lösungsstrategie berechnete Zielfunktionswert für eine Testinstanz und  $z_{best}$  der über alle Lösungsstrategien beste bekannte Zielfunktionswert zu dieser Testinstanz. Dann wird formal für jede Lösungsstrategie und jede Testinstanz ein Wert

$$= \begin{cases} 1 & \text{falls } z = z_{best}, \\ 0 & \text{falls } z > z_{best} \text{ oder die Testinstanz ungelöst ist,} \end{cases}$$

berechnet. Diese Werte werden wie oben beschrieben gemittelt. Die gemittelten Werte geben dann den Anteil der Testinstanzen an, für die eine Lösungsstrategie die beste bekannte Lösung berechnet hat. Das heißt, wenn beispielsweise in der Zeile mit  $I = 30$  ein Eintrag mit 50 % vorhanden ist, hat die entsprechende Lösungsstrategie für die Hälfte der 48 Testinstanzen mit  $I = 30$  Rückbauvorgängen Zielfunktionswerte berechnet, die dem besten gefundenen Zielfunktionswert entsprechen. In jeder Zeile ist der Eintrag zu derjenigen Lösungsstrategie, die am besten abgeschnitten hat, fettgedruckt.

Wie in den Tabellen 6.2 und 6.3 erkennbar ist, sind keine Lösungsstrategien mit dem Solver CPLEX aufgeführt. Grund dafür ist, dass das in Abschnitt 5.9 vorgestellte lineare Programm für das (RCPSP/M) bereits für die kleinste und am einfachsten zu lösende Testinstanz (30 Rückbauvorgänge, 110 aggregierte Stoffstromvorgänge, keine bindenden kumulativen Ressourcenrestriktionen) erst nach 37 Minuten optimal gelöst werden konnte. Im Vergleich dazu konnte der CP Optimizer bereits nach 0,06 Sekunden die optimale Lösung finden. Für eine etwas größere Instanz (30 Rückbauvorgänge, 140 aggregierte Stoffstromvorgänge, bindende kumulative Ressourcenrestriktionen) hat bereits das „Presolve“ laut CPLEX-Protokoll 40 Minuten gedauert und das „Root node processing“ weitere 30 Minuten. Das bedeutet, innerhalb einer Stunde konnte das eigentliche Lösungsverfahren für das lineare Programm nicht starten und es wurde somit keine zulässige Lösung gefunden. Der CP Optimizer konnte bereits nach 0,17 Sekunden eine zulässige Lösung finden. Da mit dem CP Optimizer ein leistungsfähiges exaktes Lösungsverfahren zur Verfügung steht, wird in

		gra									
Instanz- parame- ter	Werte	SGS-gra				SGS-arb				CP	
		MS		SA		MS		SA		CS	WS
		lin	step	lin	step	lin	step	lin	step	step	step
1	alle	9,11	9,83	9,61	10,27	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
2	<i>I</i>	30	12,83	13,71	12,99	13,78	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	8,33
3		120	<b>5,39</b>	5,94	6,24	6,77	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
4	<i>INV</i>	200	10,91	11,68	11,24	12,01	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
5		1000	7,31	7,98	7,98	8,54	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
6	<i>NREL</i>	50	7,49	7,73	7,97	8,33	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	4,36
7		200	10,73	11,93	11,25	12,21	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
8	<i>PREL</i>	0,25	7,51	8,26	8,27	8,77	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	5,58
9		1	10,71	11,39	10,96	11,78	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
10	<i>RFP</i>	0	<b>3,97</b>	4,83	4,20	5,00	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	5,32
11		0,5	14,25	14,83	15,03	15,55	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
12	<i>DUR</i>	<i>dur</i> <sub>1</sub>	6,37	7,30	6,78	7,55	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	4,11
13		<i>dur</i> <sub>2</sub>	10,45	11,12	10,89	11,62	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>
14		<i>dur</i> <sub>3</sub>	10,51	11,06	11,17	11,64	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>

<sup>a</sup>Es wurde nicht für alle Instanzen eine zulässige Lösung gefunden.  
Aus Darstellungsgründen wird die Tabelle unten fortgesetzt.

		agg													
		SGS-agg-d				SGS-agg-i				SGS-arb				CP	
		MS		SA		MS		SA		MS		SA		CS	WS
		lin	step	lin	step	lin	step	lin	step	lin	step	lin	step	step	step
1		8,98	10,59	9,04	10,56	<b>4,72</b>	7,19	5,03	7,04	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	5,28	5,30
2		12,25	14,41	12,42	14,40	<b>3,46</b>	6,81	3,69	6,48	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	4,58	4,65
3		5,70	6,77	5,66	6,73	5,98	7,57	6,37	7,60	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	5,98	5,94
4		10,80	13,15	10,85	13,10	<b>7,17</b>	10,89	7,32	10,44	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	8,16	8,24
5		7,15	8,03	7,24	8,03	<b>2,28</b>	3,49	2,74	3,64	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	2,40	2,35
6		6,92	7,45	7,02	7,37	2,45	3,58	2,73	3,71	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	<b>2,29</b>	2,30
7		11,04	13,73	11,06	13,75	<b>7,00</b>	10,80	7,33	10,38	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	8,27	8,29
8		7,50	9,17	7,51	9,16	3,76	5,94	<b>3,65</b>	5,48	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	4,54	4,54
9		10,45	12,01	10,57	11,97	<b>5,69</b>	8,44	6,41	8,60	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	6,03	6,05
10		4,36	6,15	4,36	6,15	4,36	6,46	4,65	6,42	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	5,26	5,29
11		13,59	15,03	13,72	14,98	<b>5,09</b>	7,93	5,41	7,67	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	5,30	5,30
12		5,94	7,69	6,12	7,62	2,24	4,45	<b>2,16</b>	4,47	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	2,63	2,68
13		10,26	11,94	10,29	11,97	<b>6,15</b>	8,75	6,28	8,31	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	6,98	7,00
14		10,73	12,14	10,71	12,09	<b>5,78</b>	8,38	6,65	8,35	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	6,24	6,20

**Tabelle 6.2:** Experimente mit Fokus auf den Stoffstrombereich: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%].



		gra									
Instanz- parame- ter	Werte	SGS-gra				SGS-arb				CP	
		MS		SA		MS		SA		CS	WS
		lin	step	lin	step	lin	step	lin	step	step	step
1	alle	<b>55,21</b>	23,96	30,21	20,83	38,54	37,50	34,38	32,29	38,54	41,67
2	<i>I</i>	30	45,83	29,17	43,75	29,17	52,08	50,00	45,83	45,83	52,08
3		120	<b>64,58</b>	18,75	16,67	12,50	25,00	25,00	22,92	18,75	25,00
4	<i>INV</i>	200	<b>52,08</b>	16,67	25,00	16,67	27,08	25,00	22,92	18,75	27,08
5		1000	<b>58,33</b>	31,25	35,42	25,00	50,00	50,00	45,83	45,83	50,00
6	<i>NREL</i>	50	62,50	41,67	37,50	35,42	64,58	62,50	58,33	54,17	64,58
7		200	<b>47,92</b>	6,25	22,92	6,25	12,50	12,50	10,42	10,42	12,50
8	<i>PREL</i>	0,25	56,25	35,42	37,50	31,25	62,50	62,50	58,33	54,17	62,50
9		1	<b>54,17</b>	12,50	22,92	10,42	14,58	12,50	10,42	10,42	14,58
10	<i>RFP</i>	0	<b>81,25</b>	43,75	56,25	41,67	39,58	37,50	33,33	33,33	39,58
11		0,5	29,17	4,17	4,17	0,00	37,50	37,50	35,42	31,25	37,50
12	<i>DUR</i>	<i>dur</i> <sub>1</sub>	<b>56,25</b>	21,88	34,38	21,88	40,63	37,50	37,50	34,38	40,63
13		<i>dur</i> <sub>2</sub>	43,75	21,88	21,88	18,75	37,50	37,50	34,38	31,25	37,50
14		<i>dur</i> <sub>3</sub>	<b>65,63</b>	28,13	34,38	21,88	37,50	37,50	31,25	31,25	37,50

Aus Darstellungsgründen wird die Tabelle unten fortgesetzt.

		agg													
		SGS-agg-d				SGS-agg-i				SGS-arb				CP	
		MS		SA		MS		SA		MS		SA		CS	WS
		lin	step	lin	step	lin	step	lin	step	lin	step	lin	step	step	step
1		30,21	26,04	32,29	27,08	50,00	35,42	48,96	38,54	37,50	36,46	39,58	34,38	42,71	44,79
2		33,33	27,08	33,33	27,08	<b>72,92</b>	45,83	66,67	52,08	54,17	50,00	56,25	50,00	54,17	52,08
3		27,08	25,00	31,25	27,08	27,08	25,00	31,25	25,00	20,83	22,92	22,92	18,75	31,25	37,50
4		27,08	20,83	27,08	20,83	50,00	27,08	43,75	27,08	25,00	22,92	29,17	18,75	31,25	31,25
5		33,33	31,25	37,50	33,33	50,00	43,75	54,17	50,00	50,00	50,00	50,00	50,00	54,17	<b>58,33</b>
6		47,92	45,83	50,00	47,92	66,67	62,50	<b>70,83</b>	64,58	58,33	60,42	62,50	56,25	<b>70,83</b>	66,67
7		12,50	6,25	14,58	6,25	33,33	8,33	27,08	12,50	16,67	12,50	16,67	12,50	14,58	22,92
8		47,92	41,67	50,00	43,75	66,67	58,33	<b>72,92</b>	62,50	62,50	60,42	64,58	56,25	62,50	64,58
9		12,50	10,42	14,58	10,42	33,33	12,50	25,00	14,58	12,50	12,50	14,58	12,50	22,92	25,00
10		47,92	41,67	47,92	41,67	52,08	39,58	47,92	39,58	39,58	37,50	43,75	33,33	43,75	50,00
11		12,50	10,42	16,67	12,50	47,92	31,25	<b>50,00</b>	37,50	35,42	35,42	35,42	35,42	41,67	39,58
12		31,25	21,88	31,25	25,00	50,00	34,38	50,00	37,50	40,63	37,50	43,75	34,38	37,50	43,75
13		28,13	25,00	28,13	25,00	50,00	31,25	<b>53,13</b>	37,50	40,63	37,50	40,63	37,50	43,75	46,88
14		31,25	31,25	37,50	31,25	50,00	40,63	43,75	40,63	31,25	34,38	34,38	31,25	46,88	43,75

**Tabelle 6.3:** Experimente mit Fokus auf den Stoffstrombereich: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%].

dieser Arbeit nicht mehr weiter auf die lineare Programmierung eingegangen. Obwohl es nicht zur Lösung eingesetzt wird, erfüllt das in Abschnitt 5.9 vorgestellte lineare Programm für das (RCPSP/M) dennoch den Zweck, das bessere Problemverständnis zu fördern.

### 6.3.2 Auswertung hinsichtlich der Transformationsschemas

Die Aggregation von Stoffstromvorgängen hat Auswirkungen auf die Realitätstreue einer Instanz  $\mathcal{I}_{\text{RCPSP}/M}$ . Dies liegt darin begründet, dass aggregierte Stoffstromvorgänge nicht unterbrochen werden können und damit im Gegensatz zu granularen Stoffstromvorgängen die Flexibilität bei der Zeit- und Kapazitätsplanung für die Stoffverarbeitung eingeschränkt wird. Da das aggregierte Transformationsschema so aufgebaut wurde, dass jede Lösung für eine aggregierte Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$  in eine zulässige Lösung für die zugrundeliegende granulare Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  rücktransformiert werden kann, lässt sich folgende theoretische Aussage treffen: Unter Annahme unbegrenzt verfügbarer Rechenleistung ist die optimale Lösung zur Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-gra}}$  immer genauso gut oder besser als die optimale Lösung zur zugehörigen Instanz  $\mathcal{I}_{\text{RCPSP}/M\text{-agg}}$ . Rein modellbedingt ist daher zu erwarten, dass die Anwendung des granularen Transformationsschemas in Rechenexperimenten zu besseren Ergebnissen als die Anwendung des aggregierten Transformationsschemas führt.

In Abschnitt 6.2.2 wurde angegeben, dass für die Menge  $\mathcal{T}_1$  die granularen Instanzen zwischen 1073 und 76 257 Stoffstromvorgänge und die aggregierten Instanzen zwischen 106 und 2348 Stoffstromvorgänge umfassen. Es ist ersichtlich, dass diese Anzahl durch die Aggregation beträchtlich reduziert wurde – im paarweisen Vergleich über alle Instanzen um durchschnittlich 93 %. Da der Rechenaufwand für die Generierungsschemas mit der Anzahl der Stoffstromvorgänge skaliert, ist zu erwarten, dass für aggregierte Instanzen deutlich schneller zulässige Lösungen gefunden werden können als für granulare Instanzen. Bei Anwendung einer Metaheuristik lässt sich folglich erwarten, dass der Lösungsraum einer aggregierten Instanz schneller abgesucht werden kann. Für den CP Optimizer kann ohne Kenntnis der verwendeten (proprietären) Lösungsmechanismen vorab keine Aussage getroffen werden, wie sich die Anzahl der Stoffstromvorgänge auf die Effizienz auswirkt. Es ist jedoch zu vermuten, dass größere Probleminstanzen auch hier zu Effizienzeinbußen führen.

Die Angaben in Tabelle 6.2 suggerieren, dass die Lösungsstrategien für aggregierte Instanzen besser abschneiden. Die insgesamt beste Lösungsstrategie ist demnach (agg|SGS-agg-i|MS|lin) mit einer durchschnittlichen relativen Abweichung von 4,72 % von der besten unteren Schranke. Als beste Lösungsstrategie für die granularen Instanzen lässt sich (gra|SGS-agg-d|MS|lin) mit einer durchschnittlichen relativen Abweichung von 9,11 % auslesen. Auch bei einem Vergleich der beiden methodisch verwandten Lösungsstrategien (gra|SGS-gra|MS|lin) und (agg|SGS-agg-d|MS|lin) (beide verwenden den Dekompositionsansatz) lässt sich erkennen, dass für die aggregierten Instanzen mit 8,98 % ein geringfügig besseres Ergebnis erzielt wurde als mit 9,11 % für die granularen Instanzen.

Während (agg|SGS-agg-i|MS|lin) innerhalb des zehnminütigen Zeitlimits je Testinstanz durchschnittlich 272 641 Iterationen (= Anzahl der Aufrufe des Generierungsschemas und damit Anzahl berechneter Schedules) ausführen konnte, waren dies für (gra|SGS-gra|MS|lin) lediglich 29 034. Diese Werte bestätigen die Erwartung, dass in aggregierten Instanzen ein größerer Teil des Lösungsraums abgesucht werden kann.

Der CP Optimizer konnte mit einem Kaltstart nur für 53 der 96 granularen Instanzen eine zulässige Lösung finden. Für die aggregierten Instanzen konnte er hingegen stets eine zulässige Lösung finden. Selbst unter Verwendung des Warmstarts konnte nur für 92 granulare Instanzen eine zulässige Lösung gefunden werden. Dies ist insofern überraschend, da der CP Optimizer beim Warmstart eine zulässige Startlösung übergeben bekommt. Offensichtlich gelingt es dem Solver nicht, diese zulässige Startlösung innerhalb des zehnminütigen Zeitlimits zu verarbeiten. Das Protokoll liefert hierzu keine hilfreichen Informationen, sondern gibt lediglich „Using starting point solution“ und „Search terminated by limit, no solution found“ aus. Dass mit dem CP Optimizer zahlreiche granulare Instanzen ungelöst bleiben, ist in Tabelle 6.2 an den häufigen Einträgen der Form „-“ in den Spalten zu (gra|CP|·|step) zu erkennen. Diese Beobachtungen bestätigen die getroffene Vermutung, dass auch die Leistungsfähigkeit des CP Optimizers von der Anzahl der Stoffstromvorgänge abhängt.

Die Angaben in Tabelle 6.3 verändern das aus Tabelle 6.2 gewonnene Bild: Die insgesamt beste Lösungsstrategie ist hier (gra|SGS-gra|MS|lin) mit 55,21 % der besten gefundenen Lösungen, d. h., das granulare Transformationsschema schneidet besser ab. Dies unterstreicht, dass die Werte in Tabelle 6.2 kritisch interpretiert werden müssen, da sie von Ausreißern beeinflusst werden können. Das binäre Maß in Tabelle 6.3 wird nicht durch Ausreißer beeinflusst, ermöglicht jedoch auch keine Aussage zur absoluten Qualität der gefundenen Lösungen. Keines der verwendeten Maße kann somit als aussagekräftiger als das andere eingestuft werden.

Mit einem Blick in die nach den einzelnen Werten der Instanzparameter gefilterten Zeilen der Tabellen 6.2 und 6.3 können fundiertere Aussagen zum Nutzen der beiden Transformationsschemas getroffen werden. Es lässt sich erkennen, dass für  $I = 30$  das aggregierte Transformationsschema in beiden Tabellen besser abschneidet, wohingegen für  $I = 120$  das granulare Transformationsschema in beiden Tabellen besser abschneidet. Für *NREL* und *PREL* zeigt sich in Tabelle 6.3, dass für die jeweils niedrigeren Werte  $NREL = 50$  und  $PREL = 0,25$  das aggregierte Transformationsschema besser ist als das granulare. Für  $NREL = 200$  und  $PREL = 1$  kehrt sich diese Rangfolge um. Basierend auf diesen Beobachtungen kann die Aussage getroffen werden, dass das aggregierte Transformationsschema eher gewählt werden sollte, je kleiner die Anzahl an Rückbauvorgängen und je kleiner die Anzahl an Stofffreisetzungen ist. Hier fallen die nachteiligen Auswirkungen der Aggregation auf die Realitätstreue nicht derart stark ins Gewicht als bei hoher Anzahl an Rückbauvorgängen und hohen Stofffreisetzungen.

Des Weiteren lässt sich für *RFP* anhand beider Tabellen ein eindeutiger Trend erkennen: Während für  $RFP = 0$  das granulare Transformationsschema besser abschneidet, ist für  $RFP = 0,5$  das aggregierte Transformationsschema klar überlegen. Das heißt, je höher die

gemeinsame erneuerbare Ressourcennutzung durch Rückbau- und Stoffstromvorgänge, desto eher sollte das aggregierte Transformationsschema gewählt werden. Diese Aussage steht auch in Einklang mit den theoretischen Darlegungen in Abschnitt 4.2.4. Sobald Rückbauvorgänge und Stoffstromvorgänge erneuerbare Ressourcen gemeinsam nutzen, sinkt die Flexibilität zur Einplanung der Stoffstromvorgänge (vgl. Beispiele 4.4 und 4.6). Die erneuerbaren Restriktionen unterbinden ohnehin, dass die Stoffverarbeitung parallel zum Rückbau ausgeführt werden kann. Ob nach einem Rückbauvorgang mehrere granulare Stoffstromvorgänge nacheinander oder ein einzelner aggregierter Stoffstromvorgang starten, macht keinen Unterschied. Daher ist bei  $RFP = 0,5$  eher zu erwarten, dass die Aggregation keine negativen Auswirkungen auf die Realitätstreue hat, was die Rechenergebnisse auch bestätigen.

Zusammenfassend lässt sich festhalten, dass die im Gesamtdurchschnitt ambivalenten Ergebnisse beider Tabellen die Komplexität des Spannungsfeldes zwischen Realitätstreue und Instanzgröße bestätigen (vgl. Abschnitt 4.2.4). Es kann in den durchgeführten Rechenexperimenten keine generelle Überlegenheit eines der beiden vorgeschlagenen Transformationsschemas festgestellt werden. Dies zeigt auch ein direkter Vergleich: Für 29 Testinstanzen wurde mit der granularen Instanz eine echt bessere Lösung als mit der aggregierten Instanz gefunden. Für 27 Testinstanzen hingegen wurde mit der aggregierten Instanz eine echt bessere Lösung gefunden. Für die verbleibenden 40 Testinstanzen wurde mit beiden Instanzen jeweils eine gleich gute Lösung gefunden. Es sind daher die individuellen Instanzcharakteristika heranzuziehen, um zwischen dem granularen und aggregierten Transformationsschema zu wählen. Unabhängig von der Lösungsqualität haben die Rechenexperimente gezeigt, dass das aggregierte Transformationsschema zu bevorzugen ist, falls die schnelle Erzeugung einer zulässigen Lösung im Vordergrund steht. Steht lediglich der CP Optimizer als Lösungsverfahren zur Verfügung, ist ebenfalls das aggregierte Transformationsschema zu bevorzugen.

