

# Ansatz zur dreidimensionalen Layoutplanung bei heterogener Raumgröße

Zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften**

der Fakultät für Maschinenbau  
des Karlsruher Instituts für Technologie (KIT)  
genehmigte

**Dissertation**

von

**Dipl.-Wi.-Ing. Martin Kubaschewski**

aus Bad Dürkheim

Tag der mündlichen Prüfung:  
Hauptreferent:  
Korreferent:

22. Januar 2016  
Prof. Dr.-Ing. Kai Furmans  
Prof. Dr. Stefan Nickel



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung –  
Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz  
(CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>

# Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Fördertechnik und Logistiksysteme des Karlsruher Instituts für Technologie.

Ich bedanke mich herzlich bei meinem Doktorvater Prof. Dr.-Ing. Kai Furmans für die Betreuung meiner Arbeit. Seine Impulse und die gewährten Freiheiten bei der Erarbeitung des Themas haben zum Gelingen wesentlich beigetragen und meine Freude am wissenschaftlichen Arbeiten gefördert.

Ebenso möchte ich mich sehr herzlich bei Prof. Dr. Stefan Nickel für die Übernahme des Korreferats sowie bei Prof. Dr. Robert Stieglitz für die Übernahme des Prüfungsvorsitzes bedanken.

Bei allen Kollegen des IFL möchte ich mich für die gemeinsame Zeit und Zusammenarbeit bedanken. Mein besonderer Dank gilt Herrn Dr.-Ing. Martin Mittwollen. Die angenehme Arbeitsatmosphäre am Institut und den freundschaftlichen Umgang werde ich immer in guter Erinnerung behalten.

Ebenso möchte ich mich bei meinen Eltern für die Unterstützung und den Rückhalt auf dem Weg durch mein Studium und während der Promotion bedanken.

Ganz Besonders bedanken möchte ich mich bei meiner Frau Katharina für alle Ihre Unterstützung, Verständnis und Liebe in den vergangenen Jahren. Ohne sie wäre mir diese Arbeit nicht möglich gewesen.

München, Juli 2016

Martin Kubaschewski



# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>1. Einleitung</b>	<b>5</b>
1.1. Problemstellung und relevante Fragestellungen . . . . .	5
1.2. Aufbau der Arbeit . . . . .	7
<b>2. Layoutplanung in der Ebene und im Raum</b>	<b>9</b>
2.1. Praktische Beispiele für Layoutplanungsprobleme . . . . .	9
2.2. Abgrenzung von Verfahren der Layoutplanung . . . . .	12
2.2.1. Statische und dynamische Probleme . . . . .	13
2.2.2. Größe und Form der Einheiten . . . . .	13
2.2.3. Anzahl der berücksichtigten Dimensionen . . . . .	15
2.2.4. Diskrete und kontinuierliche Modelle . . . . .	16
2.2.5. Stochastische Nachfrage . . . . .	20
2.3. Zweidimensionale Modelle . . . . .	21
2.3.1. Das lineare und das quadratische Zuordnungsproblem .	21
2.3.2. Space Filling Curves . . . . .	25
2.3.3. Graphenbasierte Modelle . . . . .	28
2.3.4. Stetige Modelle . . . . .	30
2.3.5. Layoutplanung unter Berücksichtigung förder- technischer Anlagen . . . . .	35
2.4. Dreidimensionale Modelle . . . . .	37
2.4.1. Ebenenbasierte Modelle . . . . .	38
2.4.2. Echt dreidimensionale Ansätze . . . . .	40
2.4.3. Layoutplanung mit Gleichgewichtsbedingungen . . . .	44
2.5. Expertensysteme . . . . .	45
<b>3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen</b>	<b>47</b>
3.1. Aufbau von IFMIF . . . . .	47

3.2. Grundlagen des Modells . . . . .	49
3.2.1. Eigenschaften der zu planenden Anlagen . . . . .	50
3.2.2. Resultierende Anforderungen an das Modell . . . . .	54
3.3. Eignung bestehender Ansätze . . . . .	57
3.4. Ansatz über Constraintprogrammierung . . . . .	60
3.4.1. Grundlagen von CP . . . . .	60
3.4.2. Anwendung auf das Modell . . . . .	65
3.5. Modellformulierung und Nebenbedingungen . . . . .	65
3.5.1. Berücksichtigte Nebenbedingungen . . . . .	68
<b>4. Lösungsverfahren</b>	<b>79</b>
4.1. Mögliche Lösungsstrategien . . . . .	79
4.1.1. Einflussgrößen der Suchstrategie . . . . .	81
4.1.2. Variablenauswahl . . . . .	83
4.1.3. Wertauswahl . . . . .	84
4.2. Bestimmung unterer Schranken . . . . .	89
4.2.1. Untere Schranken des quadratischen Zuordnungsproblems	90
4.2.2. Untere Schranke für das vorliegende Modell . . . . .	92
4.3. Eignung von Modell und Lösungsstrategie in Bezug auf die Anforderungen . . . . .	95
<b>5. Evaluierung der Ergebnisse</b>	<b>99</b>
5.1. Verwendete Testdaten . . . . .	99
5.1.1. Anwendung auf das quadratische Zuordnungsproblem .	100
5.1.2. Aus dem quadratischen Zuordnungsproblem abgeleitete Instanzen . . . . .	101
5.1.3. Vergleichbare Instanzen alternativer Formulierungen . .	103
5.1.4. Testinstanzen entsprechend der Modellanforderungen .	105
5.2. Ergebnisse der Testläufe . . . . .	107
5.2.1. Auswirkungen der unteren Schranke . . . . .	107
5.2.2. Propagieren möglicher Positionen eines Objekts . . . . .	108
5.2.3. Propagieren der Objekte . . . . .	110
5.2.4. Propagieren der Nachfrage nach einer Position . . . . .	112
5.2.5. Effizienz der Variablen- und Wertauswahl . . . . .	113
5.2.6. Zusammenfassung der Ergebnisse . . . . .	124
5.3. Auswahl der Suchstrategie . . . . .	126
5.3.1. Merkmale der Datensätze . . . . .	126
5.3.2. Einfluss der Kenngrößen auf die Suchstrategie . . . . .	129

<b>6. Zusammenfassung und Ausblick</b>	<b>131</b>
6.1. Zusammenfassung . . . . .	131
6.2. Ausblick . . . . .	134
<b>Literaturverzeichnis</b>	<b>137</b>
<b>Symbolverzeichnis</b>	<b>145</b>
<b>A. Daten der Testinstanzen</b>	<b>147</b>
A.1. Vom quadratischen Zuordnungsproblem abgeleitete Instanzen	147



# Abbildungsverzeichnis

2.1. Verschieben von Flexförderer-Elementen in einem Layout über 3 Perioden . . . . .	14
2.2. Regelmäßige und unregelmäßige Formen der Einheiten . . . . .	14
2.3. Zurücksetzen eines Produkts in einer Produktionslinie . . . . .	16
2.4. Erstmalige optimale Lösung der Instanzen von Nugent . . . . .	24
2.5. Entwicklung der Hilbertkurve (oben) und Z-Kurve (unten) erster bis dritter Ordnung . . . . .	26
2.6. Anordnung von 8 Funktionseinheiten unterschiedlicher Größe entlang einer Hilbertkurve und einer Z-Kurve 3. Ordnung . . . . .	27
2.7. Maximal planarer Graph mit blockförmigen Layout von 6 Funktionseinheiten und Umgebungsknoten $U$ . . . . .	29
2.8. Layouterzeugung mit Slicing Trees und zugehöriger Entscheidungsbaum . . . . .	31
2.9. Layouterzeugung mit Hilfe von Flexible Bays . . . . .	31
2.10. Mögliche Positionierungen der Einheit $i$ innerhalb des Layouts . . . . .	34
2.11. Layout eines Satelliten mit zwei Ebenen, <i>Quelle: Teng 2001, S. 522</i> . . . . .	39
2.12. Aufstellungsplan einer Chemieanlage, <i>Quelle: Köpcke 2004, S. 9</i> . . . . .	41
2.13. Anordnung 8 identischer Würfel, <i>Quelle: Cagan 1998, S. 787</i> . . . . .	42
2.14. Aufbau der Einheiten und Konstruktionsprinzip des Layouts nach Furuholmen . . . . .	43
2.15. Konstruktion eines Layouts aus 100 Einheiten, <i>Quelle: Furuholmen 2010, S.1</i> . . . . .	44
3.1. Funktionsprinzip der Materialforschungsanlage IFMIF . . . . .	48
3.2. Aktuelles Design von IFMIF . . . . .	49
3.3. Anordnung von Objekten unterschiedlicher Größe über eine Variante des quadratischen Zuordnungsproblems . . . . .	58
3.4. Suchbaum eines CSP mit drei Variablen . . . . .	62
3.5. Fortlaufende Indizes der Blöcke innerhalb von Objekten . . . . .	67
3.6. Reduktion des Wertebereichs aufgrund der Maße eines Objekts . . . . .	71

3.7. Objekt der Größe 4x3x3 mit ausgewählten Blöcken zur Bestimmung von Mindestmaßen . . . . .	72
3.8. Raster mit einem Objekt und allen zugehörigen maximalen Unterräumen . . . . .	75
3.9. Bestimmung der Mindestmaße mit und ohne Berücksichtigung der Orientierung eines Objekts . . . . .	77
4.1. Grafische Oberfläche zur Steuerung des Modells . . . . .	80
4.2. Bewertung des Rasters sowie Bewertung der Objekte anhand der Merkmale Flüsse und Objektgröße . . . . .	86
4.3. Bewertung einzelner Rasterpositionen anhand der Äquivalenz von Maßen . . . . .	87
5.1. Aufteilung in Blöcke am Beispiel der Instanz <i>sko100a</i> . . . . .	102
5.2. Aufteilung in Blöcke mit ähnlichen Maßen . . . . .	103
5.3. Optimale Anordnung von 12 Objekten auf 58 Positionen für <i>baz12</i> . . . . .	104
5.4. Vergleich der Lösungssuche für die von <i>sko64</i> abgeleiteten Instanzen mit und ohne Propagieren möglicher Positionen . . . . .	108
5.5. Vergleich der Lösungssuche für die von <i>wil100</i> abgeleiteten Instanzen mit und ohne Propagieren möglicher Positionen . . . . .	110
5.6. Vergleich der Lösungssuche für die von <i>sko64</i> abgeleiteten Instanzen mit und ohne Propagieren der Objekte . . . . .	111
5.7. Vergleich der Lösungssuche für die von <i>wil100</i> abgeleiteten Instanzen mit und ohne Propagieren der Objekte . . . . .	112
5.8. Lösungssuche für die Instanz <i>tho40</i> . . . . .	114
5.9. Lösungssuche für die Instanz <i>wil100</i> . . . . .	115
5.10. Summe der ein- und ausgehenden Flüsse für <i>tho40</i> . . . . .	115
5.11. Möglichkeiten der Variablenauswahl für <i>sko64A</i> bei Anwendung der konstruktiven Wertauswahl . . . . .	117
5.12. Möglichkeiten der Wertauswahl für <i>sko64B</i> bei Variablenauswahl auf Basis von Objekt- und Domänengröße . . . . .	118
5.13. Möglichkeiten der Wertauswahl für <i>sko100A</i> bei Variablenauswahl auf Basis von Objekt- und Domänengröße . . . . .	119
5.14. Lösungssuche für die Instanz <i>sim11<sub>SRFLP</sub></i> . . . . .	120
5.15. Entstehen von Lücken bei der Lösungssuche für Instanzen des SRFLP . . . . .	121
5.16. Verlauf der Lösungssuche für unterschiedliche Strategien im Fall der Instanz <i>k20a2</i> . . . . .	122

5.17. Verlauf der Lösungssuche für unterschiedliche Strategien im Fall der Instanz <i>k20b1</i> . . . . .	123
5.18. Verlauf der Lösungssuche für unterschiedliche Strategien im Fall der Instanz <i>k20b2</i> . . . . .	124
6.1. Iterative Planung von Layout und Fördersystemen . . . . .	136



# Tabellenverzeichnis

2.1. Auswahl von Layoutproblemen und Abgrenzung anhand relevanter Merkmale . . . . .	17
2.2. Größte berechnete Instanzen verschiedener Modelle mit stochastischer Nachfrage . . . . .	21
3.1. Übersicht wichtiger Eigenschaften technischer Anlagen . . . . .	55
3.2. Beispielhafte Bestimmung der Mindestmaße innerhalb eines 4x3x3 Objekts . . . . .	73
5.1. Verwendete Testinstanzen des quadratischen Zuordnungsproblems . . . . .	100
5.2. Maße der Objekte nach der Aufteilung gemäß Abbildung 5.1 . . . . .	102
5.3. Kenngrößen der erstellten Instanzen . . . . .	106
5.4. Beste gefundene Lösungen der erstellten Instanzen . . . . .	106
5.5. Einfluss der Merkmale auf Aspekte der Suchstrategie . . . . .	130



# Kurzfassung

*Martin Kubaschewski*

## **Ansatz zur dreidimensionalen Layoutplanung bei heterogener Raumgröße**

Probleme der Layoutplanung können in zahlreichen unterschiedlichen Bereichen beobachtet werden. Ob beim Produktdesign in der Elektroindustrie oder bei der Planung eines Bürokomplexes, stets ist die Platzierung mehrerer Komponenten innerhalb eines begrenzten Raums das Ziel. Zu beachten sind dabei Beziehungen zwischen den Einheiten, zum Beispiel in Form der Wärmeentwicklung eines Mobiltelefon-Prozessors und dessen Auswirkungen auf andere Komponenten oder den Laufwegen zwischen einzelnen Büros. Für diese Vielzahl an unterschiedlichen Anwendungsformen existiert eine ebenso große Zahl an unterschiedlichen Optimierungsmodellen, die mit analytischen Methoden geeignete Lösungen finden.

In dieser Arbeit liegt der Fokus auf der Layoutplanung großtechnischer Anlagen. Als besondere Charakteristika werden dabei beispielsweise die stark variierenden Größen der Komponenten als auch die auftretenden technischen und wirtschaftlichen Nebenbedingungen Berücksichtigung finden.

Im Anschluss an eine Übersicht bestehender zwei- sowie dreidimensionaler Modelle werden deren spezielle Eigenschaften und Rahmenbedingungen erläutert. Im Weiteren wird ihre Eignung auf den gewünschten Anwendungsfall untersucht. Aufbauend auf einem der Ansätze, wird ein Modell entwickelt, das anschließend mit Hilfe der sogenannten Constraint Programmierung gelöst wird. Abhängig von konkreten Eingangsdaten kann sowohl das Auffinden einer zulässigen Lösung, als auch die Suche nach einer möglichst guten oder optimalen Lösung im Vordergrund stehen. Für beide Ziele werden Methoden entwickelt, die einerseits den Lösungsraum sinnvoll begrenzen und diesen darüber hinaus effizient durchsuchen.

Die dargestellten Lösungsmethoden werden auf eine Auswahl an Testdaten angewendet. Da in der Literatur zu findende Instanzen nur in begrenztem

Rahmen den gewünschten Anforderungen entsprechen, wurden diese teilweise angepasst und zum Teil neu erstellt. Basierend auf den Rechenergebnissen werden Kenngrößen entwickelt, die die unterschiedlichen Datensätze charakterisieren. Aus diesen Parametern wird abschließend eine Entscheidungshilfe für eine passende Suchstrategie entwickelt.

# Abstract

*Martin Kubaschewski*

## **Approach to the three-dimensional facility layout planning with heterogenous room size**

Facility layout problems can be found in many different situations. Whether for the product design in electronic industry or the planning of an office block, the target is always the placing of several components within a limited space. Relations between the single units like the heating of a processor in a cell-phone and its impact on other components or the reduction of walking distances between offices have to be considered. There are as many different optimization models as different areas of application that use analytical methods to find suitable solutions.