Generell ist erneut anzumerken, dass das granulare und das aggregierte Transformationsschema lediglich zwei Vorschläge für konkrete Ausgestaltungsmöglichkeiten eines Transformationsschemas sind und es sich auch nur um ein Hilfsmittel bei der Modellierung eines realen Problems als Instanz des (RCPSP/M) handelt. Falls in der Realität beispielsweise die Möglichkeiten zur Unterbrechung der Stoffverarbeitung eingeschränkt sind, muss das granulare Transformationsschema entsprechend angepasst werden. Aufgrund der Abhängigkeit von den individuellen Instanzcharakteristika empfiehlt es sich in jedem Fall, verschiedene Ausgestaltungsmöglichkeiten zu testen und dabei auch solche Transformationsschemas zu berücksichtigen, die in dieser Arbeit nicht algorithmisch ausformuliert wurden (z. B. vertikale Aggregation; vgl. Abschnitt 4.2.5).

### 6.3.3 Auswertung hinsichtlich der Lösungsverfahren

#### Vergleich der neu entwickelten Lösungsverfahren für granulare Instanzen

Zur Lösung von granularen Instanzen stehen das speziell für granulare Instanzen entwickelte Lösungsverfahren SGS-gra und das Lösungsverfahren für beliebige Instanzen SGS-arb zur Verfügung. In Tabelle 6.2 zeigt sich, dass die Lösungsstrategien (gra|SGS-arb|·|·) für keinen der betrachteten Parameterwerte in der Lage waren, sämtliche granularen Instanzen zu lösen. Daher sind in den entsprechenden Spalten ausschließlich Einträge der Form „-“ vorhanden. Insgesamt wurde mit SGS-arb innerhalb des zehnminütigen Zeitlimits für 38 granulare Instanzen keine zulässige Lösung gefunden. Dabei handelt es sich durchweg um große Instanzen mit mehr als 18 000 Stoffstromvorgängen. SGS-gra konnte hingegen für alle granularen Instanzen eine zulässige Lösung finden.

Für einen Vergleich der beiden Lösungsverfahren ist wegen der ungelösten Instanzen das binäre Maß in Tabelle 6.3 besser geeignet. Die beste Lösungsstrategie für granulare Instanzen ist demnach (gra|SGS-gra|MS|lin) mit 55,21 % der besten gefundenen Lösungen. Die beste Lösungsstrategie unter Verwendung des Lösungsverfahrens SGS-arb ist (gra|SGS-arb|MS|lin) mit 38,54 % der besten gefundenen Lösungen.

Anhand der Durchschnittswerte in Tabelle 6.3 und aufgrund der Tatsache, dass mit SGS-arb zahlreiche granulare Instanzen ungelöst bleiben, kann festgestellt werden, dass SGS-gra im Durchschnitt das bessere der beiden Lösungsverfahren ist. Es zeigt sich, dass der Dekompositionsansatz, welcher darin umgesetzt ist, zuverlässig zulässige Lösungen finden kann. Das liegt daran, dass Stoffstromvorgänge entlang ihrer stoffstrominduzierten Zeitstruktur eingeplant werden. Auslagernde Stoffstromvorgänge folgen in dieser Struktur auf einlagernde Stoffstromvorgänge, wodurch das kumulative Ressourcenprofil bei der Einplanung niedrig gehalten wird. Beim Integrationsansatz, welcher in SGS-arb umgesetzt ist, können die Positionen ein- und auslagernder Stoffstromvorgänge in der Vorgangsfolge  $\ell$  hingegen weit auseinanderfallen. Dann werden bei eng bindenden kumulativen Ressourcenrestriktionen umfangreiche Ausplanungsschritte erforderlich. Diese Ausplanungsschritte erhöhen den Rechenaufwand beträchtlich. So wurden innerhalb des zehnminütigen Zeitlimits mit der Lösungsstrategie (gra|SGS-gra|MS|lin) durchschnittlich 29 034 Iterationen pro Testinstanz durchgeführt, während dies bei (gra|SGS-arb|MS|lin) lediglich 1753 waren.

Mit Blick auf die einzelnen Werte der Instanzparameter lässt sich aus Tabelle 6.3 ablesen, dass  $I$ ,  $PREL$  und insbesondere  $RFP$  einen Einfluss auf die Rangfolge von SGS-gra und SGS-arb haben: Für granulare Instanzen mit  $I = 30$  und mit  $PREL = 0,25$  schneidet SGS-arb besser ab als SGS-gra. Hier ist der zusätzliche Rechenaufwand durch die Ausplanungsschritte aufgrund der geringeren Anzahl an Rückbau- und Stoffstromvorgängen noch handhabbar. Für höhere Werte dieser beiden Parameter ist SGS-gra leistungsfähiger. Für granulare Instanzen mit  $RFP = 0$  liefert SGS-gra eindeutig bessere Ergebnisse, wohingegen sich bei  $RFP = 0,5$  SGS-arb als das bessere Lösungsverfahren herausstellt. Bei  $RFP = 0,5$  macht sich die durch den Integrationsansatz in SGS-arb ermöglichte Flexibilität bei der Einplanung von Vorgängen positiv bemerkbar: Während in SGS-gra die Reihenfolge für

die Einplanung von Stoffstromvorgängen durch die stoffstrominduzierte Zeitstruktur starr vorgegeben ist, können in SGS-*arb* Stoffstromvorgänge sowohl untereinander als auch mit Rückbauvorgängen vertauscht werden (solange Vorrang- und Zeitrestriktionen eingehalten sind). Durch das Vertauschen kann hinsichtlich der erneuerbaren Ressourcenrestriktionen, welche bei  $RFP = 0,5$  stärker binden, ein größerer Teil des Lösungsraums abgesucht werden. Wenn beispielsweise angenommen wird, dass alle Rückbau- und Stoffstromvorgänge eine erneuerbare Ressource der Kapazität eins nutzen, so berechnet SGS-*gra* einen Schedule, in dem immer ein Rückbauvorgang und dann die Stoffstromvorgänge seiner Stoffstromstruktur sequenziell aufeinanderfolgen. SGS-*arb* kann (unter Annahme ausreichender kumulativer Ressourcenkapazitäten) hingegen einen Schedule berechnen, in dem zuerst alle Rückbauvorgänge und dann alle Stoffstromvorgänge eingeplant sind. Ein solcher Schedule hat einen deutlich besseren Zielfunktionswert, da im (RCPSP/M) nur die Rückbauvorgänge in die Zielfunktion zählen. Zusammengefasst sollte auch SGS-*arb* als geeignetes Lösungsverfahren für granulare Instanzen in Betracht gezogen werden, sofern die kumulativen im Vergleich zu den erneuerbaren Ressourcenrestriktionen weniger stark binden. Im Extremfall, d. h. ohne kumulative Ressourcenrestriktionen, würden sich die beiden Lösungsverfahren nur noch dadurch unterscheiden, dass SGS-*arb* den vollständigen Lösungsraum absucht und SGS-*gra* nicht.

Ein direkter Vergleich zeigt ebenfalls, dass SGS-*arb* von Vorteil sein kann: Für 54 granulare Instanzen wurde mit SGS-*gra* eine echt bessere Lösung als mit SGS-*arb* gefunden. Umgekehrt wurde jedoch mit SGS-*arb* für 21 granulare Instanzen eine echt bessere Lösung als mit SGS-*gra* gefunden. Für die übrigen 21 granularen Instanzen schneiden beide Lösungsverfahren gleich gut ab. Bei den 21 granularen Instanzen, für die SGS-*arb* besser abschneidet, handelt es sich durchweg um Instanzen mit  $RFP = 0,5$ .

### Vergleich der entwickelten Lösungsverfahren für aggregierte Instanzen

Zur Lösung von aggregierten Instanzen stehen die speziell für aggregierte Instanzen entwickelten Lösungsverfahren unter Verwendung eines Generierungsschemas nach einem Dekompositionsansatz SGS-*agg-d* und nach einem Integrationsansatz SGS-*agg-i* zur Verfügung. Daneben kann auch das Lösungsverfahren für beliebige Instanzen SGS-*arb* eingesetzt werden. In Tabelle 6.2 ist klar erkennbar, dass selbst die beste Lösungsstrategie unter allen Strategien ( $agg|SGS-agg-d|\cdot|\cdot$ ) mit 8,89 % schlechter ist als die schlechteste Lösungsstrategie unter allen ( $agg|SGS-agg-i|\cdot|\cdot$ ) mit 7,19 %. Das heißt, SGS-*agg-i* dominiert SGS-*agg-d* im Gesamtdurchschnitt. Mit 4,72 % durchschnittlicher Abweichung von der besten unteren Schranke ist ( $agg|SGS-agg-i|MS|lin$ ) die beste Lösungsstrategie für aggregierte Instanzen. Des Weiteren ist in Tabelle 6.2 ersichtlich, dass die Lösungsstrategien ( $agg|SGS-arb|\cdot|\cdot$ ) für keinen der betrachteten Parameterwerte in der Lage waren, sämtliche aggregierten Instanzen zu lösen. Daher sind in den entsprechenden Spalten ausschließlich Einträge der Form „-“ vorhanden. Insgesamt wurde mit SGS-*arb* innerhalb des zehnminütigen Zeitlimits für 30 aggregierte Instanzen keine zulässige Lösung gefunden. Dabei handelt es sich durchweg um große Instanzen mit mehr als 500 Stoffstromvorgängen.

SGS-agg-d und SGS-agg-i konnten hingegen für alle aggregierten Instanzen eine zulässige Lösung finden.

Die erste Zeile in Tabelle 6.3 bestätigt die im Durchschnitt dominierende Stellung von SGS-agg-i gegenüber SGS-agg-d. Mit 50,00 % der besten gefundenen Lösungen ist (agg|SGS-agg-i|MS|lin) analog zu Tabelle 6.2 die beste Lösungsstrategie für aggregierte Instanzen. In Tabelle 6.3 kann ferner festgestellt werden, dass SGS-arb insgesamt besser als SGS-agg-d abschneidet, jedoch nicht so gut wie SGS-agg-i.

Beide Tabellen zeigen deutlich, dass SGS-agg-i im Durchschnitt das beste der drei neu entwickelten Lösungsverfahren ist, mit welchen aggregierte Instanzen gelöst werden können. Dabei ist anzumerken, dass SGS-agg-d aufgrund des Dekompositionsansatzes, bei dem Stoffstromvorgänge entlang der stoffstrominduzierten Zeitstruktur eingeplant werden, deutlich mehr Iterationen schafft als SGS-agg-i. So konnte beispielsweise die Lösungsstrategie (agg|SGS-agg-d|MS|lin) innerhalb des zehnminütigen Zeitlimits im Durchschnitt 1 245 726 Schedules berechnen und (agg|SGS-agg-i|MS|lin) lediglich 272 641. Dennoch schneiden die Lösungsstrategien mit SGS-agg-i besser ab. Wie bereits beim Vergleich der Lösungsverfahren in Abschnitt 5.6 dargelegt, ermöglicht der Integrationsansatz das Durchsuchen des gesamten Lösungsraums, während der Dekompositionsansatz nur Teile davon absucht. Die Rechenexperimente dieses Kapitels belegen, dass beim Integrationsansatz der Vorteil der globaleren Suche den Nachteil des höheren Rechenaufwands durch häufigere Ausplanungsschritte (im Durchschnitt) überwiegt.

Obwohl SGS-arb ebenfalls einen Integrationsansatz implementiert, schneidet es erkennbar schlechter ab als SGS-agg-i. Auch der direkte Vergleich zeigt, dass SGS-arb für keine der 96 aggregierten Testinstanzen eine echt bessere Lösung finden konnte als SGS-agg-i. Grund dafür ist die höhere Effizienz des speziell auf aggregierte Instanzen zugeschnittenen SGS-agg-i gegenüber SGS-arb. So konnte die Lösungsstrategie (agg|SGS-arb|MS|lin) innerhalb des zehnminütigen Zeitlimits nur durchschnittlich 6916 Schedules berechnen (im Vergleich zu den genannten 272 641 Schedules mit (agg|SGS-agg-i|MS|lin)). Die Effizienzunterschiede ergeben sich daraus, dass kumulative Ressourcenkonflikte in SGS-agg-i immer erst nach Einplanung aller relevanten Nachfolger geprüft werden (vgl. Abschnitt 5.6). In SGS-arb erfolgt die Prüfung hingegen nach jeder Einplanung eines Vorgangs. Außerdem sind in SGS-agg-i keine aufwendigen Berechnungen zur Bestimmung der relevanten Vorgänger- und Nachfolgermengen erforderlich (vgl. `RelevanteNachfolgerArb`), sondern diese werden der stoffstrominduzierten Zeitstruktur entnommen.

Ob SGS-agg-d oder SGS-arb als besser eingestuft werden kann, ist nicht eindeutig zu beantworten. Hinsichtlich der gefundenen zulässigen Lösungen ist SGS-agg-d überlegen. Hinsichtlich des Anteils bester gefundener Lösungen ist SGS-arb überlegen.

Aus beiden Tabellen lässt sich ablesen, dass der Unterschied in der Leistungsfähigkeit von SGS-agg-d und SGS-agg-i insbesondere von den Instanzparametern  $I$  und  $RFP$  abhängt. Während für  $I = 30$  oder  $RFP = 0,5$  SGS-agg-i deutlich besser abschneidet, ist der Unterschied bei  $I = 120$  oder  $RFP = 0$  eher gering. Die Rangfolge kann sich sogar umkehren, wie beispielsweise in Tabelle 6.2 bei  $I = 120$  im Vergleich von (agg|SGS-agg-d|

MS|lin) (5,70 %) und (agg|SGS-agg-i|MS|lin) (5,98 %) ersichtlich. Für geringe Werte von  $I$  kann der Integrationsansatz mit seiner globaleren Suche innerhalb von zehn Minuten einen vergleichsweise großen Teil des Lösungsraums absuchen und liefert daher sehr gute Ergebnisse. Für hohe Werte von  $I$  hingegen ist der Lösungsraum so groß, dass der Integrationsansatz innerhalb von zehn Minuten trotz seines theoretischen Potenzials keine globale Suche schafft. Daher nähern sich die Leistungsfähigkeiten von SGS-agg-d und SGS-agg-i an. Für  $RFP = 0,5$  gilt, wie bei den granularen Instanzen bereits erläutert, dass das flexible Vertauschen von Vorgängen im Integrationsansatz hinsichtlich der erneuerbaren Ressourcenrestriktionen zu besseren Lösungen führt.

Zusammengefasst werden folgende Empfehlungen festgehalten: Falls genügend Rechenzeit zur Verfügung steht, ist SGS-agg-i gegenüber SGS-agg-d zu bevorzugen. Falls die Rechenzeit beschränkt ist, müssen die individuellen Instanzcharakteristika miteinbezogen werden – insbesondere die Anzahl der Rückbauvorgänge und die Knappheit der erneuerbaren Ressourcen. Dann kann gegebenenfalls SGS-agg-d vorteilhaft sein. Dies unterstreicht auch ein direkter Vergleich: Für 36 aggregierte Instanzen wurde mit SGS-gra-i eine echt bessere Lösung als mit SGS-agg-d gefunden. Umgekehrt wurde jedoch mit SGS-agg-d für 28 aggregierte Instanzen eine echt bessere Lösung als mit SGS-agg-i gefunden. Für die übrigen 32 aggregierten Instanzen schneiden beide Lösungsverfahren gleich gut ab. Da mit SGS-agg-i bereits ein speziell für aggregierte Instanzen entwickeltes Lösungsverfahren zur Verfügung steht, das den Integrationsansatz implementiert, ist SGS-arb (das ebenfalls einen Integrationsansatz implementiert) in keinem der betrachteten Fälle überlegen. SGS-arb bietet jedoch die Möglichkeit, auch Instanzen des (RCPSP/M) zu lösen, die keine stoffstrominduzierte Zeitstruktur aufweisen. Derartige Instanzen werden in den Rechenexperimenten dieser Arbeit nicht betrachtet.

### Vergleich der neu entwickelten Lösungsverfahren mit dem CP Optimizer

Der CP Optimizer kann lediglich für alle aggregierten Instanzen innerhalb des zehnminütigen Zeitlimits eine zulässige Lösung finden. Die beiden entsprechenden Lösungsstrategien (agg|CP|CS|step) und (agg|CP|WS|step) erzielen dabei gemäß erster Zeile in Tabelle 6.2 durchschnittliche relative Abweichungen von 5,28 % und 5,30 % von der besten unteren Schranke. Diese Werte sind besser als sämtliche Durchschnittsergebnisse der Lösungsstrategien mit SGS-gra und SGS-agg-d. Lediglich die Lösungsstrategien (agg|SGS-agg-i|·|lin) schneiden gemäß Tabelle 6.2 besser ab als der CP Optimizer.

Tabelle 6.3 zeigt ein leicht verändertes Bild. Hier ist neben den Lösungsstrategien (agg|SGS-agg-i|·|lin) auch die Lösungsstrategie (gra|SGS-gra|MS|lin) besser als der CP Optimizer. Der direkte Vergleich zeigt, dass mit dem CP Optimizer nur für acht Testinstanzen eine echt bessere Lösung als mit den neu entwickelten Lösungsverfahren gefunden werden konnte. Umgekehrt konnte mit den neu entwickelten Lösungsverfahren für 48 Testinstanzen eine echt bessere Lösung gefunden werden. Für 40 Testinstanzen waren die Ergebnisse gleich gut.



Da der CP Optimizer ein exaktes Lösungsverfahren darstellt, können auch Optimalitätsbetrachtungen miteinbezogen werden. Die Lösungsstrategie (gra|CP|WS|step) konnte für 34 granulare Instanzen die Optimalität beweisen. Die Lösungsstrategie (agg|CP|WS|step) konnte für 38 aggregierte Instanzen die Optimalität beweisen. In beiden Fällen handelt es sich jedoch ausschließlich um Instanzen, für die  $z = LB_{PSPLIB}$  gilt, d. h., in denen die kumulativen Ressourcenrestriktionen vergleichsweise wenig binden.

Überraschend ist, dass der Warmstart des CP Optimizers gemäß Tabelle 6.2 für die aggregierten Instanzen um 0,02 Prozentpunkte schlechter abschneidet als der Kaltstart. Durch die vorhandene Startlösung beim Warmstart wäre stattdessen zu erwarten, dass bessere Ergebnisse geliefert werden. Laut Tabelle 6.3 ist der Warmstart wieder geringfügig besser. Insgesamt lässt sich folgern, dass der Warmstart bei aggregierten Instanzen kaum Vorteile bringt. Bei granularen Instanzen hilft er insofern, als gegenüber dem Kaltstart für 39 zusätzliche Instanzen eine zulässige Lösung gefunden werden konnte. Die final vom CP Optimizer ausgegebenen Lösungen sind hierfür im Durchschnitt jedoch nur um 1,10 % besser als die Startlösung. Daher ist auch bei granularen Instanzen der Nutzen des Warmstarts als gering zu bewerten.

Hinsichtlich der Parameterwerte ist erkennbar, dass der CP Optimizer sowohl in Tabelle 6.2 als auch in Tabelle 6.3 für  $NREL = 50$  das beste Lösungsverfahren ist. Zusätzlich ist er in Tabelle 6.3 für  $INV = 1000$  das beste Lösungsverfahren. Auch für  $PREL = 0,25$  lassen sich aus beiden Tabellen sehr gute Werte für den CP Optimizer abgelesen. Für die gegenteiligen Parameterwerte  $NREL = 200$ ,  $INV = 200$  und  $PREL = 1$  schneidet er hingegen deutlich schlechter ab. Daraus lässt sich folgern, dass der CP Optimizer im Vergleich zu den neu entwickelten Lösungsverfahren besonders leistungsfähig ist, wenn die kumulativen Ressourcenrestriktionen nicht so stark binden (d. h., wenn  $INV$  hoch ist und  $NREL$  und  $PREL$  niedrig sind). Umgekehrt bestätigt sich, dass die neu entwickelten, problemspezifischen Lösungsverfahren insbesondere bei großen Engpässen im Stoffstrombereich qualitativ hochwertige Lösungen berechnen können.

Lediglich die neu entwickelten Lösungsverfahren sind in der Lage, Instanzen mit gleichmäßigen Freisetzungen und Nutzungen über die Vorgangsausführung zu lösen (vgl. Tabellenspalten mit `lin`). Der CP Optimizer kann aufgrund fehlender Modellierungskonzepte in der Concert Technology-Bibliothek nur die Stufenvarianten der Testinstanzen lösen (vgl. Tabellenspalten mit `step`). In den Stufenvarianten wird jedoch, wie in Abschnitt 5.8 erläutert, künstlich Lagerplatz blockiert. Ein Vergleich des CP Optimizers mit den anderen Lösungsverfahren, wie er in den Tabellen 6.2 und 6.3 erfolgt, ist korrekt, falls davon ausgegangen wird, dass Freisetzungen und Nutzungen in der Realität gleichmäßig erfolgen. Dann ist der durch die Stufenvariante erlittene Qualitätsverlust dem CP Optimizer zuzurechnen, da er diese behelfsmäßige Modellierung aufgrund fehlender Schnittstellen erzwingt. Erfolgen die Freisetzungen und Nutzungen in der Praxis jedoch zum Start und zum Ende der Vorgänge, müssen die Spalten mit `lin` für den Vergleich ausgeschlossen werden. Dann ist ein *modellbereinigter Vergleich* des CP Optimizers mit den neu entwickelten Lösungsverfahren möglich. Das heißt, dass die Lösungsverfahren dann lediglich an ihrer Leistungsfähigkeit und nicht an dem mit ihnen abbildbaren Modellumfang gemessen werden.

Für den modellbereinigten Vergleich können in Tabelle 6.2 die Spalten mit `lin` übergangen und stattdessen ausschließlich Spalten mit `step` verglichen werden. Zur besseren Übersicht sind diese Spalten nochmals in Tabelle 6.4 zusammengefasst. Darunter schneidet die Lösungsstrategie (`agg|CP|CS|step`) mit 5,28 % relativer Abweichung durchschnittlich am besten ab. An zweiter Stelle steht (`agg|SGS-agg-i|SA|step`) mit 7,04 %. Auch für die einzelnen Parameterwerte sind die Lösungsstrategien mit dem CP Optimizer überlegen, außer für  $I = 120$  und  $RFP = 0$ . Hier schneidet jeweils (`gra|SGS-gra|MS|step`) besser ab.

Aufgrund der Berechnungsweise der Maße in Tabelle 6.3 ist für einen modellbereinigten Vergleich eine Neuberechnung erforderlich, bei der die Lösungen der Lösungsstrategien mit `lin` nicht für  $z_{best}$  berücksichtigt werden. Die Ergebnisse dieser Neuberechnung sind in Tabelle 6.5 dargestellt. Darin sind die Lösungsstrategien mit SA (Simulated-Annealing-Metaheuristik) zur Verkürzung der Darstellung ausgeblendet. Sie haben in keiner Zeile am besten abgeschnitten. Auch aus Tabelle 6.5 lässt sich ablesen, dass der CP Optimizer im modellbereinigten Vergleich den anderen Lösungsverfahren durchschnittlich überlegen ist. Der Kaltstart und der Warmstart sind mit 55,21 % der besten gefundenen Lösungen gleichauf. In den weiteren Zeilen der Tabelle zeigt sich dennoch, dass für einige der Parameterwerte die Lösungsstrategie (`gra|SGS-gra|MS|step`) die besseren Lösungen liefert. Hierbei handelt es sich um Parameterwerte, die zu eher großen Instanzen mit eng bindenden kumulativen Ressourcenrestriktionen führen. Im direkten modellbereinigten Vergleich lässt sich feststellen, dass für 28 Testinstanzen mit dem CP Optimizer eine echt bessere Lösung gefunden werden konnte als mit den neu entwickelten Lösungsverfahren. Umgekehrt konnte mit den neu entwickelten Lösungsverfahren für 26 Testinstanzen eine echt bessere Lösung gefunden werden. Für 42 Instanzen wurde jeweils dieselbe Lösung gefunden.

Zusammenfassend lässt sich unter Einbeziehung aller Lösungsstrategien festhalten, dass weder die generelle Überlegenheit des CP Optimizers noch eines der neu entwickelten Lösungsverfahren festgestellt werden kann. Insbesondere die Tatsache, dass die neu entwickelten Lösungsverfahren auch mit gleichmäßigen Freisetzungen und Nutzungen über die Vorgangsausführung umgehen können, steigert ihren Nutzen erheblich. Außerdem schneiden sie bei eng bindenden kumulativen Ressourcenrestriktionen besser ab. Sind hingegen eher die erneuerbaren Ressourcenrestriktionen ausschlaggebend und erfolgen die Freisetzungen und kumulativen Ressourcennutzungen zum Vorgangsstart und -ende, ist der CP Optimizer zu bevorzugen. Dies wird durch den modellbereinigten Vergleich, bei dem ausschließlich die Lösungsstrategien mit `step` verglichen werden, bestätigt. Außerdem bietet der CP Optimizer als exaktes Lösungsverfahren im Vergleich zu den heuristischen Lösungsverfahren den Vorteil, die Optimalität beweisen zu können. Dies war jedoch für die verwendeten Testinstanzen nur möglich, wenn  $z = LB_{PSPLIB}$  galt.

## CPLEX

In Abschnitt 6.3.1 wurden die Ergebnisse einer Stichprobe vorgestellt, in der zwei aggregierte Instanzen mit CPLEX gelöst werden sollten. Die dabei gemessenen sehr hohen

Instanz- parame- ter	Werte	gra				agg				
		SGS-gra	SGS-arb	CP		SGS-agg-d	SGS-agg-i	SGS-arb	CP	
				MS	MS				CS	WS
step	step	step	step	step	step	step	step	step	step	step
alle		9,83	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	10,59	7,19	- <sup>a</sup>	<b>5,28</b>	5,30
<i>I</i>	30	13,71	- <sup>a</sup>	- <sup>a</sup>	8,33	14,41	6,81	- <sup>a</sup>	<b>4,58</b>	4,65
	120	<b>5,94</b>	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	6,77	7,57	- <sup>a</sup>	5,98	5,94
<i>INV</i>	200	11,68	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	13,15	10,89	- <sup>a</sup>	<b>8,16</b>	8,24
	1000	7,98	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	8,03	3,49	- <sup>a</sup>	2,40	<b>2,35</b>
<i>NREL</i>	50	7,73	- <sup>a</sup>	- <sup>a</sup>	4,36	7,45	3,58	- <sup>a</sup>	<b>2,29</b>	2,30
	200	11,93	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	13,73	10,80	- <sup>a</sup>	<b>8,27</b>	8,29
<i>PREL</i>	0,25	8,26	- <sup>a</sup>	- <sup>a</sup>	5,58	9,17	5,94	- <sup>a</sup>	<b>4,54</b>	4,54
	1	11,39	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	12,01	8,44	- <sup>a</sup>	<b>6,03</b>	6,05
<i>RFP</i>	0	<b>4,83</b>	- <sup>a</sup>	- <sup>a</sup>	5,32	6,15	6,46	- <sup>a</sup>	5,26	5,29
	0,5	14,83	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	15,03	7,93	- <sup>a</sup>	<b>5,30</b>	5,30
<i>DUR</i>	<i>dur</i> <sub>1</sub>	7,30	- <sup>a</sup>	- <sup>a</sup>	4,11	7,69	4,45	- <sup>a</sup>	<b>2,63</b>	2,68
	<i>dur</i> <sub>2</sub>	11,12	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	11,94	8,75	- <sup>a</sup>	<b>6,98</b>	7,00
	<i>dur</i> <sub>3</sub>	11,06	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	12,14	8,38	- <sup>a</sup>	6,24	<b>6,20</b>

<sup>a</sup>Es wurde nicht für alle Instanzen eine zulässige Lösung gefunden.

**Tabelle 6.4:** Experimente mit Fokus auf den Stoffstrombereich: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%]; modellbereinigter Vergleich.

Instanz- parame- ter	Werte	gra				agg				
		SGS-gra	SGS-arb	CP		SGS-agg-d	SGS-agg-i	SGS-arb	CP	
				MS	MS				CS	WS
step	step	step	step	step	step	step	step	step	step	step
alle		50,00	37,50	38,54	46,88	27,08	35,42	36,46	<b>55,21</b>	<b>55,21</b>
<i>I</i>	30	43,75	50,00	52,08	62,50	27,08	45,83	50,00	<b>70,83</b>	66,67
	120	<b>56,25</b>	25,00	25,00	31,25	27,08	25,00	22,92	39,58	43,75
<i>INV</i>	200	<b>50,00</b>	25,00	27,08	37,50	20,83	27,08	22,92	45,83	43,75
	1000	50,00	50,00	50,00	56,25	33,33	43,75	50,00	64,58	<b>66,67</b>
<i>NREL</i>	50	56,25	62,50	64,58	70,83	45,83	62,50	60,42	<b>75,00</b>	68,75
	200	<b>43,75</b>	12,50	12,50	22,92	8,33	8,33	12,50	35,42	41,67
<i>PREL</i>	0,25	45,83	62,50	62,50	72,92	43,75	58,33	60,42	<b>75,00</b>	<b>75,00</b>
	1	<b>54,17</b>	12,50	14,58	20,83	10,42	12,50	12,50	35,42	35,42
<i>RFP</i>	0	<b>72,92</b>	37,50	39,58	54,17	41,67	39,58	37,50	52,08	54,17
	0,5	27,08	37,50	37,50	39,58	12,50	31,25	35,42	<b>58,33</b>	56,25
<i>DUR</i>	<i>dur</i> <sub>1</sub>	<b>56,25</b>	37,50	40,63	46,88	25,00	34,38	37,50	46,88	46,88
	<i>dur</i> <sub>2</sub>	31,25	37,50	37,50	46,88	25,00	31,25	37,50	62,50	<b>65,63</b>
	<i>dur</i> <sub>3</sub>	<b>62,50</b>	37,50	37,50	46,88	31,25	40,63	34,38	56,25	53,13

**Tabelle 6.5:** Experimente mit Fokus auf den Stoffstrombereich: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%]; modellbereinigter Vergleich.

Rechenzeiten zeigen, dass die lineare Programmierung in Verbindung mit der in Abschnitt 5.9 vorgestellten zeitindizierten Formulierung kein geeignetes Lösungsverfahren für die betrachteten Testinstanzen ist. Angesichts der Tatsache, dass  $LB_{PSPLIB} \geq 2400$  für alle Testinstanzen, ist diese Erkenntnis naheliegend. Die hochaufgelösten Zeithorizonte führen in Kombination mit der zeitindizierten Formulierung zu einer sehr hohen Anzahl an binären Entscheidungsvariablen  $x_{it}$ . Gegebenenfalls wären mit anderen Ansätzen zur linearen Formulierung von Schedulingproblemen (vgl. Artigues u. a. 2015) bessere Ergebnisse zu erzielen. Alternativ könnte die Anzahl der Entscheidungsvariablen mittels Herunterskalieren des Zeithorizonts verringert werden, wodurch jedoch die Dauern der Stoffstromvorgänge verfälscht werden können. Denn diese Dauern waren, wie in Abschnitt 6.2.2 beschrieben, der Grund dafür, weshalb der Zeithorizont in den Testinstanzen um den Faktor 50 hochskaliert werden musste.

### 6.3.4 Auswertung hinsichtlich der Metaheuristiken

In diesem Abschnitt werden die Lösungsstrategien paarweise unter Austausch von MS und SA verglichen (d. h. (gra|SGS-gra|MS|lin) mit (gra|SGS-gra|SA|lin), etc.). Für granulare Instanzen schneidet MS in jedem paarweisen Vergleich, sowohl in Tabelle 6.2 (sofern kein Eintrag der Form „-“ vorhanden ist) als auch Tabelle 6.3 durchschnittlich besser ab. Für aggregierte Instanzen ist hingegen keine klare Überlegenheit einer Metaheuristik erkennbar. So ist (agg|SGS-agg-d|MS|lin) gemäß Tabelle 6.2 besser als (agg|SGS-agg-d|SA|lin). In Tabelle 6.3 ist die Rangfolge dieser beiden Lösungsstrategien jedoch umgekehrt. Es kann weder insgesamt für die aggregierten Instanzen noch für ein einzelnes Lösungsverfahren eine eindeutige Aussage hinsichtlich einer besser geeigneten Metaheuristik getroffen werden.

Im direkten Vergleich lässt sich feststellen, dass für 41 granulare Instanzen mit MS eine echt bessere Lösung als mit SA gefunden werden konnte. Umgekehrt konnte mit SA für elf granulare Instanzen eine echt bessere Lösung gefunden werden. Für 53 granulare Instanzen wurde dieselbe Lösung gefunden. Wie auch schon aus den Tabellen abgeleitet, sollte demnach die Multi-Start-Metaheuristik für granulare Instanzen bevorzugt werden. Bei den aggregierten Instanzen wurde für 20 Instanzen mit MS eine echt bessere Lösung gefunden und für 13 Instanzen mit SA. Für die verbleibenden 63 aggregierten Instanzen wurde dieselbe Lösung gefunden. Es ist folglich weiterhin eine höhere Leistungsfähigkeit bei der Multi-Start-Metaheuristik zu beobachten. Der Unterschied ist jedoch geringfügiger. Werden die Ergebnisse für granulare und aggregierte Instanzen gemeinsam betrachtet, lässt sich feststellen, dass SA nur für sechs der 96 Testinstanzen eine echt bessere Lösung als MS gefunden hat.

Ob MS oder SA gewählt wird, hat nur einen geringen Einfluss auf die Anzahl der Iterationen, die mit den jeweiligen Lösungsstrategien durchgeführt werden konnten. Beispielsweise konnten innerhalb des zehnminütigen Zeitlimits mit der Lösungsstrategie (gra|SGS-gra|MS|lin) durchschnittlich 29 034 Schedules pro Testinstanz berechnet werden. Mit der Lösungsstrategie (gra|SGS-gra|SA|lin) waren dies 30 747 Schedules. Dies lässt sich darauf

zurückführen, dass die Metaheuristik im Vergleich zu den Generierungsschemas nur einen geringen Anteil der verfügbaren Rechenzeit beansprucht.

Mit Blick auf die einzelnen Werte der Instanzparameter ist in den Tabellen 6.2 und 6.3 kein besonders hervorzuhebender Einfluss auf die Leistungsfähigkeit der Metaheuristik ersichtlich.

Die Multi-Start-Metaheuristik sucht den Lösungsraum in großen Schritten ab, während die Simulated-Annealing-Metaheuristik kleine Schritte macht und auch Verschlechterungen zulässt. Den Rechenexperimenten zufolge ist es in granularen Instanzen von Vorteil, den Lösungsraum in großen Schritten abzusuchen. Diese Beobachtung kann auch dadurch zustande kommen, dass je Instanz nur ein zehnminütiges Zeitlimit zur Verfügung stand. Für höhere Zeitlimits könnte das kleinteiligere Absuchen des Lösungsraums mit der Simulated-Annealing-Metaheuristik zu besseren Ergebnissen führen. In aggregierten Instanzen hat die verwendete Metaheuristik einen geringeren Einfluss auf die Ergebnisse. Dies kann an dem ohnehin kleineren Lösungsraum liegen, der aus der Aggregation und damit der geringeren Anzahl an Stoffstromvorgängen folgt.

Es sei darauf verwiesen, dass zur Kalibrierung der Simulated-Annealing-Metaheuristik die Wertebereiche für die Verfahrensparameter gemäß Abschnitt 6.3.1 eingeschränkt wurden. Außerdem wurden nur zehn zufällig gewählte Testinstanzen verwendet und damit wurde für jede Kombination der Verfahrensparameter nur eine 60-sekündige Berechnung gestartet. Durch umfangreichere Kalibrierungsexperimente ließen sich möglicherweise bessere Verfahrensparameter berechnen, die die Leistungsfähigkeit der Simulated-Annealing-Metaheuristik erhöhen. Es ist eine Abwägung zwischen erwünschter Qualität der Kalibrierung und hinnehmbarem Rechenaufwand zu treffen. Gegebenenfalls kann auch spezielle Software hinzugezogen werden, die die Kalibrierung steuert (vgl. z. B. Hübner u. a. 2021, S. 284f).

### 6.3.5 Auswertung hinsichtlich der Freisetzungs- bzw. Nutzungsart

In diesem Abschnitt werden die Lösungsstrategien paarweise unter Austausch von `lin` und `step` verglichen (d. h. `(gra|SGS-gra|MS|lin)` mit `(gra|SGS-gra|MS|step)`, etc.). Dabei zeigt sich, dass die Verwendung von gleichmäßigen Freisetzungen und kumulativen Ressourcennutzungen über die Vorgangsdauer (d. h. `lin`) in allen Lösungsstrategien sowohl gemäß Tabelle 6.2 als auch Tabelle 6.3 durchschnittlich besser abschneidet als die Verwendung der Stufenvariante (d. h. `step`). Dies ist nicht überraschend, da in der Stufenvariante zusätzlicher Lagerplatz blockiert wird und somit jede Lösung zur Stufenvariante eine obere Schranke für die beste Lösung zur originalen Instanz beschreibt (vgl. Abschnitt 5.8).

Das Ausmaß der Unterschiede zwischen `lin` und `step` variiert abhängig vom verwendeten Lösungsverfahren. Für `SGS-arb` und `SGS-agg-d` ist der Unterschied eher gering. Für `SGS-gra` und für `SGS-agg-i` ist der Unterschied deutlicher. So berechnet beispielsweise die Lösungsstrategie `(agg|SGS-agg-i|MS|lin)` eine durchschnittliche relative Abweichung

von 4,72 % von der besten unteren Schranke und findet 50,00 % der besten gefundenen Lösungen. Für die Lösungsstrategie (agg|SGS-agg-i|MS|step) betragen diese Werte 7,19 % und 35,42 %.

Ob `lin` oder `step` gewählt wird, hat nur einen geringen Einfluss auf die Anzahl der Iterationen, die mit den jeweiligen Metaheuristiken durchgeführt werden konnten. Beispielsweise konnten innerhalb des zehnminütigen Zeitlimits mit der Lösungsstrategie (agg|SGS-agg-i|MS|lin) durchschnittlich 29 034 Schedules pro Testinstanz berechnet werden. Mit der Lösungsstrategie (agg|SGS-agg-i|MS|step) waren dies 32 213 Schedules. Für die Stufenvariante sind die Lösungsverfahren demnach etwas effizienter, da sich die Freisetzungen bzw. Nutzungen auf weniger Zeitpunkte erstrecken.

Ein eindeutiger Einfluss der einzelnen Parameterwerte auf die mit `lin` und `step` erzielte Lösungsqualität lässt sich nicht erkennen. In Tabelle 6.2 liegen die Differenzen der Werte in den Spalten `lin` und `step` in allen Zeilen stets im unteren einstelligen Prozentbereich. In Tabelle 6.3 liegen die Differenzen für SGS-`arb` und SGS-`agg-d` meist im einstelligen Prozentbereich. Für SGS-`gra` und SGS-`agg-i` hingegen lassen sich auch größere zweistellige Differenzen beobachten (z. B. hat für  $RFP = 0$  (gra|SGS-`gra`|MS|lin) 81,25 % der besten gefundenen Lösungen gefunden und (gra|SGS-`gra`|MS|step) nur 43,75 %). Für die beiden Parameterwerte  $I = 120$  und  $NREL = 50$  schneidet die Lösungsstrategie (agg|SGS-`arb`|MS|lin) in Tabelle 6.3 überraschenderweise geringfügig schlechter ab als (agg|SGS-`arb`|MS|step). Hier handelt es sich vermutlich um Zufallseinflüsse.