This work focuses on the layout planning of large-scale plants. The strongly varying component size as well as technical and economic constraints have to be considered as their main characteristics.

Subsequent to an overview of existing two- and three-dimensional models, their specific characteristics and underlying assumptions are explained. Furthermore, their capability for the above mentioned field of application is discussed. Based on one of the approaches, a model is developed, which is solved by the so-called Constraint Programming. The focus of the optimization process can be on the search for a feasible solution as well as the search for the optimal solution, depending on the concrete input data. Solution strategies have been developed for both targets. These restrict the solution space on the one hand, and search it efficiently for improved solutions.

The presented solution strategies are applied on a selection of test data. As the test instances found in the literature meet the desired requirements only to a low degree, they were partly adapted and partly newly created. Based on calculation results, key parameters were derived from the input data. Finally, a suitable solution strategy is suggested with these parameters.



# 1. Einleitung

Fragestellungen der Layoutplanung treten in zahlreichen Bereichen der Logistik, Produktion, wie auch im täglichen Leben auf. Gemeinsam sind allen Problemen, dass eine größere Zahl von Einheiten auf begrenztem Raum platziert werden soll. In der Regel soll dies auf eine Art und Weise erfolgen, die möglichst kurze Wege zwischen den Einheiten ermöglicht und gewisse zusätzliche Restriktionen berücksichtigt. Es können jedoch auch alternative Ziele wie die Minimierung von Umrüstkosten oder die minimale Abweichung von einer Formvorgabe der Einheiten verfolgt werden.

So müssen sowohl bei der Neuplanung jedes privaten oder öffentlichen Gebäudes als auch bei der Planung von Industrieanlagen Layoutprobleme gelöst werden, damit diese ihre Aufgabe effizient erfüllen. Während im privaten Bereich häufig eher qualitative Kriterien eine Rolle spielen, die sich nur schwer messen oder bewerten lassen, liegen im öffentlichen Sektor oder in Produktion und Logistik in der Regel vermehrt quantitative Kriterien zugrunde, deren Optimierung im Vordergrund steht. Ein klassisches Problem basiert beispielsweise auf der Anordnung von 30 Organisationseinheiten des Klinikums Regensburg und soll die dort auftretenden Patientenströme minimieren (Hahn und Krarup 2001).

In der Einleitung sollen nun zunächst die konkrete Problemstellung näher erläutert sowie die dabei relevanten Fragebereiche skizziert werden. Im Anschluss wird in Abschnitt 1.2 der Aufbau der Arbeit beschrieben.

## 1.1. Problemstellung und relevante Fragestellungen

Diese Arbeit wurde motiviert durch Erfahrungen bei der Planung der Fusionsforschungsanlage IFMIF (*International Fusion Materials Irradiation Facility*). Die international verteilten Design-Aufgaben an einzelnen Teilbereichen ließen die Gesamtplanung des Gebäudes als effizientes Zusammenspiel der Teilbereiche in den Hintergrund treten. Aus diesem Grund existierten für die-

se Integrationsarbeiten nur noch geringe Spielräume, so dass damit zu rechnen ist, dass die Leistungsfähigkeit der logistischen Prozesse innerhalb des Gebäudes unbefriedigend ist. Jedoch traten auch bereits in einzelnen Teilbereichen der Planung Fragestellungen bezüglich eines (dreidimensionalen) Layouts auf, bei denen rechnergestützte Modelle wertvolle Ergebnisse hätten liefern können.

Aus diesem Grund sollen in dieser Arbeit Fragestellungen behandelt werden, die mit der Layoutgenerierung derartiger Anlagen im Kontext stehen. Ziel soll es dabei sein, ein Modell mit starkem praktischem Bezug zu entwickeln. Technische Bedingungen, die die Entscheidungen der an einem Projekt beteiligten Ingenieure beeinflussen, sollen abbildbar und ihre Auswirkungen am Ergebnis sichtbar sein. Die Fragestellungen lassen sich dabei in die folgenden Kategorien einteilen:

### **Themenbereich 1: Eigenschaften existierender Modelle**

Zunächst soll untersucht werden in welchen praktischen Situationen Layoutprobleme auftreten und in wieweit rechnergestützte Methoden bei deren Lösung zur Anwendung kommen. Ein besonderer Fokus soll dabei auch auf problemspezifischen Merkmalen der Modelle liegen, die für die Anwendung interessant sind. Da auch eine Adaption einer Lösungsstrategie für den zweidimensionalen Fall eine mögliche Variante ist, soll auch die Vielzahl dieser Modelle betrachtet werden.

### **Themenbereich 2: Anforderungen an ein Modell**

Welche Charakteristika weisen IFMIF oder vergleichbare großtechnische Anlagen auf und müssen in einem Modell berücksichtigt werden? Wie weit werden diese Eigenschaften in bisherigen Modellen berücksichtigt? Diese beiden Fragen sollen den Bezug zwischen dem praktischen Anwendungsfall und den in der Literatur zu findenden Optimierungsmodellen herstellen. Dabei muss ebenso berücksichtigt werden, welche vereinfachenden Annahmen getroffen werden können und welche Eigenschaften so charakteristisch für den betrachteten Anwendungsfall sind, dass sie Eingang in ein Modell finden müssen. Ferner wird ebenso die Frage beantwortet, welcher Optimierungsansatz dabei in der Lage ist, die Realität hinreichend genau wiederzugeben.

### **Themenbereich 3: Aufbau eines Suchalgorithmus**

Im Weiteren muss die Frage beantwortet werden, welche Verfahren bei der Suche nach einer Lösung angewendet werden sollen. Dabei sollen sowohl die Zulässigkeit von Lösungen als auch die Optimalität berücksichtigt werden. Welcher Ansatz verfolgt wird, hängt dabei stark von den konkreten Eingangsdaten ab. Für beide Varianten stellt sich die Frage, anhand welcher Kriterien sich die Suche steuern lässt.

### **Themenbereich 4: Reduktion des Lösungsraums**

Um in angemessener Zeit Lösungen zu finden, ist es notwendig, auf Basis der gegebenen Informationen den Lösungsraum sinnvoll einzugrenzen. Dabei sollte im Suchprozess die Nichterfüllbarkeit einzelner Nebenbedingungen frühzeitig erkannt werden und entsprechend ein begonnener Lösungsansatz verworfen werden. Ebenso zählen dazu unter dem Gesichtspunkt der Optimalität das Abschätzen der erreichbaren Lösungsgüte und ein Abbruch, falls diese über einer bereits erreichten Lösung liegt.

### **Themenbereich 5: Einfluss unterschiedlicher Parameter**

Das Modell sollte flexibel genug sein, um auf variierende Problemstellungen und unterschiedliche Eingangsgrößen reagieren zu können. In diesem Rahmen stellt sich die Frage, anhand welcher Parameter der Suchprozess gesteuert werden soll. So müssen für einen Datensatz Kenngrößen bestimmt werden, aus denen eine Empfehlung für eine bestimmte Lösungsstrategie getroffen werden kann.

## **1.2. Aufbau der Arbeit**

Die Arbeit beginnt mit einem Überblick über bestehende Layoutplanungsmodelle und deren Anwendungsgebiete. Dabei werden in Kapitel 2 zunächst eine Reihe von Merkmalen erläutert, anhand derer sich bestehende Ansätze klassifizieren lassen. Im Anschluss werden die verbreitetsten zwei- sowie dreidimensionalen Ansätze kurz vorgestellt. Kapitel 3 stellt zunächst die Fusionsforschungsanlage IFMIF als motivierendes Beispiel kurz vor. Im Anschluss daran werden allgemein die Eigenschaften dieser und ähnlicher Anlagen analysiert und resultierende Anforderungen an ein Optimierungsmodell formuliert. Nach

einer Reflexion der Eignung bestehender Ansätze wird daraufhin der gewählte Ansatz über die sogenannte Constraint Programmierung erläutert und das verwendete Modell mit seinen Nebenbedingungen beschrieben. Aufbauend auf diesem Modell widmet sich Kapitel 4 den implementierten Lösungsstrategien und benennt deren jeweilige Einflussgrößen. Im Anschluss wird kurz die Eignung der Strategien in Hinblick auf die im vorigen Kapitel formulierten Modellanforderungen betrachtet. Das folgende Kapitel 5 beschreibt zunächst die gewählten beziehungsweise erstellten Testdatensätze um im Weiteren die daraus gewonnenen Erkenntnisse zu beschreiben. Zum Abschluss erfolgt in Kapitel 6 eine Zusammenfassung der Ergebnisse der Arbeit, sowie ein Ausblick auf weitere Forschungsthemen im Bereich der dreidimensionalen Layoutplanung.

## 2. Layoutplanung in der Ebene und im Raum

In diesem Kapitel sollen zur Verdeutlichung der Problembedeutung in Abschnitt 2.1 zunächst einige praktische Beispiele erläutert werden. Dies soll ein allgemeines Verständnis darüber geben, welche Umgebungsbedingungen für Optimierungsmodelle gelten und in welchen unterschiedlichen Formen Layoutplanungsprobleme auftreten können. Die Eigenschaften der darauf aufbauenden Modelle werden im Anschluss in Abschnitt 2.2 erläutert. Anhand dieser Charakteristika können die in den Abschnitten 2.3 und 2.4 vorgestellten zwei- beziehungsweise dreidimensionalen Layoutplanungsmodelle klassifiziert werden. Die angegebenen Modelle sollen einerseits die unterschiedlichen Herangehensweisen an das Problem beschreiben und zum anderen als Basis für das zu entwickelnde Modell dienen. Am Ende des Kapitels werden kurz auf Layoutplanungsmodellen aufbauende Expertensysteme beschrieben, die bestehende Ansätze integrieren und auf Basis der Eingabedaten ein Entscheidungssystem zur Layoutgenerierung bereitstellen.

### 2.1. Praktische Beispiele für Layoutplanungsprobleme

Fragestellungen der Layoutplanung treten in zahlreichen Bereichen der Logistik, Produktion wie auch im täglichen Leben auf. Gemeinsam sind allen Problemen, dass eine größere Zahl von Einheiten auf begrenztem Raum platziert werden sollen. In der Regel soll dies auf eine Art und Weise erfolgen, die möglichst kurze Wege zwischen den Einheiten ermöglicht und gewisse zusätzliche Restriktionen berücksichtigt. Es können jedoch auch alternative Ziele wie die Minimierung von Umrüstkosten oder die minimale Abweichung von einer Formvorgabe der Einheiten verfolgt werden.

In der Literatur existiert keine einheitliche Definition von Layoutproblemen. Beschränkt man sich zunächst auf den Bereich der Planung des Layouts eines

Betriebes (engl. *facility layout problem*), versteht man unter einem Layout die „Anordnung aller zur Produktion von Gütern oder Erbringung einer Leistung notwendigen Einrichtungen“ (Drira et al. 2007, S. 255). Dabei können auch Teilmengen einer Produktionsanlage als abgeschlossener Bereich, der eine bestimmte Leistung erbringt, betrachtet werden.

Im Bereich der Produktion findet eine Layoutplanung beispielsweise beim Aufstellen von Maschinen in einer Werkshalle statt. Bei diesem klassischen Beispiel muss entschieden werden, an welcher Stelle der Halle welche Maschine platziert werden soll. Je nach Charakteristik des Planungsproblems können unterschiedliche mathematische Formulierungen zur Abbildung in Frage kommen. Ausschlaggebend können hierfür unter anderem die Art der Fertigung (Fließfertigung oder Inselfertigung) sein. Gerade im Bereich der Produktion werden mögliche Maschinenaufstellungen durch eine Vielzahl von Restriktionen begrenzt. Darunter fallen zum Beispiel:

- Die Anbindung bestimmter Maschinen an bereits bestehende Versorgungssysteme, wie zum Beispiel Kühlmedien oder fördertechnische Einrichtungen.
- Die Vorgabe bestimmter Positionen aus statischen Gründen, zum Beispiel aufgrund des Verlaufs von Stahlträgern.
- Ein Mindestabstand zwischen Maschinen aufgrund einer gegenseitigen Beeinflussung, wie der Vermeidung von Vibrationen beim Ausführen hochpräziser Abläufe.
- Bei flüssigen oder gasförmigen Medien in Rohren technische Restriktionen aufgrund der Leistung von Pumpen.
- Die Bildung eines getrennten Sicherheitsbereichs, der vom Personal nicht betreten werden darf, beispielsweise die Einrichtung von Sicherheitszellen in einer automatisierten Produktion.

Ähnliche Problemstellungen ergeben sich auch im Bereich der Logistik bei der Planung von Distributionszentren. Neben der Berücksichtigung der einzelnen Komponenten und technischen Einrichtungen, spielt in diesem Bereich häufig auch die Erweiterbarkeit eine tragende Rolle.

Sowohl im Bereich der Produktion als auch der Logistik ist die Erhebung der Daten, die für eine analytische Berechnung notwendig sind, mit erheblichen Schwierigkeiten verbunden. Wird eine Minimierung des Transportaufwands als Ziel verfolgt, müssen für die Bestimmung der zu Grunde liegenden Transportmatrix unter anderem folgende Punkte berücksichtigt werden (Heragu 2008):

- Art und Menge der Produkte
- Fertigungs- oder Bearbeitungsschritte
- Notwendige Ausstattung mit Fördertechnik
- Notwendige Kapazitäten identischer Maschinen
- Gestaltung einzelner Fertigungszellen oder Arbeitsplätze

Durch die Art der Produkte werden bereits wesentlich die notwendigen Bearbeitungsschritte determiniert. Für alle auftretenden Produkte muss analysiert werden, welche Materialien benötigt werden und in welchen Schritten diese dem Produkt zugefügt werden. Dies umfasst auch anteilig Produktionshilfsmittel, wie zum Beispiel Werkzeuge. Sobald eine Prozessübersicht für die gesamte Produktionsanlage oder das Lager bestimmt ist, sollte für die notwendigen Transporte die in Frage kommende Fördertechnik ermittelt werden. Selbst wenn eine konkrete Auswahl noch nicht möglich ist, beispielsweise weil die zurückzulegenden Distanzen erst mit einem ersten Layoutentwurf feststehen, ist dies notwendig um die notwendigen Transportaufwände abzuschätzen. Dies schließt nicht nur die notwendigen Transporthäufigkeiten ein, sondern soll insbesondere eine kostenmäßige Relation zwischen unterschiedlichen Transportarten abbilden.

Über das Mengengerüst und die identifizierten Arbeitsschritte muss ermittelt werden, welche Kapazitäten für diese Tätigkeiten notwendig sind und wie viele parallele Arbeitsstationen eingerichtet werden müssen. Im Anschluss kann mit Hilfe der Informationen über den Prozess sowie der Anzahl der Arbeitsstationen eine Liste mit den zu platzierenden Einheiten erstellt werden. Auch der jeweilige Platzbedarf lässt sich ermitteln.

Gerade bei Einheiten mit hohem Platzbedarf können die Übergabepunkte von Material erheblichen Einfluss auf das Layout haben. In diesen Fällen ist es sinnvoll diese Einheit genauer zu betrachten, um diese Punkte zu definieren. Diese konkrete Gestaltung der Arbeitsplätze stellt wiederum ein Layoutproblem dar. Darüber hinaus muss in solchen Fällen ein geeignetes Optimierungsverfahren angewendet werden, um unterschiedliche Platzbedarfe zu berücksichtigen und auch die korrekten Ausgangspunkte zur Entfernungsmessung zu verwenden.

Dieser kurze Überblick über die notwendigen Planungsschritte soll zeigen, dass die in ein Optimierungsproblem eingehenden Daten mit erheblichem Aufwand erhoben werden müssen. Zudem sind sie unter Umständen mit erheblichen Unsicherheiten behaftet. Dies sollte ebenso bei der quantitativen Bewertung, die im Rahmen der Optimierung stattfindet, beachtet werden.