Im direkten Vergleich lässt sich feststellen, dass mit `lin` für 48 Testinstanzen eine echt bessere Lösung als mit `step` gefunden wurde. Umgekehrt wurde mit `step` für acht Testinstanzen eine echt bessere Lösung gefunden. Für 40 Testinstanzen unterscheidet sich die beste gefundene Lösung nicht. Basierend darauf wird schlussgefolgert, dass die gewählte Freisetzungs- und Nutzungsart einen klaren Einfluss auf die Rechenergebnisse hat. Daher sollte bei der Modellierung einer realen Problemstellung gewissenhaft geklärt werden, ob Stofffreisetzungen und -bewegungen gleichmäßig über die Vorgangsausführung oder jeweils zum Start bzw. Ende erfolgen. Für die Testinstanzen zeigt sich jedoch auch, dass das Ausmaß dieses Einflusses hinsichtlich der durchschnittlichen relativen Abweichungen von der unteren Schranke im unteren einstelligen Prozentbereich liegt (vgl. Tabelle 6.2). Dies liegt an der Vielzahl der in den Testinstanzen vorhandenen Restriktionen. Wären ausschließlich kumulative Ressourcenrestriktionen vorhanden, ist davon auszugehen, dass die Differenz erheblicher wäre. Schließlich kann die Stufenvariante theoretisch deutlich höhere kumulative Ressourcenprofile verursachen, wie in Beispiel 5.13 in Abschnitt 5.8 zu erkennen war.

Der Vergleich zwischen `lin` und `step` konnte ausschließlich basierend auf den neu entwickelten Lösungsverfahren angestellt werden. Für den CP Optimizer stehen die erforderlichen Modellierungskonzepte nicht zur Verfügung. Gäbe es solche Konzepte für den CP Optimizer, wird vermutet, dass die von ihm gefundenen Lösungen nochmal um wenige Prozentpunkte besser wären als die bisher für die Stufenvariante gefundenen Lösungen.

### 6.3.6 Langzeitexperimente

Um zu beurteilen, wie die Lösungsverfahren für längere Rechenzeiten als die bisher verwendeten zehn Minuten abschneiden, werden zusätzliche *Langzeitexperimente* durchgeführt. Damit der Rechenaufwand in einem praktikablen Rahmen bleibt, werden hierfür drei stellvertretende Lösungsstrategien ausgewählt:

- Die Lösungsstrategie (gra|CP|WS|step) wird gewählt, um zu beurteilen, ob der CP Optimizer mit längeren Rechenzeiten für alle granularen Instanzen eine zulässige Lösung finden kann. Außerdem soll untersucht werden, ob für die gefundenen Lösungen weitere Verbesserungen erzielt werden können. Bisher wurde beobachtet, dass der CP Optimizer sowohl im Vergleich aller Lösungsstrategien als auch im modellbereinigten Vergleich für granulare Instanzen schlechter abschneidet als die neu entwickelten Lösungsverfahren.
- Die Lösungsstrategie (agg|CP|WS|step) wird gewählt, um zu beurteilen, ob der CP Optimizer in seiner besten Lösungsstrategie weitere Verbesserungen erzielen kann, wenn längere Rechenzeiten erlaubt werden. Hinsichtlich der durchschnittlichen Abweichung von der besten unteren Schranke ist der Warmstart zwar geringfügig schlechter als der Kaltstart (vgl. Tabelle 6.2). Hinsichtlich des Anteils bester gefundener Lösungen ist er jedoch besser (vgl. Tabelle 6.3), weshalb der Warmstart für die Langzeitexperimente gewählt wird.
- Die Lösungsstrategie (agg|SGS-agg-i|MS|lin) wird gewählt, um zu beurteilen, ob die neu entwickelten Lösungsverfahren in ihrer besten Lösungsstrategie (hinsichtlich der durchschnittlichen Abweichung von der besten unteren Schranke; vgl. Tabelle 6.2) weitere Verbesserungen erzielen können, wenn längere Rechenzeiten erlaubt werden.

Für jede der 96 Testinstanzen aus der Menge  $\mathcal{T}_1$  und jede der drei ausgewählten Lösungsstrategien wird eine 60-minütige Berechnung ( $timeLimit = 3600$  s) durchgeführt. Die Ergebnisse lassen sich wie folgt zusammenfassen:

- Die Lösungsstrategie (gra|CP|WS|step) konnte die nach zehn Minuten gefundene Lösung für 30 granulare Instanzen verbessern. Die prozentualen Verbesserungen je Instanz bewegen sich zwischen null und 18,96 %, wo bei es sich bei der Verbesserung um 18,96 % um einen von zwei Ausreißern handelt. Dennoch zeigt sich, dass längere Rechenzeiten für granulare Instanzen sinnvoll sein können. Im Durchschnitt schneidet die Lösungsstrategie (gra|CP|WS|step) jedoch auch in den Langzeitexperimenten nicht besser ab als die ihr zuvor bereits überlegenen Lösungsstrategien. Dies liegt auch daran, dass für die sechs größten granularen Instanzen mit über 75 000 Stoffstromvorgängen der CP Optimizer aufgrund von Speichermangel abgestürzt ist. Damit ist die Anzahl der zulässig gelösten Instanzen in den Langzeitexperimenten mit 90 sogar geringer als in den zehnminütigen Experimenten, wo 92 Instanzen gelöst werden konnten.

- Die Lösungsstrategie (agg|CP|WS|step) konnte die nach zehn Minuten gefundene Lösung für 26 aggregierte Instanzen verbessern. Die prozentualen Verbesserungen je Instanz bewegen sich zwischen null und 1,32 %. Die durchschnittliche Abweichung von der besten unteren Schranke verringert sich von 5,30 % in den zehnminütigen Experimenten auf 5,21 % in den Langzeitexperimenten. Die nur geringfügigen Verbesserungen lassen darauf schließen, dass längere Rechenzeiten als zehn Minuten für die aggregierten Testinstanzen und den CP Optimizer nur überschaubare Vorteile bringen.
- Die Lösungsstrategie (agg|SGS-agg-i|MS|lin) konnte die nach zehn Minuten gefundene Lösung für 35 aggregierte Instanzen verbessern. Die prozentualen Verbesserungen je Instanz bewegen sich zwischen null und 3,34 %. Die durchschnittliche Abweichung von der besten unteren Schranke verringert sich von 4,72 % in den zehnminütigen Experimenten auf 4,49 % in den Langzeitexperimenten. Auch hier sind demnach bei längeren Rechenzeiten nur geringfügige Verbesserungen zu erwarten.

Die Feststellung, dass die Lösungsstrategien bereits nach zehn Minuten Lösungen finden, die sehr nahe an den nach 60 Minuten gefundenen Lösungen sind, unterstreicht ihre Praxisauglichkeit. So erhalten Projektverantwortliche innerhalb relativ kurzer Rechenzeiten einen qualitativ hochwertigen Schedule. Dies erleichtert insbesondere eine Sensitivitätsanalyse: Problemparameter, wie beispielsweise bestimmte Lager- oder Verarbeitungskapazitäten, können abgeändert werden und anschließend lässt sich ohne umfangreichen Zeitaufwand ein neuer Schedule berechnen. Dadurch kann methodisch fundiert abgeschätzt werden, wie sich Veränderungen in den Rahmenbedingungen, wie z. B. ein Ausfall einer Anlage, auf den Projektablauf auswirken. Das Optimierungswerkzeug erfüllt mit diesen Möglichkeiten das eingangs formulierte Ziel, Entscheidungen im Zusammenhang mit Investitionen oder dem Baustellenlayout zu unterstützen.

### 6.4 Experimente mit Fokus auf den Rückbaubereich

Die nachfolgend vorgestellten Experimente fokussieren sich darauf, verschiedene Einflüsse auf die Lösbarkeit des (RMCPSP) unter Variation der Charakteristika des Rückbaubereichs zu untersuchen. Daher werden die Testinstanzen der Menge  $\mathcal{T}_2$  verwendet, in denen die Instanzparameter des Rückbaubereichs variabel gestaltet sind (vgl. Abschnitt 6.1.2).

Als Testumgebung wurde derselbe Rechner wie für die Experimente mit Fokus auf den Stoffstrombereich verwendet (vgl. Abschnitt 6.3).

Zunächst werden in Abschnitt 6.4.1 der Ablauf der Rechenexperimente und eine Ergebnisübersicht vorgestellt. Im darauffolgenden Abschnitt 6.4.2 werden die Ergebnisse ausgewertet.



### 6.4.1 Ablauf und Ergebnisübersicht

In den Experimenten mit Fokus auf den Stoffstrombereich wurde festgestellt, dass die Simulated-Annealing-Metaheuristik (SA) im Durchschnitt schlechter abschneidet als die Multi-Start-Metaheuristik (MS). Da ein Vergleich der beiden Metaheuristiken nun nicht mehr im Fokus steht, werden Lösungsstrategien mit SA hier ausgeschlossen. Dies ermöglicht eine Reduktion des Rechenaufwands, insbesondere da keine erneute Kalibrierung erforderlich ist, und gleichzeitig eine übersichtlichere Ergebnisdarstellung.

Des Weiteren steht auch ein Vergleich der verschiedenen Freisetzungs- und Nutzungsarten (`lin` und `step`) nicht mehr im Fokus. Daher werden Lösungsstrategien mit `lin` hier ausgeschlossen. Dabei handelt es sich zwar um die laut vorangegangenen Experimenten besseren Lösungsstrategien. Sie ermöglichen jedoch keinen modellbereinigten Vergleich mit dem CP Optimizer. Somit ist es sinnvoller, die Lösungsstrategien mit `step` beizubehalten.

Für MultiStart wird wie in den vorangegangenen Experimenten die Prioritätsregel  $\pi$  auf Smallest Latest Start Time und der *perturbFactor* auf 0,4 gesetzt. Für jede der 96 Testinstanzen in der Menge  $\mathcal{T}_2$  und jede (verbleibende) Lösungsstrategie wird eine zehninminütige Berechnung (*timeLimit* = 600 s) durchgeführt. Die Ergebnisse sind in den Tabellen 6.6 und 6.7 dargestellt, welche analog zu den vorangegangenen Tabellenpaaren aufgebaut sind und auch dieselben Maße verwenden.

### 6.4.2 Auswertung

Im Vergleich zu den Rechenexperimenten mit Fokus auf den Stoffstrombereich können anhand der Tabellen 6.6 und 6.7 die Auswirkungen der Instanzparameter *NC*, *RS* und *RF* auf die Leistungsfähigkeit der Lösungsverfahren untersucht werden. In Tabelle 6.6 lässt sich mit Blick auf die entsprechenden Zeilen kaum ein Einfluss dieser Instanzparameter erkennen. Die Lösungsstrategien (`agg|CP|·|step`) schneiden für alle Werte der genannten Instanzparameter am besten ab. Lediglich die Instanzparameter *I* und *RFP* haben einen deutlich erkennbaren Einfluss, welcher jedoch bereits in Abschnitt 6.3.3 festgestellt wurde.

Aus Tabelle 6.7 lässt sich ablesen, dass SGS-gra für *NC* = 1,5 mehr beste bekannte Lösungen gefunden hat als der CP Optimizer. Für *NC* = 2,1 kehrt sich diese Rangfolge um. Der Instanzparameter *NC* gibt die durchschnittliche Anzahl an nichtredundanten Vorrangbeziehungen je Rückbauvorgang an. Das heißt, für niedrigere Werte von *NC* liegt ein höherer Grad an Parallelität in der Vorrangstruktur der Rückbauvorgänge vor. SGS-gra kann den Ergebnissen zufolge mit dieser höheren Parallelität besser umgehen als der CP Optimizer, wobei die Unterschiede eher geringfügig sind. Ein möglicher Grund ist, dass die verwendete Multi-Start-Metaheuristik den im Fall einer hohen Parallelität sehr großen Lösungsraum schneller absucht. Für den Instanzparameter *RS* ist kein wesentlicher Einfluss erkennbar. Für *RF* = 0,25 zeigt sich in Tabelle 6.7, dass SGS-gra und der CP Optimizer gleichauf sind, wohingegen für *RF* = 0,5 der CP Optimizer das bessere Lösungsverfahren ist. Demzufolge kann SGS-gra mit einer geringeren Anzahl an echt positiven erneuerbaren Ressourcennutzungen vergleichsweise besser umgehen als mit einer höheren

Instanz- parame- ter	Werte	gra				agg				
		SGS-gra	SGS-arb	CP		SGS-agg-d	SGS-agg-i	SGS-arb	CP	
				MS step	MS step				CS step	WS step
alle		15,38	<sup>-a</sup>	<sup>-a</sup>	14,36	15,53	11,82	<sup>-a</sup>	8,95	<b>8,90</b>
<i>I</i>	30	20,82	23,21	<sup>-a</sup>	16,35	20,14	11,14	<sup>-a</sup>	<b>7,77</b>	7,82
	120	<b>9,95</b>	<sup>-a</sup>	<sup>-a</sup>	12,37	10,93	12,49	<sup>-a</sup>	10,13	9,98
<i>NC</i>	1,5	17,04	<sup>-a</sup>	<sup>-a</sup>	16,72	17,10	14,16	<sup>-a</sup>	10,57	<b>10,44</b>
	2,1	13,72	<sup>-a</sup>	<sup>-a</sup>	12,00	13,96	9,47	<sup>-a</sup>	<b>7,32</b>	7,36
<i>RS</i>	0,2	17,60	<sup>-a</sup>	<sup>-a</sup>	15,33	17,06	12,39	<sup>-a</sup>	9,61	<b>9,47</b>
	0,5	13,17	<sup>-a</sup>	<sup>-a</sup>	13,39	14,01	11,24	<sup>-a</sup>	<b>8,28</b>	8,33
<i>RF</i>	0,25	15,65	<sup>-a</sup>	<sup>-a</sup>	16,53	16,14	13,47	<sup>-a</sup>	9,74	<b>9,57</b>
	0,5	15,12	<sup>-a</sup>	<sup>-a</sup>	12,20	14,93	10,17	<sup>-a</sup>	<b>8,16</b>	8,23
<i>RFP</i>	0	<b>5,91</b>	<sup>-a</sup>	<sup>-a</sup>	7,07	7,15	8,04	<sup>-a</sup>	6,43	6,38
	0,5	24,86	<sup>-a</sup>	<sup>-a</sup>	21,65	23,92	15,59	<sup>-a</sup>	11,47	<b>11,42</b>
<i>DUR</i>	<i>dur</i> <sub>1</sub>	13,31	<sup>-a</sup>	<sup>-a</sup>	10,41	13,16	8,67	<sup>-a</sup>	5,64	<b>5,51</b>
	<i>dur</i> <sub>2</sub>	14,20	<sup>-a</sup>	<sup>-a</sup>	13,56	14,93	11,92	<sup>-a</sup>	<b>9,38</b>	9,42
	<i>dur</i> <sub>3</sub>	18,64	<sup>-a</sup>	<sup>-a</sup>	19,12	18,51	14,86	<sup>-a</sup>	11,82	<b>11,78</b>

<sup>a</sup>Es wurde nicht für alle Instanzen eine zulässige Lösung gefunden.

**Tabelle 6.6:** Experimente mit Fokus auf den Rückbaubereich: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%].

Instanz- parame- ter	Werte	gra				agg				
		SGS-gra	SGS-arb	CP		SGS-agg-d	SGS-agg-i	SGS-arb	CP	
				MS step	MS step				CS step	WS step
alle		41,67	2,08	8,33	16,67	12,50	14,58	3,13	<b>44,79</b>	42,71
<i>I</i>	30	35,42	4,17	16,67	33,33	25,00	29,17	6,25	<b>60,42</b>	<b>60,42</b>
	120	<b>47,92</b>	0,00	0,00	0,00	0,00	0,00	0,00	29,17	25,00
<i>NC</i>	1,5	<b>41,67</b>	2,08	2,08	14,58	8,33	4,17	0,00	39,58	37,50
	2,1	41,67	2,08	14,58	18,75	16,67	25,00	6,25	<b>50,00</b>	47,92
<i>RS</i>	0,2	39,58	2,08	10,42	14,58	16,67	20,83	4,17	<b>43,75</b>	<b>43,75</b>
	0,5	43,75	2,08	6,25	18,75	8,33	8,33	2,08	<b>45,83</b>	41,67
<i>RF</i>	0,25	<b>43,75</b>	2,08	10,42	20,83	10,42	10,42	4,17	<b>43,75</b>	<b>43,75</b>
	0,5	39,58	2,08	6,25	12,50	14,58	18,75	2,08	<b>45,83</b>	41,67
<i>RFP</i>	0	<b>56,25</b>	4,17	16,67	33,33	25,00	20,83	6,25	50,00	43,75
	0,5	27,08	0,00	0,00	0,00	0,00	8,33	0,00	39,58	<b>41,67</b>
<i>DUR</i>	<i>dur</i> <sub>1</sub>	34,38	6,25	12,50	18,75	12,50	15,63	9,38	43,75	<b>46,88</b>
	<i>dur</i> <sub>2</sub>	15,63	0,00	6,25	18,75	3,13	6,25	0,00	<b>53,13</b>	40,63
	<i>dur</i> <sub>3</sub>	<b>75,00</b>	0,00	6,25	12,50	21,88	21,88	0,00	37,50	40,63

**Tabelle 6.7:** Experimente mit Fokus auf den Rückbaubereich: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%].

Anzahl. Diese Beobachtung steht in Einklang mit bereits gewonnenen Erkenntnissen: Je mehr die erneuerbaren im Vergleich zu den kumulativen Ressourcenrestriktionen in den Hintergrund treten, desto stärker können die neu entwickelten Lösungsverfahren ihre Leistungsfähigkeit zur Geltung bringen. Umgekehrt ist bei eng bindenden erneuerbaren Ressourcenrestriktionen der CP Optimizer im Vorteil.

Ungeachtet der festgestellten Unterschiede für die einzelnen Parameterwerte lässt sich zusammenfassen, dass die Charakteristika des Rückbaubereichs keinen wesentlichen Einfluss auf die Leistungsfähigkeit der Lösungsstrategien haben. Stattdessen sollten schwerpunktmäßig die Charakteristika des Stoffstrombereichs betrachtet werden, um die am besten geeignete Lösungsstrategie auszuwählen. Aufgrund der Tatsache, dass die Anzahl der Stoffstromvorgänge sowohl bei den aggregierten als auch insbesondere bei granularen Testinstanzen die Anzahl der Rückbauvorgänge deutlich überwiegt, ist diese Erkenntnis erwartbar.

## 6.5 Experimente mit Fokus auf die Zielsetzung

Wie aus der Problemformulierung des (RMCPSP) hervorgeht, wurde als Zielfunktion  $z(S) := S_{I+1}$  definiert, d. h., der Zielfunktionswert entspricht der Dauer des Rückbauprojekts. Dies führt dazu, dass Stoffstromvorgänge im (RCPSP/M) nicht direkt über Zeitbeziehungen in die Zielfunktion eingehen und somit als nicht kritisch gelten (vgl. Abschnitt 4.3.2). In anderen Worten ist es hinsichtlich der Zeitbeziehungen unerheblich, wie lange die Stoffverarbeitung über das Ende des Rückbauprojekts hinausgeht. Die Stoffverarbeitung beeinflusst die Zielfunktion jedoch indirekt, da bei Lagerengpässen Rückbauvorgänge verzögert werden können, wodurch sich die Projektdauer erhöhen kann. Die Wahl der Zielfunktion  $z(S) := S_{I+1}$  wurde mit dem hierarchischen Modellierungsansatz begründet, bei dem primär der Rückbaubereich und sekundär der Stoffstrombereich betrachtet wird.

In den nachfolgend vorgestellten Experimenten wird die Zielfunktion im (RMCPSP) durch  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  ersetzt. Durch Einbeziehung des fiktiven End-Stoffstromvorgangs  $(J+1)^o$  werden die Stoffstromvorgänge darin gleichrangig mit den Rückbauvorgängen behandelt. Da es sich bei  $I+1$  und  $(J+1)^o$  um die beiden Senken im Vorgangsknotenplan handelt, können nun Rückbau- und Stoffstromvorgänge kritisch sein, d. h. bei einer Verzögerung direkt eine Erhöhung des Zielfunktionswertes verursachen. Somit steht die Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  in Einklang mit der in der Literatur üblichen Zielsetzung zur Minimierung der Projektdauer, in welcher die Projektdauer als Zeitspanne vom Start des frühesten Vorgangs bis zum Ende des spätesten Vorgangs gilt.