Außer bei Produktionsanlagen und Distributionszentren treten Layoutpro-

bleme auch in nahezu jedem anderen Gebäudeentwurf auf, die zum Teil auch dreidimensionalen Charakter haben. In der Literatur wurden explizit beispielsweise Probleme bei der Konzeptionierung von Gefängnisbauten (Krasnow und Parker 1998), der Aufstellungsplanung von Chemieanlagen (Köpcke und Schröder 2004) oder der Aufteilung von Bürokomplexen (Johnson 1982) und Krankenhäusern (Hahn und Krarup 2001) betrachtet.

Exotischere Beispiele von Layoutproblemen treten im weitesten Sinne im Rahmen des Produktdesigns auf. Als Beispiele können dabei genannt werden: die ergonomische Belegung einer Computertastatur, die Bestückung eines Computerchips mit Schaltkreisen (Heragu 2008), die Verlegung von Kabelsträngen eines elektrischen Antriebes (Lengauer und Schäfer 2001), die Aufteilung eines Kreuzfahrtschiffs oder U-Bootes (Jang 2008) oder auch die Platzierung von einzelnen Komponenten innerhalb eines Satelliten (Zhang et al. 2008). Eine Besonderheit der Layoutoptimierung von Schiffen oder Satelliten ist das zusätzliche Einhalten von Gleichgewichtsbedingungen. Bei der Konstruktion von Schiffen muss berücksichtigt werden, dass eine gleichmäßige Verteilung der Masse über das Schiff erfolgt, um eine unerwünschte Neigung des Schiffes zu verhindern. Ähnlich muss bei Satelliten erreicht werden, dass eine stabile Rotation ohne Pendeln möglich ist.

Diese Anwendungsgebiete zeigen, in welchen unterschiedlichen Formen Layoutprobleme in der Praxis auftreten können. Jeder konkrete Fall weist dabei spezifische Restriktionen auf, die bei der Optimierung Beachtung finden müssen. Im Folgenden sollen zunächst existierende Ansätze gegenübergestellt werden, um Gemeinsamkeiten und Unterschiede zu beleuchten. Anschließend werden einige Modelle für den zwei- sowie den dreidimensionalen Fall vorgestellt werden, die in der Literatur bereits untersucht wurden.

### 2.2. Abgrenzung von Verfahren der Layoutplanung

Modelle zur Layoutplanung weisen eine Reihe von Merkmalen auf, anhand derer eine Unterscheidung möglich ist. In Anlehnung an Drira et al. (2007, S. 257) und Scholz (2010, S. 10) sollen hier vier Kriterien vorgestellt werden, die eine Einordnung der in dieser Arbeit diskutierten Modelle erleichtert.

### 2.2.1. Statische und dynamische Probleme

Bei der Mehrzahl der in der Literatur behandelten Modelle handelt es sich um statische Probleme. Statische Probleme gehen von einer einmaligen Planung mit festen Größen aus, bei denen Schwankungen der Materialflüsse durch den Mittelwert abgebildet werden. In vielen praktischen Anwendungsfällen, bei denen man von einem wechselnden Produktspektrum ausgehen kann, trifft dies allerdings nicht zu. Auch eine mit Umbaukosten verbundene Neugestaltung eines Produktionsbereichs auf Basis eines realen Layouts wird durch eine statische Problemformulierung stark relaxiert.

Diese Problematik wurde in den frühen 80er Jahren erkannt und ein entsprechendes Modell von Rosenblatt (1986) erstmals in der Literatur vorgestellt. Dabei wird über alle betrachteten Zustände hinweg berechnet, ob eine Änderung des Layouts sich lohnt oder nicht. Die Standortwechselkosten hängen bei Rosenblatt und auch folgenden Publikationen sowohl von der Art der zu bewegenden Komponenten als auch von der Entfernung ab. Da die Komplexität von dynamischen Problemen gegenüber einer statischen Lösung mit der Anzahl der zeitlichen Zustände exponentiell steigt, können für jeden Zeitpunkt nur verhältnismäßig einfache und in ihren Annahmen restriktive Modelle verwendet werden.

Wichtig anzumerken ist, dass auch in einem dynamischen Modell die zukünftige Entwicklung keiner Stochastik folgen darf, sondern zum heutigen Zeitpunkt als deterministisch anzusehen ist (Rosenblatt 1986).

In der Anwendung interessant sein könnte ein solches dynamisches Layoutproblem beispielsweise auch für den am Institut für Fördertechnik und Logistiksysteme entwickelten Flexförderer (siehe auch Mayer (2009)).<sup>1</sup> Neben der flexiblen Integration in bereits bestehende Systeme, könnte bei einer Neuplanung ein Layout erreicht werden, das einen effizienten Umbau des Systems mit den einzelnen Modulen ermöglicht. Ein solches Layout mit verschiebbaren Flexförderer-Elementen zeigt Abbildung 2.1.

### 2.2.2. Größe und Form der Einheiten

Bei Layoutproblemen wird zumeist von einer regelmäßigen Form der einzelnen Einheiten, üblicherweise Rechtecken, ausgegangen. Daneben gibt es allerdings auch unregelmäßige Formen, die zwar häufig immer noch auf rechtwinkligen

---

<sup>1</sup>Der Flexförderer ist ein modular aufgebautes Fördersystem, bei dem die einzelnen Elemente dezentral agieren und somit keine zentrale Steuereinheit notwendig ist. Damit entfällt auch eine aufwändige Neuprogrammierung des Systems nach einem Umbau.

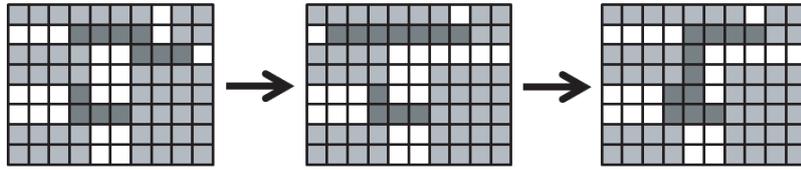


Abbildung 2.1.: Verschieben von Flexförderer-Elementen in einem Layout über 3 Perioden

Formen basieren, aber in manchen Modellen auch völlig beliebige Formen annehmen können. So nehmen Teng et al. (2001) für einen zylinderförmigen Satelliten sowohl kreisförmige als auch rechteckige Grundformen an. Bei dem dreidimensionalen Layoutproblem bei Cagan et al. (1998) können beliebig geformte Objekte verwendet werden, die allerdings über eine Rasterung wieder auf rechtwinklige Formen zurückgeführt werden. Die Objekte sind darüber hinaus frei im Raum drehbar. Die verschiedenen Grundtypen sind in Abbildung 2.2 dargestellt.

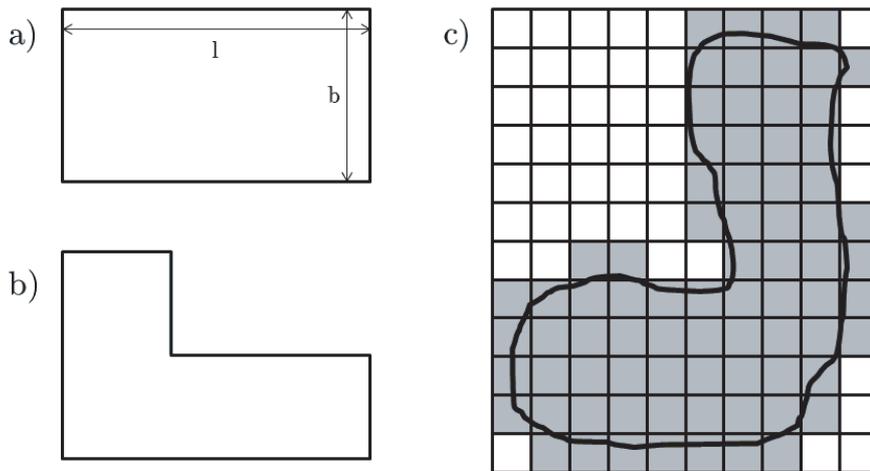


Abbildung 2.2.: Regelmäßige und unregelmäßige Formen der Einheiten

Für freie Formen, wie sie Cagan et al. verwenden (Abbildung 2.2 c)), müssen natürlich feste Vorgaben für die Form getroffen werden; d.h. die Form bleibt zur Laufzeit des Optimierungs-Algorithmus konstant. Bei regelmässi-

gen oder unregelmäßigen Formen nach Typ a) oder b) werden teilweise keine festen Vorgaben bezüglich der Form getroffen. In diesen Fällen existieren lediglich Nebenbedingungen, die die Fläche definieren; die Form ergibt sich bei der Optimierung. Diese Nebenbedingungen legen je nach Modell für eine bestimmte Einheit die Mindest- und Maximalfläche fest. Einige Modelle grenzen die Form dabei noch über das Seitenverhältnis  $a=Länge/Breite$  einer Einheit ein. In den Nebenbedingungen wird dabei jeweils eine untere und obere Grenze festgesetzt (Tam und Li 1991). In manchen Modellen ist das Seitenverhältnis auch fest vorgegeben und nur eine Seitenlänge kann variiert werden (Golmohammadi et al. 2010).

Für die Formgebung spielt auch eine Rolle, ob ein diskretes oder kontinuierliches Maß gewählt wird. Im Falle eines diskreten Maßes können die Formen ebenfalls auch nur aus ganzzahligen Vielfachen des Rasters bestehen. Bei einem sehr engen Raster nähern sich diskrete Formen dem kontinuierlichen Maß an. Im Spezialfall des in Abschnitt 2.3.1 besprochenen quadratischen Zuordnungsproblems besitzen alle Einheiten die gleiche Größe. Erst in Varianten dieses Problems kann diese Restriktion gelockert sein (Scholz 2010, S. 12).

### 2.2.3. Anzahl der berücksichtigten Dimensionen

Bei dem Großteil der in der Literatur zu findenden Modelle handelt es sich um Layoutprobleme in der Ebene. Darüber hinaus sind aber sowohl ein- als auch dreidimensionale Probleme zu finden. Eindimensionale Probleme betrachten typischerweise eine Produktionslinie, die die einzelnen Produkte durchlaufen müssen. Da in diesem Fall alle Produkte die gesamte Linie durchlaufen, kann die Transportdistanz kein entscheidendes Kriterium mehr sein. Für die einzelnen Produkte, für die eine Bearbeitungsreihenfolge vorgegeben ist, kann es allerdings passieren, dass sie in der Linie nochmal zurückgesetzt werden müssen (Abbildung 2.3). Die Minimierung dieser Vorgänge wird in diesen Modellen häufig als Optimierungskriterium verwendet (siehe zum Beispiel Braglia (1996)).

Dreidimensionale Modelle treten in zwei Variationen auf. Zunächst gibt es die tatsächlich räumlichen Layoutprobleme, die auch in dieser Arbeit behandelt werden. Auf der anderen Seite stehen die in Abschnitt 2.2.1 bereits erwähnten dynamischen Modelle, in denen die Zeit die dritte Dimension darstellt. Um Verwechslungen zu vermeiden, werden diese Modelle im Folgenden immer als „dynamische Modelle“ bezeichnet, wohingegen ausschließlich räumliche Layoutprobleme tatsächlich als dreidimensional bezeichnet werden. Eine Abgrenzung dieser dreidimensionalen Ansätze erfolgt in Abschnitt 2.4.

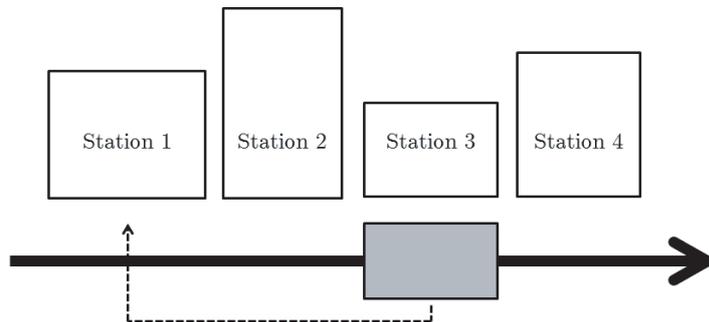


Abbildung 2.3.: Zurücksetzen eines Produkts in einer Produktionslinie

### 2.2.4. Diskrete und kontinuierliche Modelle

Ein weiteres Unterscheidungsmerkmal bei Layoutplanungsmodellen ist die Verwendung eines diskreten oder kontinuierlichen Maßes für die Positionierung der Einheiten. Dieses Kriterium hat auch einen weitreichenden Einfluss auf die anwendbaren Optimierungsalgorithmen. Diskrete Modelle führen dabei zu einer Form des quadratischen Zuordnungsproblems, kontinuierliche Modelle zu einem gemischt-ganzzahligen Problem. Daneben existieren einige Ausnahmen, wie beispielsweise der in Abschnitt 2.3.3 erläuterte graphenbasierte Ansatz. In diesem Fall werden Aussagen über das Layout nur über Nachbarschaftsbeziehungen der Einheiten ausgedrückt; eine konkrete Positionierung wird nicht vorgenommen.

Tabelle 2.1 ordnet einige in der Literatur besprochenen Layoutprobleme anhand der genannten Kriterien ein. Als „frei“ wird in der Tabelle eine Form bezeichnet, die auch unregelmäßig sein kann, „variabel“ meint hingegen, dass die Form aus einer fest vorgegebenen Menge stammen muss. Die Anzahl der Dimensionen bezieht sich dabei ausschließlich auf die Anzahl der räumlichen Dimensionen. Eine Reihe der in der Tabelle genannten Beispiele werden in den folgenden Abschnitten noch näher erläutert, beispielsweise das Verfahren der Flexible Bays oder das Erstellen eines Layout über raumfüllende Kurven (Space Filling Curves).