Die nachfolgenden Experimente sollen einerseits demonstrieren, dass die in dieser Arbeit betrachteten Lösungsverfahren auch für die Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  geeignet sind. Außerdem soll im Vergleich der Ergebnisse dieses Abschnitts mit den Ergebnissen des Abschnitts 6.3 der Einfluss der Zielfunktion auf die Leistungsfähigkeit der Lösungsstrategien untersucht werden. Um diesen Vergleich zu ermöglichen, werden die Testinstanzen der Menge  $\mathcal{T}_1$  verwendet.

Als Testumgebung wurde derselbe Rechner wie für die Experimente mit Fokus auf den Stoffstrombereich verwendet (vgl. Abschnitt 6.3).

Zunächst werden in Abschnitt 6.5.1 der Ablauf der Rechenexperimente und eine Ergebnisübersicht vorgestellt. Im darauffolgenden Abschnitt 6.5.2 werden die Ergebnisse ausgewertet.

### 6.5.1 Ablauf und Ergebnisübersicht

Da ein Vergleich der Metaheuristiken und der Freisetzungs- bzw. Nutzungsarten nicht mehr im Vordergrund steht, wird auf die Lösungsstrategien mit SA und `lin` verzichtet. Damit werden auch eine Verringerung des Rechenaufwands und eine übersichtlichere Ergebnisdarstellung ermöglicht.

Für `MultiStart` wird wie in den vorangegangenen Experimenten die Prioritätsregel  $\pi$  auf `Smallest Latest Start Time` und der `perturbFactor` auf 0,4 gesetzt. Für jede der 96 Testinstanzen in der Menge  $\mathcal{T}_1$  und jede (verbleibende) Lösungsstrategie wird eine zehnmütige Berechnung (`timeLimit` = 600 s) durchgeführt. Die Umstellung der Zielfunktion ist ohne Weiteres möglich: In den Metaheuristiken wird statt des Zeitpunkts  $S_{I+1}$  der Zeitpunkt  $\max\{S_{I+1}, S_{(J+1)^o}^o\}$  aus den jeweiligen Schedules herangezogen, um deren Qualität zu bewerten. Für den CP Optimizer wird eine zusätzliche, fiktive Vorrangbeziehung  $(I + 1, (J + 1)^o)$  zu  $E$  hinzugefügt. Dann kann die Zielfunktion im Constraintprogramm durch `endOf(act[(J + 1)^o])` ersetzt werden. Die Ergebnisse sind in den Tabellen 6.8 und 6.9 dargestellt, welche analog zu den vorangegangenen Tabellenpaaren aufgebaut sind und auch dieselben Maße verwenden.

Im Zusammenhang mit den unteren Schranken sei darauf hingewiesen, dass die Formel (6.2) zur Berechnung von  $LB_{material}$  aufgrund der veränderten Zielfunktion leicht angepasst werden muss: Der Summand mit den kumulativen Ressourcenkapazitäten und der Summand  $-1$  müssen entfernt werden, da eine Pufferung von Stoffeinheiten nun nicht mehr relevant für die Zielfunktion ist.

### 6.5.2 Auswertung

Sowohl in Tabelle 6.8 als auch in Tabelle 6.9 schneidet die Lösungsstrategie (`gra|SGS-gra|MS|step`) im Gesamtdurchschnitt am besten ab. Mit ihr wird eine durchschnittliche Abweichung von 9,09 % von der besten unteren Schranke erzielt. Für die aggregierten Instanzen schneidet die Lösungsstrategie (`agg|CP|WS|step`) mit einer durchschnittlichen relativen Abweichung von 9,13 % am besten ab. Der CP Optimizer kann bei Verwendung des Kaltstarts nur für 54 der 96 granularen Testinstanzen eine zulässige Lösung finden. Bei Verwendung des Warmstarts erhöht sich diese Zahl auf 92 granulare Instanzen. Die 96 aggregierten Instanzen kann der CP Optimizer sowohl mit dem Kalt- als auch dem Warmstart zulässig lösen. Mit einer durchschnittlichen relativen Abweichung von 9,13 %

Instanz- parame- ter	Werte	gra				agg				
		SGS-gra	SGS-arb	CP		SGS-agg-d	SGS-agg-i	SGS-arb	CP	
		MS step	MS step	CS step	WS step	MS step	MS step	MS step	CS step	WS step
alle		<b>9,09</b>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	10,18	9,86	_ <sup>a</sup>	9,14	9,13
<i>I</i>	30	9,51	_ <sup>a</sup>	_ <sup>a</sup>	<b>8,63</b>	11,18	10,07	_ <sup>a</sup>	9,39	9,37
	120	<b>8,66</b>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	9,17	9,65	_ <sup>a</sup>	8,89	8,90
<i>INV</i>	200	9,27	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	10,15	9,99	_ <sup>a</sup>	9,09	<b>9,08</b>
	1000	<b>8,91</b>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	10,20	9,73	_ <sup>a</sup>	9,19	9,19
<i>NREL</i>	50	6,66	_ <sup>a</sup>	_ <sup>a</sup>	<b>6,19</b>	7,67	7,17	_ <sup>a</sup>	6,51	6,49
	200	<b>11,52</b>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	12,68	12,55	_ <sup>a</sup>	11,77	11,77
<i>PREL</i>	0,25	7,72	_ <sup>a</sup>	_ <sup>a</sup>	<b>7,02</b>	9,30	8,37	_ <sup>a</sup>	7,73	7,73
	1	<b>10,46</b>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	11,05	11,35	_ <sup>a</sup>	10,55	10,54
<i>RFP</i>	0	8,86	_ <sup>a</sup>	_ <sup>a</sup>	<b>8,75</b>	10,23	10,19	_ <sup>a</sup>	9,68	9,68
	0,5	9,31	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	10,13	9,53	_ <sup>a</sup>	8,60	<b>8,58</b>
<i>DUR</i>	<i>dur</i> <sub>1</sub>	<b>1,25</b>	_ <sup>a</sup>	_ <sup>a</sup>	1,38	2,98	2,83	_ <sup>a</sup>	1,42	1,39
	<i>dur</i> <sub>2</sub>	14,71	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	14,74	14,23	_ <sup>a</sup>	<b>13,85</b>	<b>13,85</b>
	<i>dur</i> <sub>3</sub>	<b>11,31</b>	_ <sup>a</sup>	_ <sup>a</sup>	_ <sup>a</sup>	12,81	12,51	_ <sup>a</sup>	12,16	12,16

<sup>a</sup>Es wurde nicht für alle Instanzen eine zulässige Lösung gefunden.

**Tabelle 6.8:** Experimente mit Fokus auf die Zielfunktion: Durchschnittliche Abweichung von der besten unteren Schranke je Lösungsstrategie [%].

Instanz- parame- ter	Werte	gra				agg				
		SGS-gra	SGS-arb	CP		SGS-agg-d	SGS-agg-i	SGS-arb	CP	
		MS step	MS step	CS step	WS step	MS step	MS step	MS step	CS step	WS step
alle		<b>44,79</b>	16,67	34,38	38,54	8,33	15,63	10,42	40,63	40,63
<i>I</i>	30	25,00	25,00	47,92	<b>52,08</b>	10,42	18,75	16,67	43,75	45,83
	120	<b>64,58</b>	8,33	20,83	25,00	6,25	12,50	4,17	37,50	35,42
<i>INV</i>	200	<b>45,83</b>	16,67	31,25	35,42	8,33	16,67	12,50	39,58	39,58
	1000	<b>43,75</b>	16,67	37,50	41,67	8,33	14,58	8,33	41,67	41,67
<i>NREL</i>	50	43,75	33,33	<b>56,25</b>	52,08	12,50	22,92	20,83	33,33	33,33
	200	45,83	0,00	12,50	25,00	4,17	8,33	0,00	<b>47,92</b>	<b>47,92</b>
<i>PREL</i>	0,25	33,33	33,33	<b>52,08</b>	<b>52,08</b>	12,50	25,00	20,83	45,83	45,83
	1	<b>56,25</b>	0,00	16,67	25,00	4,17	6,25	0,00	35,42	35,42
<i>RFP</i>	0	43,75	18,75	47,92	<b>60,42</b>	14,58	16,67	10,42	37,50	35,42
	0,5	<b>45,83</b>	14,58	20,83	16,67	2,08	14,58	10,42	43,75	<b>45,83</b>
<i>DUR</i>	<i>dur</i> <sub>1</sub>	40,63	18,75	43,75	43,75	9,38	15,63	15,63	<b>50,00</b>	<b>50,00</b>
	<i>dur</i> <sub>2</sub>	31,25	18,75	34,38	34,38	6,25	21,88	9,38	<b>50,00</b>	<b>50,00</b>
	<i>dur</i> <sub>3</sub>	<b>62,50</b>	12,50	25,00	37,50	9,38	9,38	6,25	21,88	21,88

**Tabelle 6.9:** Experimente mit Fokus auf die Zielfunktion: Anteil der Instanzen, für die eine Lösungsstrategie die beste bekannte Lösung gefunden hat [%].

von der besten unteren Schranke ist er mit der Lösungsstrategie (agg|CP|WS|step) nur geringfügig schlechter als (gra|SGS-gra|MS|step).

In Tabelle 6.8 ist erkennbar, dass mit der Lösungsstrategie (gra|CP|WS|step) für die Parameterwerte  $I = 30$ ,  $NREL = 50$ ,  $PREL = 0,25$ ,  $RFP = 0$  und  $DUR = dur_1$  alle entsprechenden granularen Testinstanzen zulässig gelöst werden konnten. Abgesehen von  $DUR = dur_1$  sind die Lösungsstrategien dann auch jeweils die besten Lösungsstrategien (d. h., die Einträge sind fettgedruckt). Das lässt darauf schließen, dass der CP Optimizer für Testinstanzen mit wenigen Rückbauvorgängen und wenigen Stofffreisetzungen auch granulare Instanzen mit  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  zuverlässig lösen kann. Auch das Vorhandensein getrennter Ressourcenpools (d. h.  $RFP = 0$ ) erleichtert dem Solver die Suche nach einer zulässigen Lösung. Außerdem zeigt sich, dass Instanzen mit gleicher Dauer aller granularen Stoffstromvorgänge (d. h.  $DUR = dur_1$ ) für den CP Optimizer einfacher zu lösen sind als wenn sich die Dauern unterscheiden.

Für die Parameterwerte  $I = 120$ ,  $INV = 1000$ ,  $PREL = 1$  und  $DUR = dur_3$  ist gemäß beider Tabellen (gra|SGS-gra|MS|step) die beste Lösungsstrategie. Die Lösungsverfahren SGS-arb, SGS-agg-d und SGS-agg-i sind für keinen Parameterwert überlegen.

Werden die Tabellen 6.8 und 6.9 mit den Tabellen 6.4 und 6.5 verglichen, lässt sich der Einfluss der Zielfunktion untersuchen. Es stellt sich heraus, dass der Vorsprung des CP Optimizers durch Ersetzen der Zielfunktion abgenommen hat. Während mit der ursprünglichen Zielfunktion  $z(S) := S_{I+1}$  laut Tabelle 6.4 die Lösungsstrategie (agg|CP|CS|step) im modellbereinigten Vergleich am besten ist, schneidet mit der Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  hingegen (gra|SGS-gra|MS|step) am besten ab. Auch die Lösungsverfahren für aggregierte Instanzen sind hier im Durchschnitt nur geringfügig schlechter als der CP Optimizer. Damit ist belegt, dass die neu entwickelten Lösungsverfahren dieser Arbeit auch und sogar insbesondere für eine Zielfunktion geeignet sind, in der Rückbau- und Stoffstrombereich gleichermaßen einbezogen werden.

Hinsichtlich der Parameterwerte ist der Instanzparameter  $RFP$  hervorzuheben. Mit der ursprünglichen Zielfunktion  $z(S) := S_{I+1}$  hatte er einen deutlichen Einfluss auf die Leistungsfähigkeit von SGS-gra im Vergleich zum CP Optimizer. Für  $RFP = 0,5$  war der CP Optimizer überlegen. Mit der Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  ist dieser Einfluss nicht mehr ersichtlich. Stattdessen ist für  $RFP = 0,5$  in Tabelle 6.9 sogar SGS-gra überlegen. Diese Beobachtung ist nicht überraschend. Der Dekompositionsansatz, welcher dem SGS-gra zugrunde liegt, ermöglicht kein flexibles Vertauschen von Rückbau- und Stoffstromvorgängen bei ihrer Einplanung (vgl. Abschnitt 6.3.3). Dies ist für die Zielfunktion  $z(S) := S_{I+1}$  bei gemeinsamen Ressourcennutzungen im Rückbau- und Stoffstrombereich (d. h.  $RFP = 0,5$ ) nachteilig, da es besser sein kann, zunächst alle Rückbauvorgänge einzuplanen. Mit der Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  ist dieser Nachteil jedoch nicht mehr relevant, da auch die Stoffstromvorgänge mitberücksichtigt werden.

Der CP Optimizer konnte mit der Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  mehr aggregierte Instanzen optimal lösen. Mit der Lösungsstrategie (agg|CP|WS|step) konnte die Optimalität für 91 der 96 aggregierten Instanzen bewiesen werden (im Vergleich zu 38

mit der ursprünglichen Zielfunktion). Besonders interessant ist dabei, dass nun auch die Optimalität bewiesen werden konnte, falls  $z > LB_{PSPLIB}$ . Dies war mit der ursprünglichen Zielfunktion nicht möglich. Daraus lässt sich schließen, dass dem CP Optimizer der Optimalitätsnachweis mit der in der Literatur untypischen originalen Zielfunktion  $z(S) := S_{I+1}$  schwerer fällt, als mit der typischen Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$ .

Für 25 Testinstanzen konnte der CP Optimizer sowohl für die granulare als auch die dazugehörige aggregierte Instanz die Optimalität beweisen. Mit dieser Information kann hierfür nun eine fundierte Aussage über die Auswirkungen des Verlustes an Realitätstreue durch die Aggregation getroffen werden: Im Durchschnitt wurden für die 25 granularen Instanzen um 1,33 % bessere optimale Zielfunktionswerte berechnet als für die 25 dazugehörigen aggregierten Instanzen. Damit ist belegt, dass der Verlust an Realitätstreue einen messbaren Einfluss auf die Zielfunktionswerte hat. Allerdings ist das Ausmaß dieses Einflusses für die hier betrachteten Testinstanzen überschaubar.

Zusammenfassend lässt sich festhalten, dass die vorgestellten Lösungsverfahren (abgesehen von der linearen Programmierung, für die keine Aussage getroffen werden kann) auch für die Zielfunktion  $z(S) := \max\{S_{I+1}, S_{(J+1)^o}^o\}$  geeignet sind. Für sämtliche Parameterwerte schneidet entweder das neu entwickelte Lösungsverfahren SGS-gra oder der CP Optimizer am besten ab. Eine klare Überlegenheit eines dieser beiden Lösungsverfahren ist nicht zu identifizieren. Für einige granulare Instanzen fällt es dem CP Optimizer ohne Warmstart schwer, innerhalb von zehn Minuten zulässige Lösungen zu finden. Das Lösungsverfahren SGS-agg-i liefert ebenfalls vergleichsweise gute Ergebnisse. Somit ist für aggregierte Instanzen der Integrationsansatz wie schon bei der ursprünglichen Zielfunktion  $z(S) := S_{I+1}$  dem Dekompositionsansatz gegenüber vorzuziehen.





# 7 Zusammenfassung und Ausblick

## 7.1 Zusammenfassung

Die Europäische Union strebt als eines ihrer Umweltziele den Übergang zu einer Kreislaufwirtschaft an. Zur Erreichung dieses Umweltziels sind insbesondere im Bausektor, welcher im Jahr 2020 ca. 55 % des Abfallaufkommens in Deutschland ausmachte (Statistisches Bundesamt 2020), Anstrengungen zur Verringerung, zur Aufbereitung und zum Recycling von Abfällen erforderlich. Statt eines konventionellen Abbruchs wird daher vermehrt dazu übergegangen, Gebäude, Anlagen oder Infrastrukturen selektiv rückzubauen. Trotz der damit einhergehenden höheren Sortiertiefe von Materialien ist üblicherweise eine Aufbereitung von Bauabfällen erforderlich, um sie dem Recycling zuführen zu können (Müller 2018, S. 19 und 61). Die Aufbereitung sowie auch der Transport und die Lagerung von Materialien unterliegen in aller Regel Kapazitätsbeschränkungen, da beispielsweise die Anzahl der verfügbaren Anlagen oder der Platz auf der Baustelle begrenzt sind. Zur Erstellung eines realistischen Ablaufplans für ein Rückbauprojekt sollten daher diese stoffstrombezogenen Restriktionen in die Planung aufgenommen werden. Dadurch kann der Rückbau im Voraus so gesteuert werden, dass freigesetzte Stoffströme ohne Kapazitätsüberlastungen aufbereitet, transportiert und gelagert werden können.

Stoffstrombezogene Restriktionen sind im Rückbau kerntechnischer Anlagen von besonderer Bedeutung. Dies ist der Kombination aus der materialintensiven Bauweise und der hohen strahlenschutzrechtlichen Sicherheitsanforderungen geschuldet. So bedürfen nach § 31 der Strahlenschutzverordnung sämtliche Stoffe und Gegenstände, die aus dem Kontrollbereich eines Kernkraftwerks stammen und dem Wertstoffkreislauf zugeführt werden sollen, einer Freigabe. Zur Erteilung einer Freigabe müssen geeignete Freimessungen durchgeführt werden. Um den Anteil freigabefähiger Reststoffe zu erhöhen und damit einen Beitrag zum Übergang zu einer Kreislaufwirtschaft zu leisten, werden Stoffe aus dem Kontrollbereich mittels verschiedener Dekontaminationsverfahren aufbereitet. Zwischen der Demontage, der Anwendung von Dekontaminationsverfahren und der Durchführung von Freimessungen werden die Stoffe auf vorbestimmten Lagerflächen gepuffert. Die Dekontaminations-, Freimess- und Lagerkapazitäten stellen aufgrund mangelnder Verfügbarkeit, aus Kostengründen oder wegen Platzmangels einen beträchtlichen Engpass dar, den es in der Ablaufplanung zu berücksichtigen gilt.

Basierend auf dieser Ausgangslage hat sich diese Arbeit zum Ziel gesetzt, ein stoffstrombasiertes Optimierungswerkzeug zur Ablaufplanung von Rückbauprojekten zu entwickeln und zu testen. Aufgrund der Komplexität und des Umfangs realer Rückbauprojekte ist es

evident, dass ein solches Optimierungswerkzeug auf quantitativen Modellen basieren und rechnergestützt umgesetzt werden soll.

Die Literatur stellt bereits eine Vielzahl quantitativer Modelle und Methoden für die optimierende (Projekt-)Ablaufplanung zur Verfügung. Dem englischsprachigen Fachbegriff entsprechend wird von Schedulingmodellen und -problemen gesprochen. Mit Hilfe eines (mathematischen) Schedulingproblems lassen sich Fragestellungen zur Termin- und Kapazitätsplanung beantworten, wie beispielsweise die Frage nach Startzeiten für die (Projekt-)Vorgänge, sodass die Projektdauer minimiert wird und Vorrangbeziehungen zwischen den Vorgängen sowie vorhandene Ressourcenkapazitäten eingehalten werden. Von besonderer Bedeutung für diese Arbeit ist das aus der Literatur bekannte Schedulingproblem unter erneuerbaren und kumulativen Ressourcenrestriktionen (RCPSP/c). Da das darin enthaltene Konzept der kumulativen Ressourcen die Modellierung beschränkter Lagerkapazitäten ermöglicht, bildet es eine wichtige Grundlage für die weitere Ausarbeitung. Im (RCPSP/c) fehlt jedoch die Möglichkeit, die speziellen Charakteristika bei der Planung eines Rückbauprojekts abzubilden.