## 2.2. Abgrenzung von Verfahren der Layoutplanung

Tabelle 2.1.: Auswahl von Layoutproblemen und Abgrenzung anhand relevanter Merkmale

<i>Zeitbezug: statisch (s) oder dynamisch (d), Größe der Einheiten: gleich (g) oder variabel (v), Form der Einheiten: rechteckig (r), variabel (v) oder frei (f), Maß: diskret (d) oder kontinuierlich (k).</i>						
<b>Literatur</b>	<b>Zeitbezug</b>	<b>Größe</b>	<b>Form</b>	<b>Dimensionen</b>	<b>Maß</b>	<b>Optimierungsansatz und Anmerkung</b>
Seppänen und Moore 1970	s	-	-	2	k	Graphentheoretischer Ansatz
Johnson 1982	s	v	f	3	d	SPACECRAFT, baut auf zulässiger Lösung auf, 3 Dimension einzelne Ebenen, wenige Komponenten
Foulds 1983	s	-	-	2	k	Graphentheoretischer Ansatz
Rosenblatt 1986	d	g	r	2	d	basierend auf dem quadratischen Zuordnungsproblem (QZP)
Drezner 1987	s	v	-*	2	k	*Kreise zueinander anordnen
Hassan und Hogg 1987	s	-	-	2	k	Graphentheoretischer Ansatz
Kumara et al. 1988	s	g	r	2	d	Expertensystem
Abdou und Dutta 1990	s	v	r	2	k	Expertensystem mit Auswahl Transportmittel, Rückgriff auf andere Modelle
Bozer et al. 1994	s	v	f	3	d	Space Filling Curve (SFC) auf mehreren Ebenen
Kouvelis et al. 1995	s	g	-	1	d	Single Row Facility Layout Problem (SRFLP)
Tate und Smith 1995	s	v	r	2	k	Seitenverhältnis beschränkt, ähnlich Flexible Bays
Badiru und Arif 1996	s	v	r	2	k	Expertensystem mit Fuzzy-Relationen
Braglia 1996	s	v	r	1	k	Single Row Facility Layout Problem
Meller und Bozer 1996	s	v	f	2	d	Space Filling Curve
Cagan et al. 1998	s	v	f	3	k	Beliebige Formen, beliebige Position (keine 90-Grad Drehungen)
Pesch 1999	s	-	-	2	k	Graphentheoretischer Ansatz
Teng et al. 2001	s	v	f	3	k	Layout in rotierendem Körper, mit Gleichgewichtsbedingungen
Balakrishnan et al. 2003	d	g	r	3	d	basierend auf QZP
Köpcke und Schröder 2004	s	v	r	3	d	Ebenen, Constraint Programming

## 2. Layoutplanung in der Ebene und im Raum

Literatur	Zeitbezug	Größe	Form	Dimensionen	Maß	Optimierungsansatz und Anmerkung
Meng et al. 2004	d	g	r	2	d	basiert auf QZP, Zuordnung Produkte zu Maschinen, Maschinen zu Aufstellungsor-ten
Shayan und Chittilappilly 2004	s	v	r	2	k	Slicing Trees
Asef-Vaziri und Laporte 2005	s	v	r	2	k	Ringförmiges Layout unter Berücksichti-gung von Transportmitteln
Lee et al. 2005	s	v	r	3	k	in Ebenen, Berücksichtigung Gebäude-strukturen (Wände, etc.)
Anjos und Vannelli 2006	s	v	r	3	k	Mehrere Produktionslinien, Zuordnung Zelle und Reihenfolge als weitere Dimen-sionen, Seitenverhältnis
Liu et al. 2006	s	v	r	2	k	Stochastische Flüsse
Bock und Hoberg 2007	s	v	f	2	d	Angriffspunkte mit Wegen im Raster
Tavakkoli-Moghaddam et al. 2007	s	g	r	3	d	Gruppierung der Maschinen in Fertigungs-zellen
Bhowmik 2008	s	v	f	3	d	3. Dimension: Ebenen, mehrstufig; Form nicht beeinflussbar
Hahn et al. 2008	s	g	r	3	d	dreidimensionales QZP
Jang 2008	s	v	f	3	k	Komponenten in Unterseeboot platzieren
Solimanpur und Jafari 2008	s	v	r	2	d	Berücksichtigt Produktionskapazität, red-undante Maschinen und Fertigungsreihen-folgen
Zhang et al. 2008	s	v	f	3	k	Layout in rotierendem Körper, mit Gleich-gewichtsbedingungen
Scholz et al. 2009	s	v	r	2	k	Slicing Trees
Bénabès et al. 2010	s	v	r	2	k	Fokus auf Zugänglichkeit
Furuholmen et al. 2010	s	v	r	3	k	Nachbarschaftsbasiert, Konstruktionsheu-ristik
Golmohammadi et al. 2010	s	v	r	2	k	Flexible Bays
Wong und Komarudin 2010	s	v	r	2	k	Flexible Bays
Bonäs et al. 2007	s	v	r	3	k	Packen mit Flüssen, Zuordnung mehrere Ebenen
Ariaifar et al. 2011	s	g	r	3	d	Gruppierung der Maschinen in Fertigungs-zellen

## 2.2. Abgrenzung von Verfahren der Layoutplanung

Literatur	Zeitbezug	Größe	Form	Dimensionen	Maß	Optimierungsansatz und Anmerkung
Huang et al. 2011	s	-	-	3	k	Positionierung eines Drehkrans bei mehreren Versorgungspunkten mit unterschiedlichem Material
Kulturel-Konak und Konak 2011	s	v	r	2	k	Flexible Bays
Lee et al. 2012	d	g	r	2	d	Stochastische Flüsse, QZP

Auch wenn einige der Modelle ähnliche Eigenschaften aufweisen, kann es sich mitunter doch um sehr verschiedene Ansätze handeln, die ganz unterschiedliche Aspekte der Layoutplanung abdecken. Betrachtet man die genannten raumfüllenden Kurven (unter anderem Bozer et al. (1994) oder Meller und Bozer (1996)), so ist die Form der Einheiten zwar frei, kann jedoch nicht festgelegt werden, sondern ergibt sich aus der Optimierung. Von den Autoren wird dies damit gerechtfertigt, dass bei der Planung von Bürokomplexen nicht unbedingt die Form einer Organisationseinheit entscheidend ist, sondern tatsächlich lediglich eine bestimmte Größe gefordert ist. Dieser Ansatz wird allerdings seit etwa 20 Jahren kaum weiter verfolgt.

Ebenso finden sich in den angegebenen Modellen einige exotischere Beispiele, die zeigen, in welcher unterschiedlichen Formen Layoutprobleme auftreten können. So behandeln Huang et al. (2011) oder Jang (2008) sehr konkrete Spezialfälle, die für das in dieser Arbeit betrachtete Problem zwar keinen Ansatzpunkt liefern, jedoch ebenso eine Form der Layoutoptimierung darstellen.

Darüber hinaus können sich Modelle, die zwar ähnliche Eigenschaften aufweisen, in weiteren Details trotzdem stark unterscheiden. Dies gilt zum Beispiel für Scholz et al. (2009) und Bénabès et al. (2010). Während im ersten Fall die zur Verfügung stehende Fläche durch Schnitte so geteilt wird, dass Anforderungen der einzelnen Einheiten wie die Größe und das gewünschte Seitenverhältnis im Vordergrund stehen, liegt der Fokus bei Bénabès stärker auf der Zugänglichkeit der Einheiten. Auch können sich ähnliche Modelle, gerade wenn ein längerer Zeitraum zwischen dem Erscheinen der Veröffentlichungen liegt, aufgrund der Weiterentwicklung der Rechengeschwindigkeit in der Anzahl der damit zu platzierenden Komponenten stark unterscheiden. In Ab-

hängigkeit vom konkreten Anwendungsfall stellt diese Begrenzung mitunter aber durchaus eine realistische Größe dar.

### 2.2.5. Stochastische Nachfrage

In den bisher erwähnten Fällen wird stets von einer näherungsweise konstanten Nachfrage ausgegangen. Für einen Produktionsbetrieb bedeutet das, dass für den betrachteten Zeithorizont die Nachfrage nach den hergestellten Gütern bekannt ist und sich nicht mehr ändert. Da die Layoutplanung eines Betriebes in der Regel für einen langen Zeitraum erfolgt und dann fixiert ist, ist diese Annahme durchaus kritisch zu sehen. Aufgrund der Komplexität des Grundproblems wird jedoch in den meisten Fällen diese restriktive Annahme in Kauf genommen. Dennoch existieren einige Ansätze die eine stochastische Nachfrage berücksichtigen. Hier wären unter anderem die Arbeiten von Ariafar et al. (2011) und Tavakkoli-Moghaddam et al. (2007) zu nennen. Beide gehen in ihren Modellen von Produktionszellen aus, auf die Maschinen aufgeteilt werden. Innerhalb der Zellen erfolgt dann noch einmal eine Anordnung der Maschinen. Die Materialflusskosten setzen sich somit aus Kosten für den Fluss innerhalb der Zellen und zwischen den Zellen zusammen. Alle Einheiten besitzen die gleiche Größe.

Im Ansatz von Liu et al. (2006) wird hingegen auf die Aufteilung der Einheiten auf einzelne Zellen verzichtet. Die Einheiten besitzen eine feste, aber beliebig große rechteckige Form. Der Ansatz von Lee et al. (2012) ordnet die Einheiten ebenfalls nicht in Zellen an. Im Gegensatz zu den anderen genannten Modellen handelt es sich jedoch um einen dynamischen Ansatz. Die Form und Größe aller Einheiten ist dabei identisch.

In Tabelle 2.2 sind die größten Instanzen aufgezeigt, die mit dem jeweiligen Verfahren jeweils berechnet wurden. Offensichtlich sind diese Modelle nur für kleinere Problemstellungen geeignet. Auch wenn eine stochastische Nachfrage in größeren Instanzen bisher nicht abbildbar ist, sollte bei der Auswahl eines geeigneten Modells doch betrachtet werden, ob die Annahme einer deterministischen Nachfrage annähernd erfüllt ist.

Tabelle 2.2.: Größte berechnete Instanzen verschiedener Modelle mit stochastischer Nachfrage

Modell	Nachfrageverteilung	Anzahl Einheiten	Anzahl Produkte	Zeitperioden
Liu et al. 2006	Normalverteilt	10	4	1
Tavakkoli-Moghaddam et al. 2007	Normalverteilt	12	10	1
Ariafar et al. 2011	Gleichverteilt	7	6	1
Lee et al. 2012	Normalverteilt	12	10	5

## 2.3. Zweidimensionale Modelle

### 2.3.1. Das lineare und das quadratische Zuordnungsproblem

Das lineare Zuordnungsproblem ordnet eine Menge von Objekten  $i$  einer Position  $j$  zu, wobei je nach gewählter Position fixe Kosten in Höhe von  $c_{ij}$  auftreten. Weitere Beziehungen zwischen den einzelnen Objekten oder Positionen existieren darüber hinaus nicht. Damit lässt sich das Problem wie folgt formulieren:

$$\min f(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad (2.1)$$

u.d.N.:

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2.3)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, n, j = 1, \dots, n \quad (2.4)$$

Eine Erweiterung dieses Problems ist das 1957 von Koopmans und Beckmann entwickelte quadratische Zuordnungsproblem (QZP), das seither in zahlreichen Publikationen variiert wurde. Ziel ist es, eine Menge von  $n$  Einheiten einer entsprechenden Zahl von Standorten so zuzuordnen, dass die Transportkosten zwischen den Einheiten minimal werden. Die Transportkosten berechnen sich dabei aus dem Produkt der Transportmenge und der Distanz. Dabei steht  $t_{ij}$  für die von Einheit  $i$  zu Einheit  $j$  transportierte Menge und  $d_{kl}$  für die Distanz zwischen den Standorten  $k$  und  $l$ . Die Variable  $x_{ik}$  nimmt den Wert 1 an, falls Einheit  $i$  Standort  $k$  zugeordnet wurde, andernfalls den Wert 0. Das Problem lässt sich auf mehrere Arten formulieren. Aus der von Koopmans gewählten Notation wird auch deutlich, warum dieses Modell als *quadratische* Zuordnungsproblem bezeichnet wird, da die Variablen  $x_{ik}$  und  $x_{jl}$  in der Zielfunktion  $f(x)$  zusammen einen quadratischen Ausdruck bilden (siehe Gleichung 2.5). Das Modell lässt sich danach wie folgt formulieren:<sup>2</sup>

$$\min f(x) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n t_{ij} \cdot d_{kl} \cdot x_{ik} \cdot x_{jl} \quad (2.5)$$

u.d.N.:

$$\sum_{i=1}^n x_{ik} = 1, \quad k = 1, \dots, n \quad (2.6)$$

$$\sum_{k=1}^n x_{ik} = 1, \quad i = 1, \dots, n \quad (2.7)$$

$$x_{ik} \in \{0, 1\} \quad i = 1, \dots, n, k = 1, \dots, n \quad (2.8)$$

Die ersten beiden Nebenbedingungen stellen dabei sicher, dass einem Standort nur eine Einheit (2.6) bzw. eine Einheit nur einem Standort (2.7) zugeordnet werden darf. Bei  $x_{ik}$  handelt es sich um eine Binärvariable (2.8); das bedeutet, dass die Einheiten nicht auf mehrere Standorte aufgeteilt werden darf. Das Problem erfordert prinzipiell eine identische Anzahl an zu platzierenden

---

<sup>2</sup>Entgegen dieser immer wieder nach Koopmans zitierten Form des quadratischen Zuordnungsproblems, benutzt er selbst zunächst eine etwas abweichende Formulierung. Diese baut auf dem linearen Zuordnungsproblem auf, wobei eine Einheit  $i$ , die dem Standort  $k$  zugeordnet wird, den Gewinn  $a_{ik}$  erwirtschaftet. Dieses Element bleibt in seiner Formulierung auch im quadratischen Zuordnungsproblem erhalten. Aufbauend auf diesen, zum Teil noch relaxierten, Modellen, stellt er spieltheoretische Überlegungen an und stellt Bedingungen auf, unter denen er diese Gewinne vernachlässigt.

Einheiten und in Frage kommenden Standorten. Falls die Zahl an potentiellen Standorten die Zahl der Einheiten übersteigt, kann eine Anpassung durch die Einführung von Dummy-Einheiten vorgenommen werden. Von oder zu diesen Einheiten finden in diesem Fall keine Materialflüsse statt. Die Komplexität des Problems steigt mit der Anzahl der Einheiten sehr schnell an. Dies ist leicht an der Anzahl an möglichen Zuordnungen zu erkennen. Steht  $m$  für die Anzahl an zu platzierenden Einheiten und  $n$  für die Anzahl der Standorte mit  $m < n$ , so ergibt sich die Anzahl an möglichen Permutationen  $p$  zu

$$p = \frac{n!}{(n - m)!} \quad (2.9)$$

Betrachtet man beispielsweise eine Fläche von 10 m Länge und 10 m Breite und möchte darauf lediglich 20 Einheiten platzieren, so ergibt dies bei einer Rastergröße von jeweils 1 m bereits  $1,3 \cdot 10^{39}$  Kombinationsmöglichkeiten. Diese Komplexität zeigt sich auch an der Größe der Instanzen, die bisher optimal gelöst werden konnten. Für das quadratische Zuordnungsproblem existieren eine Reihe von Referenzinstanzen, an denen die Entwicklung sichtbar ist. Zu den bekanntesten, die auch in einer Reihe unterschiedlicher Problemgrößen existieren, zählen die Instanzen von Nugent. Diese 1968 veröffentlichten Instanzen verwenden die Manhattan-Metrik zur Distanzmessung. Die größte Instanz mit 30 Einheiten bzw. Standorten wurde im Jahr 2000 optimal gelöst. Der Fortschritt, der bei der Lösung der Instanzen gemacht wurde, ist beispielsweise in Loiola et al. (2007, S. 670) zusammengefasst (vergleiche dazu auch Abbildung 2.4).

Mittlerweile können auch viele Probleme mit einer Problemgröße zwischen 30 und 40 Einheiten gelöst werden, dabei wird in der Regel auf Computercluster zurückgegriffen, in denen mehrere Hundert Rechner gleichzeitig an der Lösung arbeiten (Hahn und Krarup 2001). Trotz des Fortschritts bei der Vernetzung von Computern, Leistungssteigerungen bei der Rechenleistung einzelner Computer und verbesserten Algorithmen, sind die optimal lösbaren Problemgrößen nach wie vor relativ klein für praktische Anwendungen. In vielen Fällen wird daher auf Heuristiken zurückgegriffen, deren Güte an bereits optimal gelösten Instanzen getestet werden kann.

Ein quadratisches Zuordnungsproblem kann ebenso durch ein äquivalentes, lineares Zuordnungsproblem formuliert werden. Dabei wird eine neue Variable  $y_{ijkl} = x_{ik} \cdot x_{jl}$  eingeführt, die angibt, ob Objekt  $i$  Platz  $k$  zugeordnet ist, während gleichzeitig Objekt  $j$  Platz  $l$  zugeordnet ist. Das lineare Problem ergibt sich dann nach (Lawler 1963) zu

## 2. Layoutplanung in der Ebene und im Raum

---

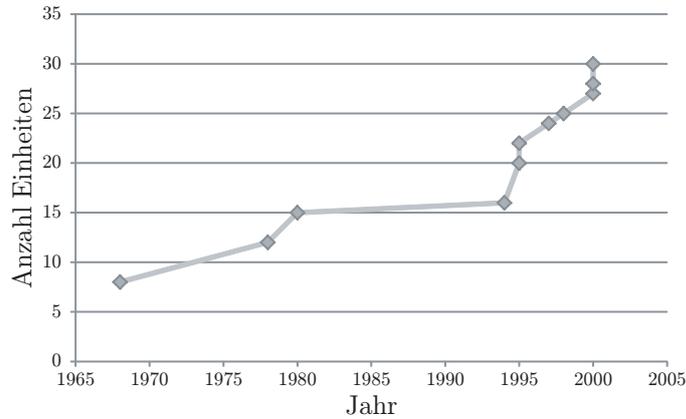


Abbildung 2.4.: Erstmalige optimale Lösung der Instanzen von Nugent

$$\min f(x) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ijkl} \cdot y_{ijkl} \quad (2.10)$$

u.d.N.

$$\sum_{i=1}^n x_{ik} = 1, \quad k = 1, \dots, n \quad (2.11)$$

$$\sum_{k=1}^n x_{ik} = 1, \quad i = 1, \dots, n \quad (2.12)$$

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n y_{ijkl} = n^2 \quad (2.13)$$

$$x_{ik} + x_{jl} - 2y_{ijkl} \geq 0, \quad \forall i, j, k, l \in \{1, \dots, n\} \quad (2.14)$$

$$x_{ik}, y_{ijkl} \in \{0, 1\} \quad i, j, k, l = 1, \dots, n, \quad (2.15)$$

Diese Formulierung ist eine Möglichkeit, eine untere Schranke nach Abschnitt 4.2.1 zu bestimmen.