Das stoffstrombasierte Optimierungswerkzeug soll Projektverantwortliche primär dabei unterstützen, zu beurteilen, welche Auswirkungen die stoffstrombezogenen Restriktionen auf die Dauer und den Ablaufplan (dem englischsprachigen Begriff entsprechend in dieser Arbeit überwiegend als *Schedule* bezeichnet) eines Rückbauprojekts haben. Sekundär soll es auch eine Antwort auf die Frage nach dem exakten zeitlichen Ablauf der Stoffverarbeitung liefern. Dieser Abstufung entsprechend wird von einer primären Planungsebene, die sich mit der Rückbauplanung befasst, und einer sekundären Planungsebene, die sich mit der Stoffstromplanung befasst, gesprochen. Zur Formalisierung dieser Problemstellung wird zunächst ein Stoffstrommodell entwickelt, welches zur Beschreibung der bei einem Rückbauprojekt freigesetzten Stoffströme dient. Basierend darauf wird das neuartige Schedulingproblem unter Ressourcen- und Stoffstromrestriktionen (RMCPSP) entwickelt. Darin ist mathematisch die Frage formuliert, welche Startzeiten für die Rückbauvorgänge gewählt werden sollen, sodass die Projektdauer minimiert und Vorrangbeziehungen zwischen den Rückbauvorgängen sowie vorhandene erneuerbare Ressourcen-, Verarbeitungs- und Lagerkapazitäten eingehalten werden. Das (RMCPSP) ist mit seinem Fokus auf die Rückbauvorgänge auf der primären Planungsebene angesiedelt. Die Verbindungen zur sekundären Planungsebene der Stoffstromplanung sind mittels abstrakter Funktionen ausgedrückt. Damit gelingt es, die aufgrund des Repetitionstyps eher dem produktionswirtschaftlichen Kontext zuordenbare Stoffverarbeitung mit dem Rückbau, welcher klassische Projektmerkmale aufweist, konzeptionell zu verbinden.

Ein Vergleich mit existierenden Ansätzen in der Literatur zeigt, dass das entwickelte Stoffstrommodell verschiedene Konzepte von graphentheoretischen Modellen aus dem Produktionsmanagement vereint. Es unterscheidet sich jedoch insofern von existierenden Modellen, als die Stoffströme in dieser Arbeit als inhomogen gelten, da sie abhängig von ihrer Nutzungsvergangenheit individuelle Charakteristika aufweisen können und daher auch individuell modelliert werden müssen. Dies wird über die sogenannte stoffstrominduzierte Zeitstruktur umgesetzt, welche die Stoffeinheiten stets in Verbindung mit demjenigen Rückbauvorgang hält, aus welchem sie stammen. Als ein wesentliches

Alleinstellungsmerkmal des (RMCPSP) gegenüber existierenden Schedulingproblemen gilt außerdem die Stellung der Stoffströme als sekundäres Untersuchungsobjekt. Das heißt, im (RMCPSP) wird das primär geplante Rückbauprojekt (und damit auch die Zielsetzung) nur beeinflusst, falls es zu einer Überlastung der Verarbeitungs- und Lagerkapazitäten bei der Stoffverarbeitung kommt. In der entsprechenden Literatur hingegen ist die Verarbeitung und Lagerung von Stoffen stets primäres Untersuchungsobjekt, da die behandelten Problemstellungen produktionswirtschaftlicher Art sind.

Aufgrund der sekundären Stellung des Stoffstrombereichs ist im (RMCPSP) keine zeitliche Dimension für die Stoffverarbeitung spezifiziert. Die stoffstrombezogenen Restriktionen sind nur auf abstrakte Weise formuliert. Zur Lösung des (RMCPSP), d. h. zur Berechnung eines Ablaufplans für das Rückbauprojekt, ist eine Konkretisierung des Modells erforderlich. Es ist hierbei naheliegend, die Stoffverarbeitung analog zum Rückbau vorgangsbasiert zu modellieren. Dazu wird der Begriff der Stoffstromvorgänge in Ergänzung zu den Rückbauvorgängen etabliert. Die vorgangsbasierte Modellierung ermöglicht es, auf die aus der Literatur bekannten Konzepte, wie beispielsweise kumulative Ressourcen, zurückzugreifen. Die Abstufung zwischen primärer und sekundärer Ebene wird aufgelöst, weshalb das konkretisierte Problem als Schedulingproblem zur integrierten Planung von Rückbauprojekt und Stoffströmen (RCPSP/M) bezeichnet wird. Darin ist die Frage formuliert, welche Startzeiten für die Rückbau- und Stoffstromvorgänge gewählt werden sollen, sodass die Projektdauer minimiert und Vorrangbeziehungen zwischen den Rückbauvorgängen, die stoffstrominduzierte Zeitstruktur sowie vorhandene erneuerbare und kumulative Ressourcenkapazitäten eingehalten werden.

Da es für die Erstellung von Stoffstromvorgängen keine idealtypische Herangehensweise gibt, werden verschiedene Ausgestaltungsmöglichkeiten betrachtet und algorithmisch als sogenannte Transformationsschemas ausformuliert. Mit dem granularen Transformationsschema wird für jeden Verarbeitungsschritt, der auf jede Stoffeinheit angewendet wird, ein separater, granularer Stoffstromvorgang erzeugt. Mit dem aggregierten Transformationsschema werden mehrere granulare Stoffstromvorgänge zu aggregierten Stoffstromvorgängen verschmolzen. Die Verschmelzung verringert die Größe einer Probleminstanz, allerdings ist auch ein Verlust an Flexibilität bei der Planung der Stoffverarbeitung hinzunehmen. Transformationsschemas sind lediglich als ein Hilfsmittel bei der Modellierung eines realen Problems als Instanz des (RCPSP/M) zu verstehen. Es sind weitere Transformationsschemas denkbar und es ist Projektverantwortlichen auch freigestellt, ein Rückbauprojekt direkt als Instanz des (RCPSP/M) zu modellieren. Dennoch ist es zum Verständnis des Gesamtkonzepts in dieser Arbeit zwingend erforderlich, zunächst das abstrakte (RMCPSP) zu formulieren und danach zum konkretisierten (RCPSP/M) überzuleiten. Denn nur so lässt sich der Zweck der Transformationsschemas, die Stoffverarbeitung in die Ablaufplanung des Rückbauprojekts zu integrieren, darstellen.

Das (RCPSP/M) ist der Ansatzpunkt für Lösungsverfahren, die rechnergestützt Startzeiten für die Rückbau- und Stoffstromvorgänge einer Probleminstanz berechnen, sodass die Projektdauer minimiert wird. Aufgrund der kumulativen Ressourcen handelt es sich um ein NP-schweres Optimierungsproblem. Darüber hinaus ist bereits die Suche nach einer zulässigen Lösung ein NP-vollständiges Problem. Ein Vergleich mit existierenden Ansät-

zen in der Literatur zeigt, dass exakte Lösungsverfahren existieren, die mit kumulativen Ressourcen umgehen können und somit auch zur Lösung des (RCPSP/M) geeignet sind. Allerdings ist zu erwarten, dass die exakten Lösungsverfahren aufgrund der NP-Schwere des Problems bei großen Probleminstanzen an die Grenzen ihrer Leistungsfähigkeit stoßen. Weitere existierende Lösungsverfahren scheidet aus, da sie nicht hinreichend detailliert beschrieben oder nur auf spezialisierte Probleme anwendbar sind. Damit begründet sich die Notwendigkeit zur Entwicklung neuer Lösungsverfahren für das (RCPSP/M).

Um auch für große Probleminstanzen effizient Lösungen berechnen zu können, wird ein heuristischer Ansatz gewählt. Grundlage bilden die in der Schedulingliteratur ausgiebig erforschten prioritätsregelbasierten Verfahren, welche aus einer Prioritätsregel und einem Generierungsschema bestehen. Bei einem Generierungsschema handelt es sich um einen Algorithmus, der durch sukzessives Einplanen von Vorgängen (d. h. Bestimmung von Startzeiten für Vorgänge) einen zulässigen Ablaufplan bestimmt. Für erneuerbare Ressourcen existieren bereits Generierungsschemas. Ein wesentlicher Beitrag dieser Arbeit ist die Entwicklung von Generierungsschemas für kumulative Ressourcen, die in einer vergleichbaren Detailtiefe bisher nicht vorliegen. Im Einzelnen werden vier neuartige Generierungsschemas entwickelt:

- Ein Generierungsschema nach einem Dekompositionsansatz, welches speziell für Instanzen des (RCPSP/M), die mit dem granularen Transformationsschema erzeugt wurden, geeignet ist.
- Ein Generierungsschema nach einem Dekompositionsansatz, welches speziell für Instanzen des (RCPSP/M), die mit dem aggregierten Transformationsschema erzeugt wurden, geeignet ist.
- Ein Generierungsschema nach einem Integrationsansatz, welches speziell für Instanzen des (RCPSP/M), die mit dem aggregierten Transformationsschema erzeugt wurden, geeignet ist.
- Ein Generierungsschema nach einem Integrationsansatz, welches für beliebige Instanzen des (RCPSP/M) geeignet ist.

Beim Dekompositionsansatz werden Rückbau- und Stoffstromvorgänge getrennt voneinander auf zwei algorithmischen Ebenen eingeplant. Beim Integrationsansatz werden Rückbau- und Stoffstromvorgänge gleichrangig auf derselben algorithmischen Ebene eingeplant. Beide Ansätze bieten jeweils Vor- und Nachteile, beispielsweise hinsichtlich der Effizienz zur Erzeugung zulässiger Ablaufpläne oder der Größe des abgesuchten Lösungsraums. Neben den neu entwickelten Generierungsschemas werden auch die Constraint- und die lineare Programmierung als exakte Lösungsverfahren für das (RCPSP/M) vorgestellt.

Zur Beurteilung der Leistungsfähigkeit der Lösungsverfahren und zur Anwendungs-demonstration werden Rechenexperimente durchgeführt. Um eine möglichst große Bandbreite an realen Problemstellungen abzudecken und aufgrund mangelnder Praxisdaten werden systematisch Testinstanzen erzeugt. Die neu entwickelten Generierungsschemas werden wahlweise in eine Multi-Start- oder eine Simulated-Annealing-Metaheuristik eingebettet,

welche die Prioritätsregeln zur Durchsuchung des Lösungsraums während der Rechenzeit variieren. Als Solver für die Constraintprogrammierung wird IBM ILOG CP Optimizer und als Solver für die lineare Programmierung IBM ILOG CPLEX gewählt. Unter Austausch verschiedener variabler Bestandteile, wie der Metaheuristik, des Lösungsverfahrens oder des Transformationsschemas wird eine Serie umfangreicher Rechendurchläufe ausgeführt und tabellarisch aufbereitet.

Die Auswertung zeigt, dass die neu entwickelten Lösungsverfahren in der Lage sind, große und damit realistische Probleminstanzen innerhalb kurzer Rechenzeiten zulässig und mit hoher Qualität zu lösen. Sie haben jeweils individuelle Stärken und Schwächen, die abhängig von den Charakteristika der betrachteten Probleminstanz (wie beispielsweise der Anzahl an Rückbauvorgängen oder der Anzahl an freigesetzten Stoffeinheiten) zur Geltung kommen. Im Vergleich zum CP Optimizer wird außerdem deutlich, dass die neu entwickelten Lösungsverfahren hinsichtlich des abbildbaren Modellumfangs überlegen sind. So können nur sie mit Stofffreisetzungen und kumulativen Ressourcennutzungen umgehen, die sich gleichmäßig über die Dauer der Vorgänge erstrecken. Bereinigt von modelltheoretischen Unterschieden zeigt der CP Optimizer ebenfalls eine sehr hohe Leistungsfähigkeit. Er ist den neu entwickelten Lösungsverfahren überlegen, falls die erneuerbaren im Vergleich zu den kumulativen Ressourcenrestriktionen stärker binden (d. h. mehr Engpässe beim Personal oder bei Maschinen vorliegen als bei der Lagerfläche). Die lineare Programmierung stellt sich als ungeeignetes Lösungsverfahren heraus, da die hohe Anzahl an binären Entscheidungsvariablen zu große Leistungseinbußen verursacht.

Die vorliegende Arbeit liefert sowohl hinsichtlich des Modells als auch der entwickelten Lösungsverfahren einen Beitrag zum Stand der Forschung, wie die jeweiligen Vergleiche mit Ansätzen in der Literatur zeigen. Ausgehend von den Grundlagen über die Entwicklung des Stoffstrommodells, die Formulierung des (RMCPSP), die Transformationsschemas, die Formulierung des (RCPSP/M), die Lösungsverfahren bis hin zu den Rechenexperimenten mit ihrer Auswertung und den darin enthaltenen Empfehlungen wird ein Gesamtkonzept entwickelt, welches das *stoffstrombasierte Optimierungswerkzeug* und damit Antwort auf die eingangs formulierte Forschungsfrage darstellt.

## 7.2 Kritische Würdigung und Ausblick

In den Modelleinschränkungen (vgl. Abschnitt 3.3) wurde festgelegt, dass ausschließlich die Zielsetzung zur Minimierung der Projektdauer betrachtet wird. Dies wurde damit motiviert, dass im Rückbau kerntechnischer Anlagen nach Thierfeldt und Schartmann (2009, S. 34) sowie Laufs (2018, S. 336) eine möglichst schnelle Projektdurchführung angestrebt wird. Gleichzeitig ermöglicht der Fokus auf die Minimierung der Projektdauer auch eine Anknüpfung an die existierenden Schedulingprobleme (RCPSP) und (RCPSP/c). Die Minimierung der Projektdauer ist laut Węglarz u. a. (2011, S. 182) die in der Schedulingliteratur am häufigsten betrachtete Zielsetzung. Es lässt sich vermuten, dass dies nicht nur der Motivation aus der Praxis geschuldet ist, sondern auch der Tatsache, dass die Minimierung der Projektdauer als reguläre Zielfunktion zu den intuitivsten Zielsetzungen

gehört. Außerdem werden im Gegensatz zu anderen Zielsetzungen, wie beispielsweise finanziellen Zielsetzungen, weniger Daten benötigt. Es ist nicht ausgeschlossen, dass für die Rückbauplanung auch andere Zielsetzungen relevant sind. So verfolgt Hübner (2019) mit seinem Planungswerkzeug für den Rückbau kerntechnischer Anlagen die Zielsetzung der Minimierung der Ressourcenbeschaffungs- und Restbetriebskosten. Bartels (2009) wiederum minimiert den Kapitalwert. Für solche nichtregulären Zielsetzungen wären die in dieser Arbeit entwickelten Lösungsverfahren nicht mehr geeignet, sondern müssten angepasst werden. Zur Minimierung der Ressourcenbeschaffungskosten müssten die Generierungsschemas beispielsweise so angepasst werden, dass Vorgänge nicht nur frühestmöglich eingeplant werden, sondern auch zu späteren Zeitpunkten, zu denen die Ressourcenprofile geringer sein können. Damit könnte die Fragestellung beantwortet werden, wie viele Lager- und Verarbeitungskapazitäten mindestens benötigt werden, um ein Rückbauprojekt zu einem gegebenen Endtermin fertigzustellen.

Der in Abschnitt 2.2.5 für die erneuerbaren Ressourcen vorgestellte Mehr-Modus-Fall könnte auf die kumulativen Ressourcen übertragen werden. Hierdurch würde sich die Komplexität des Schedulingproblems weiter erhöhen, da bereits das Entscheidungsproblem zur Existenz einer zulässigen Moduzuweisung bei Vorliegen von nichterneuerbaren Ressourcen NP-vollständig ist. Für kumulative Ressourcen wäre das Problem mindestens ebenso komplex, da sie eine Generalisierung der nichterneuerbaren Ressourcen sind. Der Mehr-Modus-Fall in Verbindung mit kumulativen Ressourcen würde den Modellumfang dahingehend erweitern, dass auch Entscheidungen über alternative Lagerflächen in die Problemstellung eingebunden werden könnten. Für das in dieser Arbeit betrachtete Problem wurde hingegen bei den Modellabgrenzungen (vgl. Abschnitt 3.3) definiert, dass keine Entscheidungen zwischen verschiedenen Lagerflächen getroffen werden.

Während für aggregierte und beliebige Instanzen des (RCPSP/M) ein Generierungsschema nach einem Integrationsansatz entwickelt wurde, wurde für granulare Instanzen ausschließlich der Dekompositionsansatz gewählt. In Abschnitt 5.7 ist zwar begründet, weshalb auf einen Integrationsansatz verzichtet wurde. Aufgrund der Erkenntnisse aus den Rechenexperimenten, in denen der Integrationsansatz für aggregierte Instanzen besser als der Dekompositionsansatz abgeschnitten hat, wäre jedoch auch für granulare Instanzen ein entsprechender Ansatz in Betracht zu ziehen. Dies könnte dazu beitragen, die für bestimmte Instanzcharakteristika, wie beispielsweise einer geringen Anzahl an Rückbauvorgängen, identifizierten Schwachstellen des Lösungsverfahrens für granulare Instanzen auszugleichen.

Weitere Verbesserungen bei der Leistungsfähigkeit der neu entwickelten Lösungsverfahren wären möglicherweise mit hybriden Ansätzen zu erzielen. In einer ersten Berechnungsstufe könnte für die aggregierte Instanz eine gute zulässige Lösung gesucht werden, die dann als Startlösung für die zugehörige granulare Instanz bei der weiteren Durchsuchung des Lösungsraums in einer zweiten Berechnungsstufe dient. Dies ist bei Verwendung der vorgestellten Transformationsschemas möglich, da jede zulässige Lösung für eine aggregierte Instanz in eine zulässige Lösung für die zugrundeliegende granulare Instanz rücktransformiert werden kann.

Die aufgrund der verfügbaren Schnittstellen vorliegende Einschränkung, dass der CP Optimizer nur die Stufenvariante einer Instanz lösen kann, könnte behelfsmäßig umgangen werden: Ein Vorgang könnte durch verschiedene Teilvorgänge ersetzt werden, die über Zeitbeziehungen mit gleich großen zeitlichen Mindest- und Höchstabständen zu einer nichtunterbrechbaren Kette verknüpft werden. Jedem Teilvorgang kann dann ein seinem Anteil an der Dauer des Ursprungsvorgangs entsprechender Teil der kumulativen Ressourcennutzung zugeordnet werden. Damit wird eine gleichmäßige kumulative Ressourcennutzung über die Vorgangsausführung angenähert. Um zu identifizieren, mit welcher Granularität die Teilvorgänge erzeugt werden sollten ohne den Rechenaufwand übermäßig zu erhöhen, müsste eine umfangreiche Serie an Rechenexperimenten durchgeführt werden.

Zur Lösung des (RCPSP/M) mittels linearer Programmierung könnten neben der zeitindizierten Formulierung weitere Formulierungsansätze auf das Problem angepasst und getestet werden. Ein Herunterskalieren des Zeithorizonts zur Verringerung der Anzahl an binären Entscheidungsvariablen könnte ebenfalls in Betracht gezogen werden. Hierbei wäre eine experimentelle Untersuchung erforderlich, inwiefern die durch die Herunterskalierung entstehenden Rundungsungenauigkeiten bei den Vorgangsdauern sich auf die Lösungsqualität auswirken würden.

Die Leistungsfähigkeit der Lösungsverfahren wurde anhand fiktiver Testinstanzen beurteilt. Obwohl die Testinstanzen auf verschiedenen Instanzparametern basieren, deren Wertebereiche variiert wurden, ist nicht belegt, dass sie realen Probleminstanzen entsprechen. Insbesondere die durch die PSPLIB vorgegebene Beschränkung auf maximal 120 Rückbauvorgänge schränkt die Praxisnähe der Testinstanzen ein. Der ideale Test des in dieser Arbeit entwickelten Optimierungswerkzeugs bestünde aus einer Berechnung eines Ablaufplans für ein reales, bereits abgeschlossenes Rückbauprojekt und einem anschließenden Vergleich mit dem ursprünglichen Ablaufplan. Könnte dabei eine Verringerung der Projektdauer im optimierten Ablaufplan festgestellt werden, wäre der Nutzen nachgewiesen. Da der Großteil der kerntechnische Rückbauprojekte jedoch erst in diesen und den kommenden Jahren anläuft, sind für diesen Anwendungsfall noch wenig historische Daten verfügbar. Auch Daten aus laufenden Planungsverfahren liegen oft noch nicht in der benötigten Genauigkeit vor. Daneben sind aus Geheimhaltungsgründen der Zugang und die Möglichkeiten zur Veröffentlichung von Daten aus solchen Projekten stark eingeschränkt. Daher wurde auf fiktive Testinstanzen zurückgegriffen, wie dies beispielsweise auch in Neumann, Schwindt und Trautmann (2005, S. 505) und Bartels (2009, S. 104) erfolgt. Hübner (2019, S. 443f) verwendet ebenfalls keine realen Daten zur Validierung seines Planungswerkzeugs für den Rückbau kerntechnischer Anlagen, kann jedoch auf ein Referenzprojekt zurückgreifen, welches von Experten hinsichtlich seiner Realitätstreue validiert wurde.

In dieser Arbeit wurde ein deterministisches Schedulingmodell entwickelt. Umfangreiche und zeitlich langandauernde Rückbauprojekte unterliegen allerdings großen Unsicherheitsfaktoren. Insbesondere kerntechnische Rückbauprojekte sind aufgrund ihrer komplexen Natur und der fehlenden Erfahrungswerte anfällig für Planabweichungen. Da der radiologische Ausgangszustand vieler Stoffe nicht im Voraus bekannt ist, können unvorhergesehene

Mehrarbeiten erforderlich werden. Für die Ablaufplanung könnten zukünftig Konzepte entwickelt werden, solche Unsicherheiten zu berücksichtigen. Hierfür existieren in der Literatur zahlreiche Vorschläge. Beispielsweise berücksichtigt Hübner (2019) Unsicherheiten in seinem Modell zur Planung eines kerntechnischen Rückbauprojekts, indem die Vorgangsdauern und die Zeitstruktur stochastisch beschrieben werden. Anschließend werden verschiedene Szenarien simuliert, wobei jedes Szenario eine mögliche Ausprägung der Dauern und der Zeitstruktur darstellt. Für jedes Szenario wird ein optimierter Ablaufplan berechnet. Die optimierten Ablaufpläne werden hinsichtlich ihrer Ausführbarkeit in anderen Szenarien miteinander verglichen. Diese als Robustheit bezeichnete Eigenschaft stellt für Projektverantwortliche neben dem Zielfunktionswert eine weitere Kennzahl dar, nach welcher sie den am besten geeigneten Ablaufplan auswählen können. Das in dieser Arbeit entwickelte Optimierungswerkzeug könnte ebenfalls in ein solches Konzept eingebettet werden. Es wären ein stochastisches Stoffstrommodell sowie für die Stoffströme geeignete Robustheitsmaße zu entwickeln. Eine Herausforderung in der kerntechnischen Anwendung stellt zum jetzigen Stand des Projektfortschritts die Quantifizierung der Unsicherheiten dar, da Erfahrungswerte fehlen.

Mit fortschreitender Planung ist insbesondere in der kerntechnischen Branche damit zu rechnen, dass sich in Zukunft verschiedene Richtungen zur Verfeinerung des entwickelten Optimierungswerkzeugs ergeben. Voraussetzung dafür ist eine geeignete Dokumentation und die Bereitschaft der beteiligten Akteure, Erfahrungen und Daten bereitzustellen. Hierbei sollten Anstrengungen unternommen werden, mögliche Hemmnisse im Zusammenhang mit Geheimhaltungsabsichten und -pflichten abzubauen. Auch Befürchtungen, dass durch die Anwendung eines Optimierungswerkzeugs nachträglich Schwachstellen in der Planung identifiziert und dies den Beteiligten zum Nachteil ausgelegt werden könnte, sollte entgegengetreten werden. Der Rückbau kerntechnischer Anlagen sollte als eine gesamtgesellschaftliche Aufgabe wahrgenommen werden, die in Deutschland und weltweit noch über Jahrzehnte bestehen wird und die es zu optimieren gilt.



# Literatur

- Agarwal, Anurag, Selcuk Colak und Selcuk Erenguc (2015). „Metaheuristic methods“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 57–74. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- ARGE Stilllegung Biblis (2020). *Kraftwerk Biblis Block A – Gutachten zum 2. Antrag auf Abbau nach § 7 Abs. 3 Atg (A017/17): Erstellt im Auftrag des Hessischen Ministeriums für Umwelt, Klimaschutz, Landwirtschaft und Verbraucherschutz von der ARGE Stilllegung Biblis*. URL: [https://umwelt.hessen.de/sites/umwelt.hessen.de/files/2021-09/gutachten\\_block\\_a\\_nicht\\_barrrierefrei.pdf](https://umwelt.hessen.de/sites/umwelt.hessen.de/files/2021-09/gutachten_block_a_nicht_barrrierefrei.pdf) (besucht am 07.02.2023).
- Arnold, Dieter und Kai Furmans (2009). *Materialfluss in Logistiksystemen*. 6., erweiterte Auflage. VDI-Buch. Berlin, Heidelberg: Springer. ISBN: 978-3-642-01405-5. DOI: 10.1007/978-3-642-01405-5.
- Artigues, Christian u. a. (2015). „Mixed-integer linear programming formulations“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 17–41. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Ballestín, Francisco und Rosa Blanco (2015). „Theoretical and practical fundamentals“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 411–427. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Baptiste, Philippe, Claude Le Pape und Wim Nuijten (1999). „Satisfiability tests and time-bound adjustments for cumulative scheduling problems“. In: *Annals of Operations Research* 92.0, S. 305–333. DOI: 10.1023/A:1018995000688.
- Bartels, Jan-Hendrik (2009). „Anwendung von Methoden der ressourcenbeschränkten Projektplanung mit multiplen Ausführungsmodi in der betriebswirtschaftlichen Praxis: Rückbauplanung für Kernkraftwerke und Versuchsträgerplanung in Automobilentwicklungsprojekten“. Diss. Wiesbaden: Technische Universität Clausthal. ISBN: 978-3-8349-1696-9. DOI: 10.1007/978-3-8349-8228-5.
- Bartels, Jan-Hendrik, Thorsten Gather und Jürgen Zimmermann (2011). „Dismantling of nuclear power plants at optimal NPV“. In: *Annals of Operations Research* 186.1, S. 407–427. DOI: 10.1007/s10479-010-0788-9.
- Bartusch, M., Rolf H. Möhring und Franz J. Radermacher (1988). „Scheduling project networks with resource constraints and time windows“. In: *Annals of Operations Research* 16.1, S. 199–240. DOI: 10.1007/BF02283745.

- Baumann, Philipp und Norbert Trautmann (2014). „A hybrid method for large-scale short-term scheduling of make-and-pack production processes“. In: *European Journal of Operational Research* 236.2, S. 718–735. DOI: 10.1016/j.ejor.2013.12.040.
- Belaid, Rabah, Vincent T'kindt und Carl Esswein (2012). „Scheduling batches in flowshop with limited buffers in the shampoo industry“. In: *European Journal of Operational Research* 223.2, S. 560–572. DOI: 10.1016/j.ejor.2012.06.035.
- Blazewicz, Jacek, Jan Karel Lenstra und Alexander H. G. Rinnooy Kan (1983). „Scheduling subject to resource constraints: Classification and complexity“. In: *Discrete Applied Mathematics* 5.1, S. 11–24. DOI: 10.1016/0166-218X(83)90012-4.
- Bloech, Jürgen u. a. (2014). *Einführung in die Produktion*. 7., korrigierte und aktualisierte Auflage 2014. Springer-Lehrbuch. Berlin: Springer Gabler. ISBN: 978-3-642-31892-4. DOI: 10.1007/978-3-642-31893-1.
- Blömer, Ferdinand und Hans-Otto Günther (1998). „Scheduling of a multi-product batch process in the chemical industry“. In: *Computers in Industry* 36.3, S. 245–259. DOI: 10.1016/S0166-3615(98)00075-X.
- (2000). „LP-based heuristics for scheduling chemical batch processes“. In: *International Journal of Production Research* 38.5, S. 1029–1051. DOI: 10.1080/002075400189004.
- BMUB (Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit) (2021). *Leitfaden zur Stilllegung, zum sicheren Einschluss und zum Abbau von Anlagen oder Anlagenteilen nach § 7 des Atomgesetzes*. Bonn. URL: [https://www.base.bund.de/SharedDocs/Downloads/BASE/DE/rsh/3-bmub/3\\_73.pdf?\\_\\_blob=publicationFile&v=1](https://www.base.bund.de/SharedDocs/Downloads/BASE/DE/rsh/3-bmub/3_73.pdf?__blob=publicationFile&v=1) (besucht am 07.02.2023).
- Doctor, Fayed F. (1993). „Heuristics for scheduling projects with resource restrictions and several resource-duration modes“. In: *International Journal of Production Research* 31.11, S. 2547–2558. DOI: 10.1080/00207549308956882.
- Bofill, Miquel u. a. (2022). „The sample analysis machine scheduling problem: Definition and comparison of exact solving approaches“. In: *Computers & Operations Research* 142, S. 105730. DOI: 10.1016/j.cor.2022.105730.
- Bouleimen, Kamel und H. Lecocq (2003). „A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version“. In: *European Journal of Operational Research* 149.2, S. 268–281. DOI: 10.1016/S0377-2217(02)00761-0.
- Boysen, Nils, Stefan Bock und Malte Fliedner (2013). „Scheduling of inventory releasing jobs to satisfy time-varying demand: An analysis of complexity“. In: *Journal of Scheduling* 16.2, S. 185–198. DOI: 10.1007/s10951-012-0266-0.
- Briskorn, Dirk (2020). *Operations Research*. Berlin, Heidelberg: Springer Gabler. ISBN: 978-3-662-60782-4. DOI: 10.1007/978-3-662-60783-1.
- Briskorn, Dirk, Byung-Cheon Choi u. a. (2010). „Complexity of single machine scheduling subject to nonnegative inventory constraints“. In: *European Journal of Operational Research* 207.2, S. 605–619. DOI: 10.1016/j.ejor.2010.05.036.
- Briskorn, Dirk, Florian Jaehn und Erwin Pesch (2013). „Exact algorithms for inventory constrained scheduling on a single machine“. In: *Journal of Scheduling* 16.1, S. 105–115. DOI: 10.1007/s10951-011-0261-x.

- 
- Briskorn, Dirk und Joseph Y.-T. Leung (2013). „Minimizing maximum lateness of jobs in inventory constrained scheduling“. In: *Journal of the Operational Research Society* 64.12, S. 1851–1864. DOI: 10.1057/jors.2012.155.
- Briskorn, Dirk und Erwin Pesch (2013). „Variable very large neighbourhood algorithms for truck sequencing at transshipment terminals“. In: *International Journal of Production Research* 51.23-24, S. 7140–7155. DOI: 10.1080/00207543.2013.849825.
- Briskorn, Dirk und Philipp Zeise (2019). „A cyclic production scheme for the synchronized and integrated two-level lot-sizing and scheduling problem with no-wait restrictions and stochastic demand“. In: *OR Spectrum* 41.4, S. 895–942. DOI: 10.1007/s00291-019-00555-y.
- Brucker, Peter u. a. (1999). „Resource-constrained project scheduling: Notation, classification, models, and methods“. In: *European Journal of Operational Research* 112.1, S. 3–41. DOI: 10.1016/S0377-2217(98)00204-5.
- Brusa, L. u. a. (2002). „The decommissioning of the Trino nuclear power plant“. In: URL: <https://www.osti.gov/biblio/829556> (besucht am 07.02.2023).
- Burkard, Rainer E. u. a. (1998). „A process scheduling problem arising from chemical production planning“. In: *Optimization Methods and Software* 10.2, S. 175–196. DOI: 10.1080/10556789808805710.
- Carlier, Jacques und Aziz Moukrim (2015). „Storage resources“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 177–189. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Carlier, Jacques, Aziz Moukrim und Huang Xu (2009). „The project scheduling problem with production and consumption of resources: A list-scheduling based algorithm“. In: *Discrete Applied Mathematics* 157.17, S. 3631–3642. DOI: 10.1016/j.dam.2009.02.012.
- Coelho, José und Mario Vanhoucke (2011). „Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers“. In: *European Journal of Operational Research* 213.1, S. 73–82. DOI: 10.1016/j.ejor.2011.03.019.
- (2018). „An exact composite lower bound strategy for the resource-constrained project scheduling problem“. In: *Computers & Operations Research* 93, S. 135–150. DOI: 10.1016/j.cor.2018.01.017.
- (2020). „Going to the core of hard resource-constrained project scheduling instances“. In: *Computers & Operations Research* 121, S. 104976. DOI: 10.1016/j.cor.2020.104976.
- Davari, Morteza u. a. (2020). „Minimizing makespan on a single machine with release dates and inventory constraints“. In: *European Journal of Operational Research* 286.1, S. 115–128. DOI: 10.1016/j.ejor.2020.03.029.
- Debels, Dieter u. a. (2006). „A hybrid scatter search/electromagnetism meta-heuristic for project scheduling“. In: *European Journal of Operational Research* 169.2, S. 638–653. DOI: 10.1016/j.ejor.2004.08.020.
- Demeulemeester, Erik und Willy Herroelen (1992). „A branch-and-bound procedure for the multiple resource-constrained project scheduling problem“. In: *Management Science* 38.12, S. 1803–1818. DOI: 10.1287/mnsc.38.12.1803.
- (1996). „An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem“. In: *European Journal of Operational Research* 90.2, S. 334–348. DOI: 10.1016/0377-2217(95)00358-4.

- Demeulemeester, Erik, Willy Herroelen und Salah E. Elmaghraby (1996). „Optimal procedures for the discrete time/cost trade-off problem in project networks“. In: *European Journal of Operational Research* 88.1, S. 50–68. DOI: 10.1016/0377-2217(94)00181-2.
- Deutscher Bundestag (1994a). *Bericht der Enquete-Kommission „Schutz des Menschen und der Umwelt – Bewertungskriterien und Perspektiven für umweltverträgliche Stoffkreisläufe in der Industriegesellschaft“*. Bonn. URL: <https://dserver.bundestag.de/btd/12/082/1208260.pdf> (besucht am 07.02.2023).
- Deutscher Bundestag (Enquete-Kommission des Deutschen Bundestages „Schutz des Menschen und der Umwelt“) (1994b). *Verantwortung für die Zukunft – Wege zum nachhaltigen Umgang mit Stoff- und Materialströmen*. Bonn.
- DIN Deutsches Institut für Normung (2016). *Leitlinien Projektmanagement (ISO 21500:2012)*. Berlin.
- E.ON Kernkraft GmbH (2008). *Stade: Stilllegung und Rückbau des Kernkraftwerks – vom Kernkraftwerk zur Grünen Wiese*. Hannover. URL: [https://www.atommuellreport.de/fileadmin/Dateien/pdf/Datenblaetter/Stade/EON\\_Stade-Rueckbau\\_Unt\\_2008-03.pdf](https://www.atommuellreport.de/fileadmin/Dateien/pdf/Datenblaetter/Stade/EON_Stade-Rueckbau_Unt_2008-03.pdf) (besucht am 07.02.2023).
- Elmaghraby, Salah E. (1977). *Activity networks: Project planning and control by network models*. A Wiley-Interscience publication. New York: Wiley. ISBN: 0-471-23861-9.
- Elmaghraby, Salah E. und Jerzy Kamburowski (1992). „The analysis of activity networks under generalized precedence relations (GPRs)“. In: *Management Science* 38.9, S. 1245–1263. DOI: 10.1287/mnsc.38.9.1245.
- EnBW Kernkraft GmbH (2018). *Stilllegung und Abbau von Anlagenteilen des Kernkraftwerks Philippsburg Block 2: Kurzbeschreibung*. URL: [https://um.baden-wuerttemberg.de/fileadmin/redaktion/m-um/intern/Dateien/Dokumente/3\\_Umwelt/Kernenergie/Genehmigungsverfahren/KKP/KKP\\_2/180116\\_Kurzbeschreibung\\_Stilllegung\\_und\\_Abbau\\_KKP2.pdf](https://um.baden-wuerttemberg.de/fileadmin/redaktion/m-um/intern/Dateien/Dokumente/3_Umwelt/Kernenergie/Genehmigungsverfahren/KKP/KKP_2/180116_Kurzbeschreibung_Stilllegung_und_Abbau_KKP2.pdf) (besucht am 07.02.2023).
- (o. D.). *Ablauf der Reststoffbearbeitung*. URL: <https://www.enbw.com/media/konzern/images/kernkraft/einrichtungen-der-reststoffbearbeitungszentren-kkp-gkn.pdf> (besucht am 07.02.2023).
- Europäische Kommission (2016). *Commission staff working document accompanying the document communication from the commission: Nuclear illustrative programme presented under article 40 of the euratom treaty for the opinion of the european economic and social committee; SWD/2016/0102 final*. URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=SWD:2016:102:FIN> (besucht am 07.02.2023).
- Europäische Union (2020). *Verordnung (EU) 2020/852 des Europäischen Parlaments und des Rates vom 18. Juni 2020 über die Errichtung eines Rahmens zur Erleichterung nachhaltiger Investitionen und zur Änderung der Verordnung (EU) 2019/2088*. URL: <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32020R0852> (besucht am 07.02.2023).
- Franck, Birger, Klaus Neumann und Christoph Schwindt (2001). „Truncated branch-and-bound, schedule-construction, and schedule-improvement procedures for resource-constrained project scheduling“. In: *OR Spektrum* 23.3, S. 297–324. DOI: 10.1007/PL00013356.
- Fröhling, Magnus (2006). „Zur taktisch-operativen Planung stoffstrombasierter Produktionssysteme: Dargestellt an Beispielen aus der stoffumwandelnden Industrie“. Diss.