Neben dem hier vorgestellten (zweidimensionalen) quadratischen Zuordnungsproblem gibt es auch einige Veröffentlichungen zu drei- oder mehrdimensionalen quadratischen Zuordnungsproblemen. Dabei beschreibt der Begriff „Di-

mension“ nicht mehrere räumliche Dimensionen (diese werden als eine Dimension abgebildet), sondern die Zuordnung zu verschiedenen Zeitpunkten, oder von mehreren Paketen in eine Reihe von Containern. Literatur zum dreidimensionalen Problem ist beispielweise bei Hahn et al. (2008) zu finden, eine Formulierung für das n-dimensionale Problem ist in Bonäs et al. (2007) zu finden. Insgesamt existiert für diesen Bereich relativ wenig Literatur, da die Anwendungsgebiete eher begrenzt sind.

### 2.3.2. Space Filling Curves

Space Filling Curves eignen sich für die Layoutplanung, falls lediglich der Flächenbedarf einzelner Einheiten, nicht jedoch eine konkrete Form vorgegeben ist. Unter einer Space Filling Curve versteht man dabei eine Linie, die eine zweidimensionale Fläche oder allgemein einen n-dimensionalen Hyperwürfel durchläuft. Mathematisch ist eine Space-Filling Curve nach Sagan (1994, S. 5) folgendermaßen definiert:

**Definition 1 (Space Filling Curve)** *Das Bild einer stetigen Abbildung  $f : I \rightarrow \mathbb{R}^n (n > 1)$  wird genau dann als raumfüllend bezeichnet, falls  $I \subset \mathbb{R}$  eine kompakte Menge ist und  $f(I)$  ein Jordan-Maß (Fläche, Raum,...) größer 0 besitzt.*

Eine Abbildung nach dieser Definition ist surjektiv, aber nicht injektiv.<sup>3</sup> Da die Injektivität bedeutet, dass Elemente der Zielmenge sich eindeutig einem Element der Urbmenge zuordnen lassen, kann eine solche Kurve sich nicht selbst kreuzen.

Den bekannten Space Filling Curves liegt ein Konstruktionsmuster zugrunde, das in beliebig vielen Stufen verfeinert werden kann (vgl. auch Abbildung 2.5). Durch unendliches Wiederholen des Musters füllt die Kurve letztendlich einen definierten Bereich nach Definition 1 vollständig aus.

Es existiert eine Vielzahl unterschiedlicher Space Filling Curves im zwei- sowie dreidimensionalen Raum, die für die unterschiedlichsten Anwendungsgebiete genutzt werden. Eine konkrete Auswahl kann nur in Abhängigkeit von den Anforderungen des Anwendungsfalls getroffen werden. Für den Bereich der Layoutplanung ist es bedeutsam, dass durch die Space Filling Curve zusammenhängende Flächen entstehen. Daraus folgt, dass Funktionseinheiten, egal in welcher Reihenfolge sie entlang der Kurve angeordnet werden, nicht zerteilt werden können. In Abbildung 2.6 ist leicht ersichtlich, dass beispielsweise

<sup>3</sup>Der Beweis, dass die Abbildung nicht injektiv ist, wurde von E. Netto 1879 erbracht.



eine Z-Anordnung im Gegensatz zu einer Hilbertkurve nicht unbedingt zu zusammenhängenden Funktionseinheiten führen muss. Demnach ist eine solche Z-Anordnung nicht geeignet.

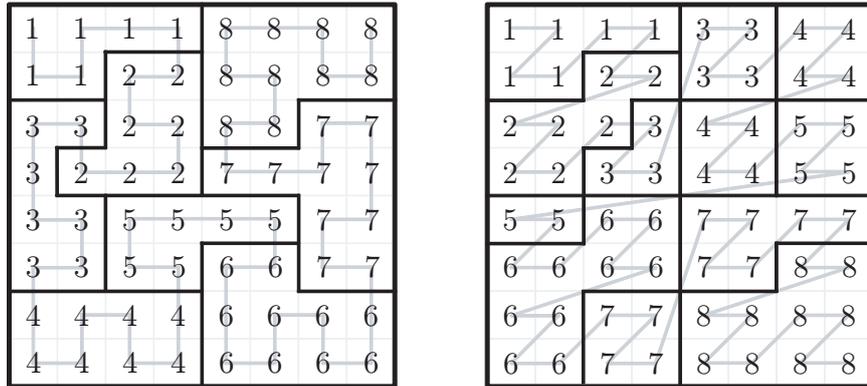


Abbildung 2.6.: Anordnung von 8 Funktionseinheiten unterschiedlicher Größe entlang einer Hilbertkurve und einer Z-Kurve 3. Ordnung

Die typischen Konstruktionsmuster beruhen auf einer quadratischen Fläche und füllen diese mit einem symmetrischen Muster vollständig aus. Für die Anwendung im Bereich der Layoutplanung muss zudem jedoch berücksichtigt werden, dass die zur Verfügung stehende Fläche oder der zur Verfügung stehende Raum nicht die gleiche Größe in allen Dimensionen besitzen muss. Darüber hinaus können Hindernisse den Verlauf der Kurve beeinflussen und müssen natürlich auch nicht symmetrisch verteilt sein. Die Konstruktion geeigneter Kurven kann dabei manuell oder mit automatisierten Verfahren erfolgen. Neben der Auswahl einer geeigneten Space Filling Curve muss bei diesem Ansatz auch die Reihenfolge beachtet werden, in der die Einheiten angeordnet werden. Diese Reihenfolge muss neben den Materialflüssen zwischen den Einheiten auch die Charakteristik der Kurve berücksichtigen.

Das Feld der Layoutplanung ist für die Anwendung von Space Filling Curves eher ein exotischeres Beispiel, zu dem auch nur wenig aktuelle Literatur existiert.

### 2.3.3. Graphenbasierte Modelle

Anfang der 70er Jahre entstand der Ansatz Layoutprobleme über Graphen zu lösen. Dieses Verfahren trifft zunächst keine Aussagen über die konkrete Position einer Einheit, sondern bewertet nur die Vorteile unterschiedlicher Nachbarschaftsbeziehungen. Die Umsetzung des daraus resultierenden Graphen in ein konkretes Layout ist eine weitere Schwierigkeit in diesem Ansatz. Formulierungen dieses Modells finden sich beispielsweise in Seppänen und Moore (1970), Foulds und Robinson (1976) oder Foulds (1983). Dabei existieren  $n$  Knoten für die zu platzierenden Einheiten, die in der Menge  $V$  enthalten sind. Diese Menge wird in der Regel noch um einen weiteren Knoten, der die Außenwelt darstellt, erweitert. Eine Kante im Graph bedeutet, dass die beiden verbundenen Einheiten im Layout nebeneinander angeordnet werden müssen. Die Mengen  $N$ ,  $F$  und  $E$  enthalten alle theoretisch möglichen Verbindungen zwischen zwei Knoten  $i$  und  $j$ . Dabei gilt für die Knotenpaare in  $N$ , dass sie benachbart sein müssen und für die Paare in  $F$ , dass sie nicht benachbart sein dürfen. Alle anderen Paare sind Elemente der Menge  $E$ . Der Parameter  $w_{ij}$  gibt für jedes Paar von Einheiten an, welcher Nutzen durch eine benachbarte Platzierung erzielt wird. Ob zwei Einheiten benachbart sind, lässt sich an der Variablen  $x_{ij}$  erkennen, die in diesem Fall den Wert 1 annimmt. Damit lässt sich das Problem folgendermaßen formulieren:

$$\max f(x) = \sum_{\{i,j\} \in E} w_{ij}x_{ij} \quad (2.16)$$

u.d.N.:

$$(V, E' \cup N) \text{ ist ein planarer Graph} \quad (2.17)$$

$$x_{ij} = 1, \quad \{i, j\} \in N \quad (2.18)$$

$$x_{ij} = 0, \quad \{i, j\} \in F \quad (2.19)$$

$$w_{ij} \in \mathbb{R}^+ \quad (2.20)$$

$$x_{i,j} \in \{0, 1\} \quad (2.21)$$

Demnach soll gemäß Gleichung 2.16 der Gesamtnutzen, der aus allen benachbarten Einheiten resultiert, maximiert werden. Dabei müssen die geforderten oder zu vermeidenden Nachbarschaften berücksichtigt werden (Gleichung 2.18 und 2.19). Ohne Nebenbedingung 2.17 wäre es jetzt optimal, alle Verbindungen, die nicht in Menge  $F$  enthalten sind, zuzulassen, um den damit verbun-

denen Nutzen zu realisieren. Der entstehende Graph ließe sich anschließend aber nicht mehr in einem Layout realisieren. Unter einem *planaren* Graphen versteht man dabei einen Graphen, der sich ohne eine Überschneidung von Kanten in der Ebene zeichnen lässt. Der Spezialfall eines *maximal planaren* Graphen ist gegeben, falls zu einem planaren Graphen keine Kante mehr hinzugefügt werden kann, ohne dass eine Überschneidung entsteht. Da davon ausgegangen wird, dass alle Nachbarschaften positiven Nutzen bringen, muss eine optimale Lösung folglich immer ein maximal planarer Graph sein. Ein solcher Graph mit einem passendem Layout ist in Abbildung 2.7 zu sehen.

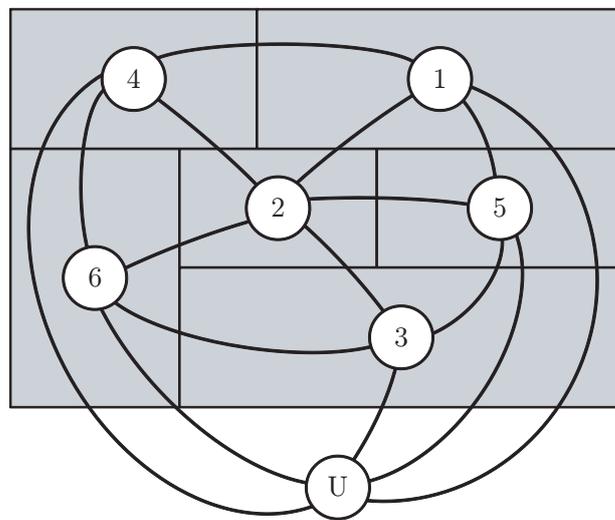


Abbildung 2.7.: Maximal planarer Graph mit blockförmigen Layout von 6 Funktionseinheiten und Umgebungsknoten  $U$

Prinzipiell sind für einen nach diesem Modell bestimmten optimalen Graphen eine Vielzahl von Layoutkonfigurationen möglich. Bei der praktischen Anwendung besteht zudem das Problem, dass die Größe der Funktionseinheiten bei der Ermittlung des Graphen nicht berücksichtigt wird. Problematisch ist dies insbesondere in Fällen, in denen der Flächenbedarf der Einheiten mit dem Wert der mit ihnen verknüpften Nachbarschaftsbeziehungen negativ korreliert. Beispielsweise könnten einer kleinen Einheit viele Nachbarschaftsbeziehungen zugeordnet werden. Die Erstellung des Layouts würde sich dann schwierig gestalten, da die Außenseite der Einheit nur wenig Kontaktfläche für angrenzende Einheiten bietet. Es kann ebenso nicht davon ausgegangen

werden, dass ein maximal planarer Graph sich in jedem Fall durch ein Layout mit regelmäßigen Flächen darstellen lässt.

Ein weiteres Problem ist, dass über das Bestehen einer Nachbarschaft hinaus keine weitere Aussage über die relative Position getroffen wird. Einheiten, die nicht benachbart sind, aber trotzdem im Layout sehr nah zueinander stehen, erzielen im ersten Schritt der Optimierung keinen Nutzen. Da in der Praxis die Transportdistanz aber auch in diesen Fällen ein relevantes Kriterium darstellt, ist neben der fehlenden Berücksichtigung des Flächenbedarfs auch dieser Aspekt kritisch zu sehen.

### 2.3.4. Stetige Modelle

Stetige Layoutplanungsmodelle lassen sich in zwei Gruppen unterteilen. Auf der einen Seite stehen dabei Schnittmodelle, in denen die zur Verfügung stehende Fläche solange durch horizontale und vertikale Schnitte geteilt wird, bis jede Einheit platziert werden kann. Zu diesen Verfahren zählen beispielsweise das 1983 vorgestellte Konzept der Slicing Trees (Stockmeyer 1983) oder das Erstellen von streifenförmigen Layouts durch Flexible Bays, das Anfang der 90er Jahre entwickelt wurde (Tate und Smith 1995). Dieses Vorgehen lässt sich ebenso als Entscheidungsbaum darstellen. Voraussetzung für diese Art von Modellen ist eine gewisse Freiheit bei der Gestaltung der einzelnen Komponenten. Das bedeutet, dass der Flächenbedarf und das Seitenverhältnis einer Einheit im Idealfall in gewissen Grenzen variabel sind. Zwar lassen sich auch Einheiten mit festen Dimensionen integrieren, allerdings bekommen diese eine Fläche zugewiesen, die in der Regel größer als die Einheit selbst ist. Daher muss anschließend wiederum betrachtet werden, wie die Einheiten genau innerhalb des ihnen zugewiesenen Raumes platziert werden. Ebenso erfordert die Rotation von Einheiten einen größeren Rechenaufwand. Die Reihenfolge der Schnitte mit dem entsprechenden Entscheidungsbaum und das Erzeugen des Layouts ist in Abbildung 2.8 dargestellt.

Im Gegensatz dazu wird im Falle der Flexible Bays zunächst die gesamte Fläche in Streifen unterteilt. Anschließend werden den Einheiten je nach Flächenbedarf Abschnitte innerhalb dieser Streifen zugewiesen. Dieses Vorgehen ist in Abbildung 2.9 dargestellt. Aktuellere Ansätze zu den Verfahren der Slicing Trees und Flexible Bays finden sich beispielsweise bei Shayan und Chittilappilly (2004) und Scholz et al. (2009), beziehungsweise bei Wong und Komarudin (2010) oder Kulturel-Konak und Konak (2011).

Auf der anderen Seite steht eine große Zahl von gemischt-ganzzahligen Problemstellungen, bei denen Einheiten so platziert werden müssen, dass Über-

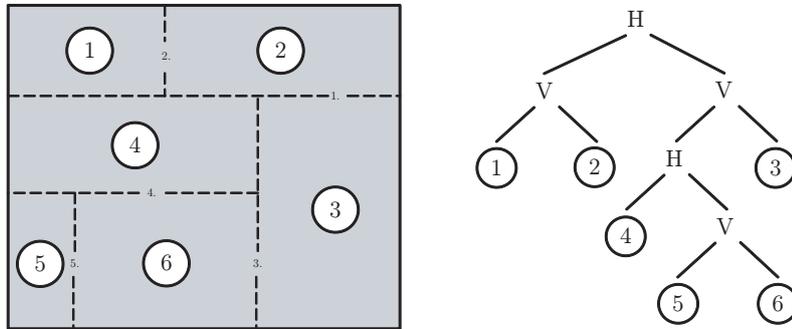


Abbildung 2.8.: Layouterzeugung mit Slicing Trees und zugehöriger Entscheidungsbaum

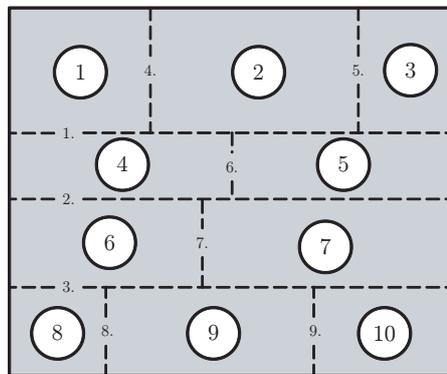


Abbildung 2.9.: Layouterzeugung mit Hilfe von Flexible Bays

schnidungen vermieden werden können. Diese lassen sich häufig auf ähnliche Weise auch als dreidimensionales Problem formulieren, wenn auch die Berechnung wesentlich komplexer ist. Prinzipiell werden dabei verschieden große Rechtecke in einem begrenzten Koordinatensystem so angeordnet, dass Überlappungen vermieden werden und die Materialflüsse minimiert werden. Je nach Bezugspunkt (beispielsweise Mitte oder Ecke eines Rechtecks) oder mit der Möglichkeit der Drehung variieren die Modelle, weisen jedoch grundsätzlich Parallelen auf. In der Art der Formulierung ähneln diese Modelle auch der Klasse der Packprobleme, wenn auch in den beiden Fällen unterschiedliche Ziele verfolgt werden. So lässt sich das Problem beispielsweise in Anlehnung an das Packproblem bei Chen et al. (1995) und Verwendung einer rechtwinkligen Distanzmessung folgendermaßen formulieren:

$$\min f(x) = \sum_{i=1}^n \sum_{k=1}^n t_{ik} \cdot d_{ik} \quad (2.22)$$

mit

$$d_{ik} = \left[ |x_i + 0,5 \cdot (lx_i \cdot l_i + (1 - lx_i) \cdot b_i) - x_k - 0,5 \cdot (lx_k \cdot l_k + (1 - lx_k) \cdot b_k)| \right. \\ \left. + |y_i + 0,5 \cdot (lx_i \cdot b_i + (1 - lx_i) \cdot l_i) - y_k - 0,5 \cdot (lx_k \cdot b_k + (1 - lx_k) \cdot l_k)| \right]$$

u.d.N.:

$$x_i + l_i \cdot lx_i + b_i \cdot (1 - lx_i) \leq x_k + (1 - links_{ik}) \cdot M \quad \forall i, k \in S, i < k \quad (2.23)$$

$$x_k + l_k \cdot lx_k + b_k \cdot (1 - lx_k) \leq x_i + (1 - rechts_{ik}) \cdot M \quad \forall i, k \in S, i < k \quad (2.24)$$

$$y_i + b_i \cdot lx_i + l_i \cdot (1 - lx_i) \leq x_k + (1 - hinter_{ik}) \cdot M \quad \forall i, k \in S, i < k \quad (2.25)$$

$$y_k + b_k \cdot lx_k + l_k \cdot (1 - lx_k) \leq x_i + (1 - vor_{ik}) \cdot M \quad \forall i, k \in S, i < k \quad (2.26)$$

$$x_i + l_i \cdot lx_i + b_i \cdot (1 - lx_i) \leq L \quad \forall i \in S \quad (2.27)$$

$$y_i + l_i \cdot (1 - lx_i) + b_i \cdot lx_i \leq B \quad \forall i \in S \quad (2.28)$$

$$links_{ik} + rechts_{ik} + hinter_{ik} + vor_{ik} \geq 1 \quad \forall i, k \in S, i < k \quad (2.29)$$

$$x_i, y_i, z_i \geq 0 \quad (2.30)$$

$$links_{ik}, rechts_{ik}, vor_{ik}, hinter_{ik} \in \{0, 1\} \quad (2.31)$$

$$lx_i \in \{0, 1\} \quad (2.32)$$

Dabei bezeichnen  $L$  und  $B$  die Länge und Breite der zur Verfügung stehenden Fläche,  $l_i$  und  $b_i$  stehen für die entsprechenden Maße einer Einheit  $i$ . Der linke untere Punkt einer Einheit wird durch die Variablen  $x_i$  und  $y_i$  im Koordinatensystem festgelegt. Ist die Längsseite der Einheit  $i$  entlang der x-Achse

orientiert, nimmt die Variable  $lx_i$  den Wert 1 an, andernfalls 0. Darüberhinaus existieren Binärvariablen zur Beschreibung der relativen Lage zueinander ( $links_{ik}$ ,  $rechts_{ik}$ , etc.). Der Parameter  $M$  ist zur Linearisierung der Nebenbedingungen notwendig und entspricht einer hinreichend großen natürlichen Zahl. Die Menge  $S$  umfasst alle anzuordnenden Funktionseinheiten.

Die Zielfunktion  $f(x)$  in Gleichung 2.22 misst dabei unter Berücksichtigung der Orientierung alle zwischen den Einheiten auftretenden Entfernungen und multipliziert diese mit den jeweiligen Transportintensitäten  $t_{ik}$ . Ausgangspunkt für die Messung ist dabei immer der Mittelpunkt jeder Einheit. Die Nebenbedingungen 2.23 bis 2.26 betrachten paarweise die Position der Einheiten und verhindern ein Überlappen. Ebenso muss darauf geachtet werden, dass alle Einheiten innerhalb der zulässigen Fläche angeordnet werden, das geschieht in den Bedingungen 2.27 und 2.28. Nebenbedingung 2.29 stellt sicher, dass paarweise mindestens ein Test auf Überlappen durchgeführt wird, jedoch nicht mehr als notwendig. Die folgenden Nebenbedingungen geben den zulässigen Wertebereich der Variablen an.

Stetige Modelle mit diesen oder ähnlichen Formulierungen sind gegenüber Ansätzen wie dem in Abschnitt 2.3.1 vorgestellten quadratischen Zuordnungsproblem sehr viel flexibler. Darüber hinaus könnte ebenso auf einen fixen Punkt als Start und Ziel der Materialflüsse zugunsten eines oder mehrerer Punkte mit frei wählbaren Koordinaten innerhalb der Fläche verzichtet werden. Sei  $T_i$  die Menge aller Unterpunkte einer Einheit  $i$ , so nimmt die Zielfunktion die in Gleichung 2.33 gegebene Form an.

$$\min f(x) = \sum_{i=1}^n \sum_{k=1}^n \sum_{s=1}^{|T_i|} \sum_{t=1}^{|T_k|} t_{i,k,s,t} \cdot d_{i,k,s,t} \quad (2.33)$$

mit

$$\begin{aligned} d_{i,k,s,t} = & \quad |x_i + lx_i \cdot x_{i,s} \cdot (2 \cdot pn_i - 1) + (1 - lx_i) \cdot y_{i,s} \cdot (1 - 2 \cdot pn_i) - \\ & \quad x_k - lx_k \cdot x_{k,t} \cdot (2 \cdot pn_k - 1) - (1 - lx_k) \cdot y_{k,t} \cdot (1 - 2 \cdot pn_k)| + \\ & \quad |y_i + lx_i \cdot y_{i,s} \cdot (2 \cdot pn_i - 1) + (1 - lx_i) \cdot y_{i,s} \cdot (2 \cdot pn_i - 1) - \\ & \quad y_k - lx_k \cdot y_{k,t} \cdot (2 \cdot pn_k - 1) - (1 - lx_k) \cdot y_{k,t} \cdot (2 \cdot pn_k - 1)| \end{aligned}$$

Jede Einheit besitzt dabei ein eigenes Koordinatensystem, innerhalb dessen die Position der Unterpunkte durch  $x_{i,s}$  und  $y_{i,s}$  gegeben ist. Neben den zwei Orientierungen einer Einheit, die im ersten Ansatz möglich waren (Längsseite

## 2. Layoutplanung in der Ebene und im Raum

entlang der x-Achse oder entlang der y-Achse), spielt es jetzt ebenso eine Rolle, ob die Längsseite in positiver oder negativer Richtung einer Achse verläuft. Diese Fälle sind zusätzlich zu berücksichtigen, da eine Einheit unter Berücksichtigung von Unterpunkten nur noch in Ausnahmefällen (nämlich einem einzigen Unterpunkt in der Mitte der Einheit) symmetrisch ist. Hierfür wird die Binärvariable  $pn_i$  verwendet, die angibt ob die x-Achse einer Einheit den gleichen Richtungssinn wie eine der Hauptachsen hat ( $pn_i=1$ ) oder nicht ( $pn_i=0$ ). Durch die Kombination der Binärvariablen  $lx_i$  und  $pn_i$  ist es möglich alle vier auftretenden Fälle abzubilden. Dies ist ebenso in Abbildung 2.10 dargestellt.

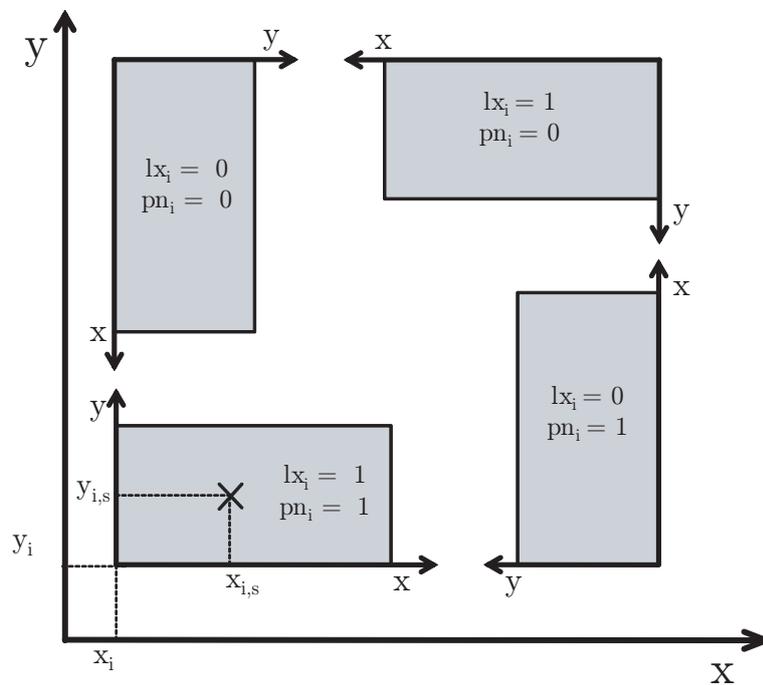


Abbildung 2.10.: Mögliche Positionierungen der Einheit  $i$  innerhalb des Layouts

Sinnvoll kann ein solches Modell sein, wenn sich ein Layout durch ein stufenweises Vorgehen ergibt, in dem beispielsweise zunächst ein einzelner Arbeitsplatz, anschließend die Anordnung von Arbeitsplätzen innerhalb einer Fertigungsinsel, und zuletzt das Gesamtlayout bestimmt werden. Die Annah-

me, dass z.B. alle Materialflüsse von der Mitte einer Fertigungsinsel ausgehen oder dort enden, kann dabei das Ergebnis erheblich verfälschen.

### 2.3.5. Layoutplanung unter Berücksichtigung fördertechnischer Anlagen

Im Bereich der Produktion werden einzelne Fertigungszellen häufig über Fördertechnik in Form von Stetig- oder Unstetigförderern miteinander verbunden. Die Art der technischen Ausstattung kann dabei erheblichen Einfluss auf die Gestaltung des Layouts haben, da gewisse Randbedingungen durch die Technik vorgegeben werden. So ist beispielsweise bei der Gestaltung eines Layouts, das einen Hängeförderer zum Transport von Werkstücken verwendet, darauf zu achten, dass eine Anordnung der Zellen möglichst so erfolgt, dass Kreuzungen der Schienen vermieden werden. Dies würde folglich zu einem ringförmigen Layout führen, das über den Hängeförderer verbunden wird. Ebenso sollte bei Verwendung von Gabelstaplern oder fahrerlosen Transportsystemen darauf geachtet werden, dass ausreichend breite Fahrstraßen eingeplant werden. Auch in diesem Fall wird es aus Durchsatz- bzw. Sicherheitsgründen sinnvoll sein, unnötige Kreuzungen der Wege zu vermeiden.

Im Bereich der Fertigung durchläuft ein Produkt in den meisten Fällen eine festgelegte Abfolge von Arbeitsschritten, die in innerhalb einer verhältnismäßig stabilen Durchlaufzeit bewältigt werden. Je weniger diese Schrittfolge für unterschiedliche Produkte variiert, desto eher werden bestimmte Grundmuster, wie ein linien- oder kreisförmiger Aufbau zu beobachten sein. Von diesen lassen sich nach Ganesharajah et al. (1998) vier Typen identifizieren:<sup>4</sup>

- Linienförmige Anordnung
- Ringförmige Anordnung
- Leiterförmige Anordnung
- Vernetzte Transportsysteme

Eine Reihe von Layoutplanungsmodellen versucht diese Besonderheit zu berücksichtigen und eine technikgerechtes Layout zu generieren. In der Literatur sind insbesondere Ansätze für linienförmige (siehe zum Beispiel Kouvelis et al. (1995) oder Braglia (1996)) und kreisförmige Muster (unter anderem Asef-Vaziri und Laporte (2005)) zu finden.

---

<sup>4</sup>Ganesharajah bezieht sich in seiner Veröffentlichung speziell auf fahrerlose Transportsysteme. Allerdings lassen sich diese Muster auch bei Anwendung anderer Transportarten beobachten.

## 2. Layoutplanung in der Ebene und im Raum

---

Zur Erstellung linearer Anordnungen entlang einer Förderstrecke wird in der Regel versucht Bewegungen entgegen der Flussrichtung zu minimieren. Die Kosten für den Fluss entlang des Stroms sind für alle Produkte identisch, selbst wenn sie nicht die gleiche Zahl an Bearbeitungsschritten aufweisen. Entsprechend kann die Formulierung wie z.B. bei Kouvelis et al. (1995) erfolgen:

$$\min_{\alpha \in \pi} f(\alpha) = \sum_{i=1}^n \sum_{j=1}^n t_{ij} \cdot d(\alpha(i), \alpha(j)) \cdot 1_{\alpha(i), \alpha(j)} \quad (2.34)$$

mit

$$1_{\alpha(i), \alpha(j)} = \begin{cases} 1, & \text{falls } \alpha(i) > \alpha(j) \\ 0, & \text{sonst} \end{cases} \quad (2.35)$$

Dabei steht  $\alpha$  für einen Vektor mit  $n$  Einträgen im Wertebereich von  $1, 2, \dots, n$ . Der  $i$ -te Eintrag des Vektors gibt die Position von Einheit  $i$  innerhalb der Linie an. Wie in den vorigen Modellen gibt  $t_{ij}$  den Fluss von Einheit  $i$  zu Einheit  $j$  und  $d(\alpha(i), \alpha(j))$  die Distanz zwischen diesen Einheiten an. Die Indikatorfunktion  $1_{\alpha(i), \alpha(j)}$  stellt sicher, dass dabei nur Transporte entgegen der Flussrichtung berücksichtigt werden. Dies impliziert, dass der Fluss von der ersten zur  $n$ -ten Position verläuft. Ziel ist es nun, eine Permutation  $\pi$  der Werte  $1, 2, \dots, n$  zu finden, die die Kosten minimiert. Demnach handelt es sich um eine alternative Formulierung eines quadratischen Zuordnungsproblems. Das bedeutet ebenso, dass keine unterschiedlichen Größen der anzuordnenden Einheiten berücksichtigt werden können.