- 
- Wiesbaden: Universität Karlsruhe (TH). ISBN: 978-3-8350-0449-8. DOI: 10.1007/978-3-8350-9294-5.
- Frühwirth, Thom und Slim Abdennadher (2003). *Essentials of constraint programming*. Cognitive technologies. Berlin, Heidelberg: Springer. ISBN: 978-3-540-67623-2. DOI: 10.1007/978-3-662-05138-2.
- Garey, Michael R. und David S. Johnson (1979). *Computers and intractability: A guide to the theory of NP-completeness*. A series of books in the mathematical sciences. New York: Freeman. ISBN: 978-0-7167-1045-5.
- Gehring, Marco, Rebekka Volk und Frank Schultmann (2022). „On the integration of diverging material flows into resource-constrained project scheduling“. In: *European Journal of Operational Research* 303.3, S. 1071–1087. DOI: 10.5445/IR/1000145307.
- Gehring, Marco, Franziska Winkler u. a. (2021). *Projektmanagementsoftware und Scheduling: Aktuelle Bestandsaufnahme von Funktionalitäten und Identifikation von Potenzialen*. Working Paper Series In Production and Energy, No. 60. Karlsruher Institut für Technologie (KIT). DOI: 10.5445/IR/1000138068.
- Geiger, Martin Josef (2017). „A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem“. In: *European Journal of Operational Research* 256.3, S. 729–741. DOI: 10.1016/j.ejor.2016.07.024.
- Gerhards, Patrick (2020). „The multi-mode resource investment problem: a benchmark library and a computational study of lower and upper bounds“. In: *OR Spectrum* 42.4, S. 901–933. DOI: 10.1007/s00291-020-00595-9.
- Günther, Hans-Otto (1992). „Netzplanorientierte Auftragsterminierung bei offener Fertigung“. In: *OR Spektrum* 14.4, S. 229–240. DOI: 10.1007/BF01733409.
- Habibi, Farhad, Farnaz Barzinpour und Seyed Jafar Sadjadi (2018). „Resource-constrained project scheduling problem: Review of past and recent developments“. In: *Journal of Project Management* 3.2, S. 55–88. DOI: 10.5267/j.jpm.2018.1.005.
- Hartmann, Sönke (1998). „A competitive genetic algorithm for resource-constrained project scheduling“. In: *Naval Research Logistics (NRL)* 45.7, S. 733–750. DOI: 10.1002/(SICI)1520-6750(199810)45:7<733::AID-NAV5>3.0.CO;2-C.
- Hartmann, Sönke und Rainer Kolisch (2000). „Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem“. In: *European Journal of Operational Research* 127.2, S. 394–407. DOI: 10.1016/S0377-2217(99)00485-3.
- He, Zhengwen, Nengmin Wang und Renjing Liu (2015). „The multi-mode capital-constrained net present value problem“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 513–532. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Herroelen, Willy (2005). „Project scheduling – theory and practice“. In: *Production and Operations Management* 14.4, S. 413–432. DOI: 10.1111/j.1937-5956.2005.tb00230.x.
- Hofstedt, Petra und Armin Wolf (2007). *Einführung in die Constraint-Programmierung: Grundlagen, Methoden, Sprachen, Anwendungen*. eXamen.press. Berlin, Heidelberg: Springer. ISBN: 978-3-540-23184-4. DOI: 10.1007/978-3-540-68194-6.
- Hübner, Felix (2019). „Planung und Modellierung des Rückbaus kerntechnischer Anlagen unter der Berücksichtigung von Unsicherheiten – Ein Beispiel zur Planung von

- Großprojekten“. Diss. Karlsruhe: Karlsruher Institut für Technologie (KIT). ISBN: 978-3-7315-0911-0. DOI: 10.5445/KSP/1000091848.
- Hübner, Felix u. a. (2021). „Solving the nuclear dismantling project scheduling problem by combining mixed-integer and constraint programming techniques and metaheuristics“. In: *Journal of Scheduling* 24.3, S. 269–290. DOI: 10.5445/IR/1000131337.
- Hwang, Ching-Lai und Abu Syed Md. Masud (1979). *Multiple Objective Decision Making – Methods and Applications*. Bd. 164. Berlin, Heidelberg: Springer. ISBN: 978-3-540-09111-0. DOI: 10.1007/978-3-642-45511-7.
- IBM (2019). *Constraint programming with CP Optimizer: Overview*. URL: <https://www.ibm.com/docs/en/icos/12.9.0?topic=cpcp-overview> (besucht am 07.02.2023).
- Jaehn, Florian und Erwin Pesch (2019). *Ablaufplanung: Einführung in Scheduling*. Berlin, Heidelberg: Springer. ISBN: 978-3-662-58779-9. DOI: 10.1007/978-3-662-58780-5.
- Kallrath, Josef (2002). „Planning and scheduling in the process industry“. In: *OR Spectrum* 24.3, S. 219–250. DOI: 10.1007/s00291-002-0101-7.
- Kelley, James E. (1961). „Critical-path planning and scheduling: Mathematical Basis“. In: *Operations Research* 9.3, S. 296–320. DOI: 10.1287/opre.9.3.296.
- Kloock, Josef (1969). *Betriebswirtschaftliche Input-Output-Modelle: Ein Beitrag zur Produktionstheorie*. Bd. 12. Betriebswirtschaftliche Beiträge. Wiesbaden: Gabler Verlag. ISBN: 978-3-663-03117-8. DOI: 10.1007/978-3-663-04306-5.
- Kolisch, Rainer (1995). *Project scheduling under resource constraints*. Heidelberg: Physica. ISBN: 978-3-7908-0829-2. DOI: 10.1007/978-3-642-50296-5.
- (1996a). „Efficient priority rules for the resource-constrained project scheduling problem“. In: *Journal of Operations Management* 14.3, S. 179–192. DOI: 10.1016/0272-6963(95)00032-1.
- (1996b). „Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation“. In: *European Journal of Operational Research* 90.2, S. 320–333. DOI: 10.1016/0377-2217(95)00357-6.
- (2015). „Shifts, types, and generation schemes for project schedules“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 3–16. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Kolisch, Rainer und Sönke Hartmann (1999). „Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis“. In: *Project Scheduling: Recent Models, Algorithms and Applications*. Hrsg. von Jan Węglarz. Boston, MA: Springer US, S. 147–178. ISBN: 978-1-4615-5533-9. DOI: 10.1007/978-1-4615-5533-9\_7.
- (2006). „Experimental investigation of heuristics for resource-constrained project scheduling: An update“. In: *European Journal of Operations Research* 174.1, S. 23–37. DOI: 10.1016/j.ejor.2005.01.065.
- Kolisch, Rainer und Arno Sprecher (1996). „PSPLIB – a project scheduling problem library“. In: *European Journal of Operational Research* 96, S. 205–216. DOI: 10.1016/S0377-2217(96)00170-1.
- (2005a). *Project scheduling problem library – PSPLIB*. URL: <http://www.om-db.wi.tum.de/psplib/main.html> (besucht am 07.02.2023).

- 
- (2005b). *Project scheduling problem library – PSPLIB, Parameter settings*. URL: <http://www.om-db.wi.tum.de/psplib/files/par.sm> (besucht am 07.02.2023).
- Kolisch, Rainer, Arno Sprecher und Andreas Drexl (1995). „Characterization and generation of a general class of resource-constrained project scheduling problems“. In: *Management Science* 41.10, S. 1693–1703. DOI: 10.1287/mnsc.41.10.1693.
- Kondili, Emilia, Constantinos C. Pantelides und Roger W. H. Sargent (1993). „A general algorithm for short-term scheduling of batch operations – I. MILP formulation“. In: *Computers & Chemical Engineering* 17.2, S. 211–227. DOI: 10.1016/0098-1354(93)80015-F.
- Koné, Oumar u. a. (2013). „Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources“. In: *Flexible Services and Manufacturing Journal* 25.1-2, S. 25–47. DOI: 10.1007/s10696-012-9152-5.
- Kreislaufwirtschaft Bau (2021). *Mineralische Bauabfälle Monitoring 2018: Bericht zum Aufkommen und zum Verbleib mineralischer Bauabfälle im Jahr 2018*. Hrsg. von Bundesverband Baustoffe – Steine und Erden e. V. Berlin. URL: <https://kreislaufwirtschaftbau.de/Download/Bericht-12.pdf> (besucht am 07.02.2023).
- Kripfganz, Jochen und Holger Perlt (2020). *Praktische Komplexitätstheorie in Beispielen: Ein interaktives E-Book*. Hanser eLibrary. München: Hanser. ISBN: 9783446467088. DOI: 10.3139/9783446465329.
- Kyriakidis, Thomas S., Georgios M. Kopanos und Michael C. Georgiadis (2012). „MILP formulations for single- and multi-mode resource-constrained project scheduling problems“. In: *Computers & Chemical Engineering* 36, S. 369–385. DOI: 10.1016/j.compchemeng.2011.06.007.
- Laborie, Philippe (2003). „Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results“. In: *Artificial Intelligence* 143, S. 151–188. DOI: 10.1016/S0004-3702(02)00362-4.
- (2009). „IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems“. In: *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*. Hrsg. von Willem-Jan van Hoesve und John N. Hooker. Berlin, Heidelberg: Springer, S. 148–162. ISBN: 978-3-642-01929-6. DOI: 10.1007/978-3-642-01929-6\_12.
- Laborie, Philippe u. a. (2018). „IBM ILOG CP Optimizer for scheduling“. In: *Constraints* 23.2, S. 210–250. DOI: 10.1007/s10601-018-9281-x.
- Laufs, Paul (2018). „Der Rückbau von Kernkraftwerken und die Entsorgung radioaktiver Abfälle“. In: *Reaktorsicherheit für Leistungskernkraftwerke 2*. Hrsg. von Paul Laufs. Berlin: Springer Vieweg, S. 321–475. ISBN: 978-3-662-54164-7. DOI: 10.1007/978-3-662-54164-7\_7.
- Liberatore, Matthew J., Bruce Pollack-Johnson und Colleen A. Smith (2001). „Project management in construction: Software use and research directions“. In: *Journal of Construction Engineering and Management* 127.2, S. 101–107. DOI: 10.1061/(ASCE)0733-9364(2001)127:2(101).
- Méndez, Carlos A. u. a. (2006). „State-of-the-art review of optimization methods for short-term scheduling of batch processes“. In: *Computers & Chemical Engineering* 30.6-7, S. 913–946. DOI: 10.1016/j.compchemeng.2006.02.008.

- Mika, Marek, Grzegorz Waligóra und Jan Węglarz (2008). „Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times“. In: *European Journal of Operational Research* 187.3, S. 1238–1250. DOI: 10.1016/j.ejor.2006.06.069.
- Möhring, Rolf H. (1984). „Minimizing costs of resource requirements in project networks subject to a fixed completion time“. In: *Operations Research* 32.1, S. 89–120. DOI: 10.1287/opre.32.1.89.
- Morsy, Ehab und Erwin Pesch (2015). „Approximation algorithms for inventory constrained scheduling on a single machine“. In: *Journal of Scheduling* 18.6, S. 645–653. DOI: 10.1007/s10951-015-0433-1.
- Müller, Anette (2018). *Baustoffrecycling: Entstehung – Aufbereitung – Verwertung*. Springer-Link Bücher. Wiesbaden: Springer Vieweg. ISBN: 978-3-658-22988-7. DOI: 10.1007/978-3-658-22988-7.
- Murata, Tadao (1989). „Petri nets: Properties, analysis and applications“. In: *Proceedings of the IEEE* 77.4, S. 541–580. DOI: 10.1109/5.24143.
- Neumann, Klaus und Christoph Schwindt (1997). „Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production“. In: *OR Spektrum* 19.3, S. 205–217. DOI: 10.1007/BF01545589.
- (2002). „Project scheduling with inventory constraints“. In: *Mathematical Methods of Operations Research* 56, S. 513–533. DOI: 10.1007/s001860200251.
- Neumann, Klaus, Christoph Schwindt und Norbert Trautmann (2005). „Scheduling of continuous and discontinuous material flows with intermediate storage restrictions“. In: *European Journal of Operational Research* 165.2, S. 495–509. DOI: 10.1016/j.ejor.2004.04.018.
- Neumann, Klaus, Christoph Schwindt und Jürgen Zimmermann (2003). *Project scheduling with time windows and scarce resources: Temporal and resource-constrained project scheduling with regular and nonregular objective functions*. 2. ed. Berlin: Springer. ISBN: 3-540-40125-3. DOI: 10.1007/978-3-540-24800-2.
- Nuclear Energy Agency (2022). *Feedback mechanisms for lessons learnt in the decommissioning of nuclear facilities: An international snapshot*. URL: [https://www.oecd-nea.org/jcms/pl\\_76191/](https://www.oecd-nea.org/jcms/pl_76191/) (besucht am 07.02.2023).
- Patterson, James H. u. a. (1989). „An algorithm for a general class of precedence and resource constrained scheduling problems“. In: *Advances in Project Scheduling*. Hrsg. von Roman Slowinski und Jan Węglarz. Studies in Production and Engineering Economics. Burlington: Elsevier Science, S. 3–28. ISBN: 978-0-444-87358-3. DOI: 10.1016/C2009-0-08770-4.
- Pellerin, Robert, Nathalie Perrier und François Berthaut (2020). „A survey of hybrid metaheuristics for the resource-constrained project scheduling problem“. In: *European Journal of Operational Research* 280.2, S. 395–416. DOI: 10.1016/j.ejor.2019.01.063.
- Pinedo, Michael (2018). *Scheduling: Theory, algorithms, and systems*. Fifth edition. Cham, Heidelberg, New York, Dordrecht, London: Springer. ISBN: 978-3-319-26580-3. DOI: 10.1007/978-3-319-26580-3.
- Preussen Elektra (2020). *Kernkraftwerk Stade*. URL: <https://www.preussenelektra.de/de/unsere-kraftwerke/kraftwerkstade.html> (besucht am 07.02.2023).



- 
- Pritsker, A. Alan B., Lawrence J. Waiters und Philip M. Wolfe (1969). „Multiproject scheduling with limited resources: A zero-one programming approach“. In: *Management Science* 16.1, S. 93–108.
- Quintanilla, Sacramento u. a. (2015). „Integer preemption problems“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 231–250. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Reyck, Bert de und Willy Herroelen (1999). „The multi-mode resource-constrained project scheduling problem with generalized precedence relations“. In: *European Journal of Operational Research* 119.2, S. 538–556. DOI: 10.1016/S0377-2217(99)00151-4.
- Rosen, Anja (2021). „Urban Mining Index: Entwicklung einer Systematik zur quantitativen Bewertung der Kreislaufkonsistenz von Baukonstruktionen in der Neubauplanung“. Diss. Bergische Universität Wuppertal. ISBN: 978-3-7388-0606-9. URL: <https://www.baufachinformation.de/urban-mining-index/buecher/253987> (besucht am 07.02.2023).
- Schnell, Alexander und Richard F. Hartl (2016). „On the efficient modeling and solution of the multi-mode resource-constrained project scheduling problem with generalized precedence relations“. In: *OR Spectrum* 38.2, S. 283–303. DOI: 10.1007/s00291-015-0419-6.
- (2017). „On the generalization of constraint programming and boolean satisfiability solving techniques to schedule a resource-constrained project consisting of multi-mode jobs“. In: *Operations Research Perspectives* 4, S. 1–11. DOI: 10.1016/j.orp.2017.01.002.
- Schultmann, Frank (1998). „Kreislaufführung von Baustoffen: Stoffflußbasiertes Projektmanagement für die operative Demontage- und Recyclingplanung von Gebäuden“. Diss. Berlin: Universität Karlsruhe (TH). ISBN: 978-3-503-05021-5.
- (2003). *Stoffstrombasiertes Produktionsmanagement: Betriebswirtschaftliche Planung und Steuerung industrieller Kreislaufwirtschaftssysteme*. Bd. 58. Technological economics. Berlin: Schmidt. ISBN: 978-3-503-07803-5.
- Schwindt, Christoph (1999). „A branch-and-bound algorithm for the project duration problem subject to temporal and cumulative resource constraints“. In: *Proceedings of International Conference on Industrial Engineering and Production Management*. Glasgow. URL: [https://www.wiwi.tu-clausthal.de/fileadmin/WIWI/documents/BWL\\_Produktion\\_und\\_Logistik/Download/iepm\\_1999.pdf](https://www.wiwi.tu-clausthal.de/fileadmin/WIWI/documents/BWL_Produktion_und_Logistik/Download/iepm_1999.pdf) (besucht am 07.02.2023).
- Schwindt, Christoph, Rafael Fink und Norbert Trautmann (2007). „A priority-rule based method for scheduling in chemical batch production“. In: *IEEE International Conference on Industrial Engineering and Engineering Management, 2007*. Hrsg. von Martin Helander. IEEE Service Center, S. 1347–1351. ISBN: 978-1-4244-1528-1. DOI: 10.1109/IEEM.2007.4419412.
- Schwindt, Christoph und Tobias Paetz (2015). „Continuous preemption problems“. In: *Handbook on Project Management and Scheduling Vol. 1*. Hrsg. von Christoph Schwindt und Jürgen Zimmermann. International Handbooks on Information Systems. Berlin: Springer, S. 251–295. ISBN: 978-3-319-05443-8. DOI: 10.1007/978-3-319-05443-8.
- Schwindt, Christoph und Norbert Trautmann (2000). „Batch scheduling in process industries: An application of resource-constrained project scheduling“. In: *OR Spektrum* 22, S. 501–524. DOI: 10.1007/s002910000042.

- Sourd, Francis und Jérôme Rogerie (2005). „Continuous filling and emptying of storage systems in constraint-based scheduling“. In: *European Journal of Operational Research* 165.2, S. 510–524. DOI: 10.1016/j.ejor.2004.04.019.
- Sprecher, Arno, Sönke Hartmann und Andreas Drexl (1997). „An exact algorithm for project scheduling with multiple modes“. In: *OR Spektrum* 19.3, S. 195–203. DOI: 10.1007/BF01545587.
- Sprecher, Arno, Rainer Kolisch und Andreas Drexl (1995). „Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem“. In: *European Journal of Operational Research* 80.1, S. 94–102. DOI: 10.1016/0377-2217(93)E0294-8.
- Statistisches Bundesamt (2020). *Abfallwirtschaft: Kurzübersicht Abfallbilanz – Zeitreihe*. URL: <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Umwelt/Abfallwirtschaft/Tabellen/liste-abfallbilanz-kurzuebersicht.html?nn=211528> (besucht am 07.02.2023).
- Talbot, F. Brian (1982). „Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case“. In: *Management Science* 28.10, S. 1197–1210. DOI: 10.1287/mnsc.28.10.1197.
- Talbot, F. Brian und James H. Patterson (1978). „An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems“. In: *Management Science* 24.11, S. 1163–1174. URL: <https://www.jstor.org/stable/2630490> (besucht am 07.02.2023).
- Thierfeldt, Stefan und Frank Schartmann (2009). *Stilllegung und Rückbaukerntechnischer Anlagen: Erfahrungen und Perspektiven: Erstellt im Auftrag des Bundesministeriums für Bildung und Forschung*. Aachen. URL: <https://www.ptka.kit.edu/ptka-alt/downloads/ptka-wte-e/WTE-E-Entsorgungsforschung-Broschuere-Stilllegung-und-Rueckbau-BRENK.pdf> (besucht am 07.02.2023).
- Thomas, Paul R. und Said Salhi (1998). „A tabu search approach for the resource constrained project scheduling problem“. In: *Journal of Heuristics* 4.2, S. 123–139. DOI: 10.1023/A:1009673512884.
- Umweltbundesamt (2021). *Bauabfälle*. URL: <https://www.umweltbundesamt.de/daten/ressourcen-abfall/verwertung-entsorgung-ausgewaehlter-abfallarten/bauabfaelle#verwertung-von-bau-und-abbruchabfallen> (besucht am 07.02.2022).
- van Peteghem, Vincent und Mario Vanhoucke (2014). „An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances“. In: *European Journal of Operational Research* 235.1, S. 62–72. DOI: 10.1016/j.ejor.2013.10.012.
- Vanhoucke, Mario und José Coelho (2016). „An approach using SAT solvers for the RCPSP with logical constraints“. In: *European Journal of Operational Research* 249.2, S. 577–591. DOI: 10.1016/j.ejor.2015.08.044.
- Vilím, Petr, Philippe Laborie und Paul Shaw (2015). „Failure-directed search for constraint-based scheduling“. In: *Integration of AI and or Techniques in Constraint Programming*. Hrsg. von Laurent Michel. Bd. 9075. Lecture Notes in Computer Science. Cham: Springer, S. 437–453. ISBN: 978-3-319-18007-6. DOI: 10.1007/978-3-319-18008-3\_30.
- Volk, Rebekka (2017). „Proactive-reactive, robust scheduling and capacity planning of deconstruction projects under uncertainty“. Diss. Karlsruhe: Karlsruher Institut für Technologie (KIT). ISBN: 978-3-7315-0592-1. DOI: 10.5445/KSP/1000060265.

- 
- Volk, Rebekka u. a. (2019). „The future of nuclear decommissioning – a worldwide market potential study“. In: *Energy Policy* 124, S. 226–261. DOI: 10.1016/j.enpol.2018.08.014.
- Voß, Stefan und Andreas Witt (2007). „Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application“. In: *International Journal of Production Economics* 105.2, S. 445–458. DOI: 10.1016/j.ijpe.2004.05.029.
- Voudouris, Vasilios T. und Ignacio E. Grossmann (1996). „MILP model for scheduling and design of a special class of multipurpose batch plants“. In: *Computers & Chemical Engineering* 20.11, S. 1335–1360. DOI: 10.1016/0098-1354(96)81812-6.
- Węglarz, Jan u. a. (2011). „Project scheduling with finite or infinite number of activity processing modes – A survey“. In: *European Journal of Operational Research* 208.3, S. 177–205. DOI: 10.1016/j.ejor.2010.03.037.
- Zimmermann, Jürgen, Julia Rieck und Christoph Stark (2006). *Projektplanung: Modelle, Methoden, Management: mit 80 Tabellen*. Springer-Lehrbuch. Berlin, Heidelberg, New York: Springer. ISBN: 978-3-540-28413-0. DOI: 10.1007/3-540-28483-4.