Ein typisches Beispiel für kreis- oder ringförmige Layoutmuster ist neben der Produktion auch die Einrichtung von Baustellen mit einem Drehkran (Chau und Anson 2002). Dabei sollen Materialien so bereitgestellt werden, dass sie auf möglichst kurzem Weg an die richtige Position gebracht werden können. Entgegen dem Fall der linienförmigen oder ringförmigen Anordnung von Maschinen in der Produktion, ist dabei die Flussrichtung kein Kriterium. Während Chau und Anson Positionen auf einer kreisförmigen Ebene bestimmen, werden bei Huang et al. (2011) alle Ziel- und Bereitstellpositionen in einem dreidimensionalen Raum angenommen. Für die Position des Drehkrans wird eine Auswahl aus mehreren in Frage kommenden Alternativen getroffen.

## 2.4. Dreidimensionale Modelle

Im Falle von dreidimensionalen Ansätzen lassen sich nicht mehr so einfach bestimmte Formulierungsvarianten identifizieren, da viele der in der Literatur genannten Methoden für sehr spezielle Anwendungsfälle ausgelegt sind. Eine Reihe von Modellen stellt dabei auch eine Erweiterung von zweidimensionalen Problemen dar. Die Komplexität für den dreidimensionalen Fall steigt dabei allerdings deutlich. Dies ist beispielsweise bei dem in Abschnitt 2.3.1 behandelten quadratischen Zuordnungsproblem der Fall, das allein aus der Formulierung nicht auf eine bestimmte Anzahl an Dimensionen beschränkt ist. Bei diesem Modell steckt die Anzahl an Dimensionen implizit in der verwendeten Entfernungsmatrix. Ebenso lässt sich das Konzept der Space Filling Curves prinzipiell auch in den dreidimensionalen Raum übertragen. Auch wenn die notwendigen theoretischen Grundlagen in Form entsprechender Space Filling Curves vorhanden sind, findet sich in der Literatur kein Ansatz, der ein dreidimensionales Layout auf diese Weise erzeugt. Der Grund dafür dürfte die Tatsache sein, dass sich die entstehende Form der Einheiten nicht steuern lässt und für völlig variabel geformte dreidimensionale Objekte keine praktischen Anwendungsfälle existieren.

Ebenso kann das durch die Gleichungen 2.22 bis 2.29 definierte stetige Layoutplanungsmodell leicht durch zusätzliche Variablen und Parameter auf dreidimensionale Anwendungen erweitert werden. Dabei sollte jedoch bedacht werden, ob tatsächlich sämtliche Rotationsachsen zugelassen werden sollen. Beispielsweise wird eine Rotation um die x- oder y-Achse und damit die Möglichkeit, eine Einheit „auf den Kopf zu drehen“ im Allgemeinen in der Praxis nicht wünschenswert sein.

Andere in der Literatur zu findende Ansätze sollen im Folgenden in die Kategorien

- ebenenbasierte Modelle
- „echt“ dreidimensionale Modelle
- und Modelle mit Gleichgewichtsbedingungen

unterteilt werden. Dabei kann ein Modell zwar durchaus Aspekte aus mehreren Kategorien enthalten. In den meisten Fällen wird jedoch ein bestimmtes Merkmal vorherrschend sein. Einen guten Überblick über einige Lösungsverfahren und unterschiedliche Ansätze geben Cagan et al. (2002).

### 2.4.1. Ebenenbasierte Modelle

Eine Möglichkeit, die Problemkomplexität eines dreidimensionalen Ansatzes zu reduzieren, ist das Verwenden von Ebenen. Betrachtet man beispielsweise die Aufteilung eines Bürokomplexes in verschiedene Organisationseinheiten wie zum Beispiel bei Johnson (1982), Bland (1999) oder Bhowmik (2008), ist ein solches Vorgehen auch durchaus realitätsnah.

Die Lösung von Problemen, die diesem Ansatz folgen, kann in der Regel über Methoden erfolgen, wie sie in Abschnitt 2.3 vorgestellt wurden. Bei Johnson (1982) und Bozer et al. (1994) wird dabei das dreidimensionale Problem an sich nur bedingt betrachtet. Aufbauend auf einer Ausgangslösung, wird durch das Vertauschen von Einheiten eine Verbesserung erreicht. Das Layout der Ebenen wird bei Bozer et al. beispielsweise über eine Space Filling Curve bestimmt.

Ein anderer Ansatz, wie von Bhowmik (2008) verfolgt, ist die Problemlösung in zwei Stufen. Zunächst wird dabei das vertikale Problem separat betrachtet und im Anschluss jedes der  $m$  Optimierungsprobleme für sich gelöst. Die dabei notwendige Unabhängigkeit der  $m$  einzelnen Probleme ist durch die Annahme gegeben, dass die Ebenen über einen Punkt zusammenhängen. Das heißt, es muss nicht berücksichtigt werden, dass es gegebenenfalls unterschiedlich lange Wege von einer Organisationseinheit zu einer anderen gibt, von denen der kürzeste gewählt werden muss. Die Lösung erfolgt über eine Clustering der Einheiten, die hohe Materialflüsse aufweisen, und einen anschließenden Branch-and-Bound-Algorithmus.

Das von Bhowmik (2008) verwendete vertikale Modell ist wie folgt definiert:

$$\min f(x) = \sum_{k=1}^m \sum_{l=1}^m \sum_{i=1}^n \sum_{j=1}^n t_{ij} \cdot d_{kl} \cdot x_{ik} \cdot x_{jl} \quad (2.36)$$

u.d.N.:

$$\sum_{i=1}^n x_{ik} \cdot a_k \leq A_k, \quad k = 1, \dots, m \quad (2.37)$$

$$\sum_{k=1}^m x_{ik} = 1, \quad i = 1, \dots, n \quad (2.38)$$

$$x_{ik} \in \{0, 1\} \quad i = 1, \dots, n, k = 1, \dots, m \quad (2.39)$$

Offensichtlich ist die Formulierung sehr nahe an dem in Abschnitt 2.3.1 vorgestellten quadratischen Zuordnungsproblem. Bedingt durch die unterschiedliche Anzahl von Einheiten und Ebenen ist es jedoch neben einer leichten Abweichung in der Zielfunktion auch notwendig, den verfügbaren Platz pro Ebene zu berücksichtigen. Nebenbedingung 2.37 stellt daher sicher, dass die aufsummierten Flächen  $a_k$  der einer Ebene zugeordneten Einheiten deren maximale Fläche  $A_k$  nicht übersteigen.

Auf der anderen Seite können ebenenbasierte Modelle auch für das Design eines bestimmten Produkts sinnvoll sein. In diesen Fällen stellen die Ebenen Ansatzpunkte zur Montage der einzelnen Einheiten dar. Dies wird beispielsweise auch bei dem Design von Satelliten genutzt (siehe zum Beispiel Teng et al. (2001), Zhang et al. (2008), Zhu et al. (2010)).

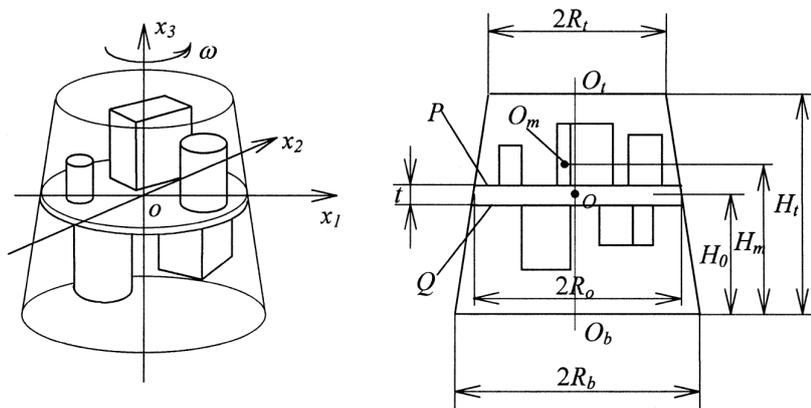


Abbildung 2.11.: Layout eines Satelliten mit zwei Ebenen, *Quelle: Teng 2001, S. 522*

Eine weitere Gruppe von Modellen beschäftigt sich mit dem Design von mehrschichtigen Chips, insbesondere mit sogenannten System-on-a-Chip (SoC). SoCs sind Chips, die mehrere Funktionseinheiten, wie zum Beispiel einen Prozessor, einen Grafikchip oder Speicherbausteine auf einem Chip vereinen. Sie finden häufig Anwendung, wenn möglichst platzsparende Lösungen gefragt sind, wie zum Beispiel Smartphones (Furber 2002). Solche Chips sind ebenfalls in Ebenen aufgebaut. Ihr Layout kann jedoch nicht mit den bisher vorgestellten Ansätzen gelöst werden, da die vielfältigen Möglichkeiten, die einzelnen Ebenen miteinander zu verbinden, in anderen Anwendungsfällen nicht vorhanden sind. Mit dem Design solcher Chips beschäftigen sich unter anderem Wu et al. (2010) oder Jiang et al. (2009).

### 2.4.2. **Echt dreidimensionale Ansätze**

Als „echt“ dreidimensionale Probleme sollen im folgenden Modelle bezeichnet werden, bei denen keine Ebenen explizit definiert sind, beziehungsweise bei denen Einheiten nicht fest einer Ebene zugeordnet werden. Nach dieser Abgrenzung kann auch ein diskretes Modell als echt dreidimensional bezeichnet werden, obwohl es durch seine diskreten Abstände implizit bereits einzelne Ebenen enthält.

Anwendungsfälle für solche dreidimensionalen Problemstellungen lassen sich im Bereich der Produktion oder Logistik kaum finden, da in diesen Fällen fast nie Einheiten übereinander, sondern immer in einer Ebene angeordnet werden. Auch Bürogebäude besitzen immer einzelne Etagen, so dass in diesen Fällen eher die Ansätze aus Abschnitt 2.4.1 angewandt werden. In der Literatur werden deshalb solche Modelle eher im Bereich des Produktdesigns angewandt, wie der Kabelführung innerhalb eines elektrischen Antriebs (Lengauer und Schäfer 2001). Allerdings gehen diese Problemstellungen oft in den Bereich der Packprobleme über, da zwar Wechselwirkungen zwischen Komponenten in Nebenbedingungen behandelt werden, aber Flüsse (wie im genannten Beispiel die Verkabelung) eher eine untergeordnete Rolle spielen. Andererseits ließen sich solche Methoden auch bei der Planung großtechnischer Anlagen nutzen, die aus Einheiten sehr unterschiedlicher Größe bestehen und sich deswegen schlecht in Ebenen abbilden lassen. Für diesen Bereich gibt es bisher noch fast keine Veröffentlichungen. In Kapitel 3 werden die genauen Anforderungen für diesen Anwendungsfall vorgestellt und ein entsprechendes Modell entworfen. Eine Möglichkeit zur Lösung dreidimensionaler Layoutprobleme ist wie in Abschnitt 2.4 bereits erwähnt, die Verwendung des quadratischen Zuordnungsproblems. Ein solches Vorgehen wird beispielsweise von Köpcke und Schröder (2004) verfolgt. In diesem Fall sollen unterschiedliche, quaderförmige Komponenten einer Chemieanlage so positioniert werden, dass einerseits minimale Flusskosten erreicht und andererseits die Nebenbedingungen bestmöglich erfüllt werden. Die Nebenbedingungen sind entsprechend der Anwendung technischer Natur und bedingen beispielsweise eine Positionierung innerhalb des Stahlbaus, in einer bestimmten Ebene oder auch in Relation zu einer bestimmten Komponente. Das Ergebnis einer solchen Planung ist in Abbildung 2.12 zu sehen.

Obwohl derartige Layoutprobleme auf dem quadratischen Zuordnungsproblem basieren, lassen sie sich aufgrund ihrer vielfältigen Nebenbedingungen nur noch bedingt diesem vergleichen. Auch die betrachtete Instanzgröße unterscheidet sich deutlich. Während der Fokus der Forschung beim ursprüng-



Abbildung 2.12.: Aufstellungsplan einer Chemieanlage, *Quelle: Köpcke 2004, S. 9*

lichen quadratischen Zuordnungsproblem auf der optimalen Lösung von Instanzen zwischen 30 und 40 Einheiten lag, verwenden die Ansätze mit Nebenbedingungen ein sehr viel feineres Raster, in die eine vergleichbare Anzahl an Komponenten unterschiedlicher Größe angeordnet werden soll. Demnach verschiebt sich der Schwerpunkt von der reinen Flussminimierung zum Finden eines praktisch realisierbaren Layouts.

Während Objekte in Ansätzen, die auf dem quadratischen Zuordnungsproblem beruhen, lediglich in  $90^\circ$ -Schritten rotiert werden können, stellt Cagan et al. (1998) eine Möglichkeit vor, Objekte beliebiger Form frei drehbar in einem Layout zu positionieren. Die Objekte werden dabei durch ein Raster aufgelöst. In seinem Ansatz stellt er darüber hinaus verschiedene Arten von Zielkriterien bereit, unter anderem zur Berücksichtigung der Packdichte, der Flusskosten oder der Lage des Schwerpunktes. Durch die Annahme beliebiger Formen, Positionen und den unterschiedlichen Optimierungskriterien ist Cagans Modell äußerst flexibel. Natürlich führt diese Flexibilität zu einem erheblichen Rechenaufwand. Ein in seiner Veröffentlichung betrachtetes Beispiel berechnet die Anordnung von 8 identischen Würfeln in einem Raum, der der 8fachen Größe eines Würfels entspricht. Im Laufe der Optimierung nimmt er eine Verletzung von Nebenbedingungen (Überlappung von Komponenten oder Lage außerhalb des zur Verfügung stehenden Raumes) in Kauf und versucht diese zu reduzieren (siehe Abbildung 2.13). Die letzte gezeigte Konfiguration entstand nach 29.000 Iterationen und weist nur noch eine

leichte Abweichung von der optimalen Packungsdichte auf.

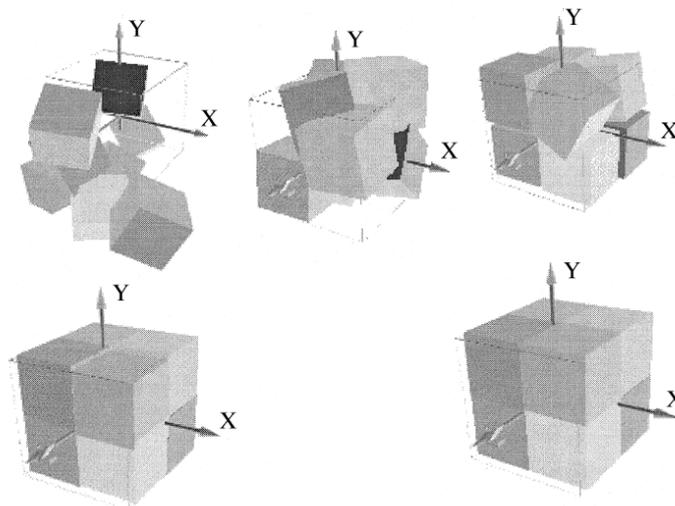


Abbildung 2.13.: Anordnung 8 identischer Würfel, *Quelle: Cagan 1998, S. 787*

In Anbetracht der notwendigen Rechenzeit sollte natürlich stets überlegt werden, ob das zu lösende Problem solche Freiheiten überhaupt benötigt. Würde die Anordnung der Würfel als quadratisches Zuordnungsproblem definiert, würde jede beliebige Lösung die optimale Packungsdichte gewährleisten. Demnach ist Cagans Ansatz auch eher für ein effizientes Produktdesign, in dem die freie Form und Rotation tatsächlich gegeben ist, sinnvoll.

Eine weitere Möglichkeit dreidimensionale Layoutprobleme zu lösen ist bei Furuholmen et al. (2010) zu finden. Dabei handelt es sich um ein stetiges Modell, das vom Prinzip ähnlich zu der Formulierung gemäß den Gleichungen 2.22 bis 2.29 ist. Gemeinsam ist den Ansätzen, dass sie rechteckige Flächen überschneidungsfrei anordnen und die Flüsse jeweils im Mittelpunkt einer Einheit beginnen beziehungsweise enden. Allein die Möglichkeit, Einheiten zu drehen, ist in diesem Ansatz nicht gegeben. Eine äquivalente Formulierung für den zweidimensionalen Fall ist auch bei Xie und Sahinidis (2008) zu finden.

Ein weiterer Unterschied ist das Konstruktionsverfahren, dem alle Lösungen gemäß Furuholmen et al. entsprechen müssen. Dabei wird davon ausgegangen, dass zusätzliche Einheiten sich nur an bereits platzierten Einheiten dem Layout hinzufügen lassen. Dabei müssen die Mittelpunkte zweier Einheiten jeweils aneinander ausgerichtet sein. Demnach werden alle Einheiten gemäß

der Regel  $m_i : a_d : m_j$  eingeplant. Dabei stehen  $m_i$  und  $m_j$  für zwei der  $n$  zu planenden Einheiten und  $a_d$  für eine Richtungsoperation der Menge  $A = \{N, S, E, W, U, D, \#\}$ , dabei stehen die Buchstaben für die Himmelsrichtungen beziehungsweise oben und unten,  $\#$  steht für „keine Operation“. Beim Hinzufügen einer Einheit in das Layout darf keine Überschneidung der Einheiten oder deren Sicherheitsbereiche auftreten. Darüber hinaus geht Furuholmen davon aus, dass jede Einheit pro Seitenfläche einen mittig angeordneten Ausgangs- oder Endpunkt für Rohrverbindungen hat und jeder Quader von einem Sicherheitsbereich variabler Größe umgeben ist. Der Aufbau dieser Einheiten und die Konstruktion des Layouts ist in Abbildung 2.14 zu sehen.

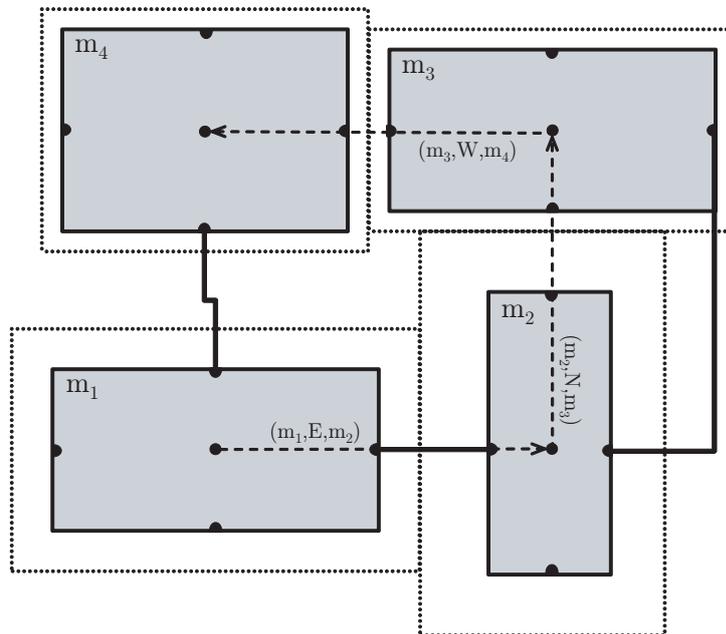


Abbildung 2.14.: Aufbau der Einheiten und Konstruktionsprinzip des Layouts nach Furuholmen

Furuholmen berücksichtigt in seinem Modell zwei Kostenarten. Zum einen fixe Kosten für das Einrichten einer Rohrverbindung, abhängig von ihrer Länge, sowie zur transportierten Menge proportionale Flusskosten. Es sei  $L$  die Menge aller durch das Konstruktionsverfahren erzeugbaren Layouts. Die Zielfunktion ergibt sich dann zu

$$\min_{l \in L} c = \sum_{i=1}^n \sum_{j>i}^n (p_{ij} d_{p,ij} + t_{ij} d_{s,ij}). \quad (2.40)$$

Dabei kennzeichnet  $p_{ij}$  die notwendigen Rohrverbindungen zwischen den Einheiten  $i$  und  $j$  und  $d_{p,ij}$  die kürzeste Entfernung zwischen zwei der jeweils sechs möglichen Verbindungsstellen. Der Parameter  $t_{ij}$  kennzeichnet die Transportmenge zwischen zwei Einheiten und  $d_{s,ij}$  die entsprechende Distanz. In der Veröffentlichung wird weiter eine Heuristik angegeben, mit der Layouts konstruiert werden können. Ein Ergebnis dieses Verfahrens ist in 2.15 zu sehen.

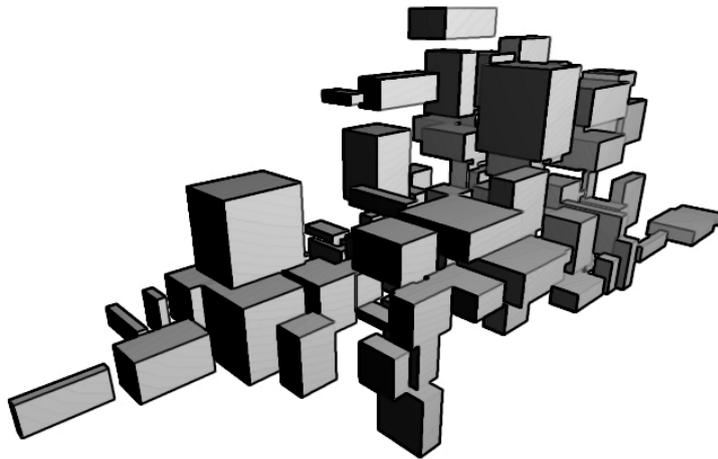


Abbildung 2.15.: Konstruktion eines Layouts aus 100 Einheiten, *Quelle: Furuholmen 2010, S.1*

### 2.4.3. Layoutplanung mit Gleichgewichtsbedingungen

Eine letzte Gruppe von Problemen, die hier betrachtet werden sollen, behandeln das dreidimensionale Layout von Produkten unter Berücksichtigung von Gleichgewichtsbedingungen. Zwar können auch in diesen Fällen Flussgrößen zwischen den Einheiten eine Rolle spielen, allerdings werden üblicherweise möglichst geringe Abweichungen aus einem Gleichgewicht verfolgt. Diese Bedingungen können statisch sein (z.B. die Gewichtsverteilung der Komponenten in einem U-Boot (Jang 2008)) oder auch dynamisch (Momentengleichgewicht bei einem Satelliten wie bei Teng et al. (2001) oder Zhang et al. (2008)).

Wie bei den Modellen im vorigen Abschnitt, wird auch hier versucht die Problemkomplexität auf verschiedene Weise zu reduzieren. Neben der Einteilung in Ebenen, die in allen Modellen angewendet werden, wird häufig auf eine genauere, aber auch aufwendigere Abbildung komplexer Geometrien in einem Raster verzichtet und stattdessen auf einfache Formen (Kreise und Rechtecke) zurückgegriffen (Teng et al. 2001). Nur wenige Ansätze lassen auch beliebige Formen zu, die anschließend durch mehrere umschreibende Kreise angenähert werden (Zhu et al. 2010).

Stellvertretend für die verschiedenen Ansätze sollen hier kurz die Rahmenbedingungen beim Design eines Satelliten wie bei Teng et al. (2001) genannt und erläutert werden:

- Alle Einheiten müssen innerhalb eines gegebenen Raumes angeordnet sein und dürfen sich nicht überschneiden
- Für eine stabile Rotation dürfen Abweichungen vom dynamischen Gleichgewicht eine bestimmte Grenze nicht überschreiten und sollen minimiert werden

Neben den auch in anderen Modellen benötigten Maßen der einzelnen Komponenten und des Behälters selbst, werden nun auch die einzelnen Massen, Massenschwerpunkte sowie die Rotationsgeschwindigkeit des Satelliten benötigt. Ähnlich zu den Layoutproblemen in der Ebene, wird eine Lösung auch bei diesen Modellen in zwei Stufen bestimmt. Zunächst erfolgt dabei eine Aufteilung auf die einzelnen Ebenen, um anschließend die einzelnen zweidimensionalen Probleme zu lösen.

## 2.5. Expertensysteme

Unabhängig vom gewählten Modell oder der darin berücksichtigten Anzahl an Dimensionen, existiert eine Vielzahl von Ansätzen, die bestehende Modelle in ein Expertensystem für eine ganzheitliche Planung integrieren. Diese Systeme sind als Software implementiert, in die der Nutzer alle notwendigen Daten eintragen kann und die auch das Entscheidungssystem bereitstellt. Dabei werden oft mehrere Ziele verfolgt, die unter anderem folgende Aspekte umfassen (Abdou und Dutta (1990) sowie Kumara et al. (1988)):

- Minimierung der Materialflusskosten
- Minimierung der Kosten für die Fördertechnik
- Maximierung des Sicherheitsniveaus

- Maximale Erfüllung von Nebenbedingungen bei überbestimmten Problemen

Je nach Anwendungsbereich können Expertensysteme jedoch völlig unterschiedliche Eingangsdaten benötigen und auch unterschiedliche Ergebnisse liefern. So steht bei Abdou und Dutta (1990) das Finden eines passenden Layoutmusters (lineare Anordnung, Blocklayout, etc.) im Vordergrund. Dazu berücksichtigt er beispielsweise das zu produzierende Produktspektrum, die geforderte Flexibilität oder auch den gewünschten Automatisierungsgrad. In einem anderen Ansatz können im Gegensatz dazu mehr Aspekte im Vordergrund stehen, die sich auf Wechselwirkungen zwischen den zu platzierenden Einheiten beziehen. So finden bei Kumara et al. (1988) beispielsweise Störgrößen wie die Geräuschentwicklung oder Wärmeproduktion von Maschinen Eingang in die Entscheidung. Ebenso kann allerdings auch die Frage der Zugänglichkeit zu einer Straße oder physikalische Größen wie das Gewicht von gelagertem Material eine Rolle spielen. Dies wird zum Beispiel bei der Planung einer Baustelle in Betracht gezogen (Tam et al. 2002).

Expertensysteme basieren immer auf einer Interaktion eines Nutzers mit dem als Entscheidungshilfe dienenden Computerprogramm. Aus diesem Grund wird in diesem Bereich häufig die sogenannte Fuzzy-Logik angewandt (siehe zum Beispiel Badiru und Arif (1996) oder Tam et al. (2002)). Dies dient dazu, Unschärfen der Sprache in messbare Größen zu wandeln. Beispielsweise lässt sich die Bedeutung von Nebenbedingungen (,muss erfüllt sein‘, ,sehr wichtig‘, ,wichtig‘, ,eher unwichtig‘) auf diesem Wege erfassen.

Dieser ausführliche Literaturüberblick beleuchtete die verschiedenen Ansätze, die zur Lösung von Layoutproblemen verfolgt werden. Ebenso wurden jeweils die Anwendungsgebiete aufgezeigt, für die die einzelnen Modelle entwickelt wurden. Dadurch sollte ein allgemeines Verständnis von Layoutproblemen vermittelt werden, das für das im folgenden Kapitel entwickelte Modell hilfreich ist.

# 3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen

Aufbauend auf den vorgestellten Ansätzen, soll nun ein Modell entwickelt werden, das geeignet ist, die Anforderungen technischer Großanlagen abzubilden. Den Anstoß für diese Arbeit lieferten Layoutprobleme im Rahmen des Designs der Materialforschungsanlage *International Fusion Materials Irradiation Facility* (IFMIF). Diese soll deshalb in Abschnitt 3.1 zunächst kurz vorgestellt werden. Im Anschluss werden in Abschnitt 3.2 die resultierenden Anforderungen für den allgemeinen Fall diskutiert werden, auf denen das Modell aufbaut. Dies umfasst beispielsweise die Charakteristika der einzelnen Komponenten, aber auch auftretende technische und wirtschaftliche Nebenbedingungen. Nachdem in Abschnitt 3.3 die Eignung bestehender Modelle diskutiert wurde, wird anschließend in den folgenden beiden Abschnitten 3.4 und 3.5 ein Modell auf Basis der Constraintprogrammierung entworfen.

## 3.1. Aufbau von IFMIF

IFMIF ist eine geplante Anlage, die Materialien auf ihre Eignung für den Einsatz in zukünftigen Fusionsreaktoren testen soll. Hierfür werden Deuteronen linear beschleunigt und auf einen flüssigen Lithium-Film gelenkt. Aus diesem treten Neutronen aus, die anschließend auf drei Testmodule treffen. In diesen Modulen befinden sich unterschiedliche Werkstoffproben, die für einen Einsatz in einem Fusionsreaktor in Frage kommen. Durch die Bestrahlung mit Neutronen werden diese Proben je nach Position innerhalb des Testfeldes unterschiedlich stark geschädigt. Das Feld, in dem sich die Proben befinden, ist dabei in drei Schädigungsbereiche aufgeteilt, in denen jeweils ein Modul platziert wird. Das Modul mit der höchsten Schädigungsrate wird als *High Flux Test Module* (HFTM) bezeichnet, die beiden anderen entsprechend als

### 3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen

---

*Middle- und Low Flux Test Module* (MFTM und LFTM). Das Prinzip der Forschungsanlage ist in Abbildung 3.1 dargestellt.

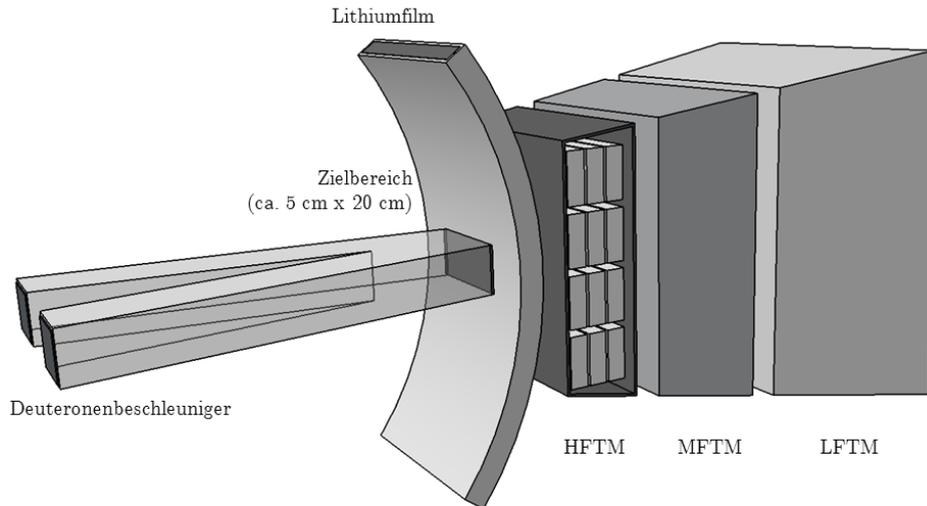


Abbildung 3.1.: Funktionsprinzip der Materialforschungsanlage IFMIF

Um eine ausreichend hohe Materialschädigung durch Neutronen zu erreichen, wurde eine sehr komplexe Anlage entworfen, die aus einer Vielzahl an Komponenten besteht. Neben den Hauptbestandteilen, zu denen neben den beiden Beschleunigern auch der Lithiumkreislauf und die Testzelle gehören, existiert eine große Zahl an Räumen, die für den Betrieb notwendig sind. Dazu zählen beispielsweise Hot Cells, in denen aktivierte Komponenten gehandhabt werden können, Versorgungsräume für Kühlsysteme oder auch Bereiche zur Lagerung aktivierter Abfälle. Zwischen all diesen Bestandteilen der Anlage finden Materialflüsse statt, die zum größten Teil ferngesteuert durch Kräne oder Manipulatoren ausgeführt werden.

Bisher ist geplant, die Werkstoffproben elf Monate pro Jahr zu bestrahlen. An diese Phase schließen sich Wartungsarbeiten an, die nach einem Monat abgeschlossen sein sollen. Aufgrund der vielen ferngesteuerten Handhabungsvorgänge in diesem Zeitraum, muss eine Layoutkonfiguration gefunden werden, die es erlaubt alle notwendigen Vorgänge innerhalb der gegebenen Zeit auszuführen. Die Transportzeiten und -kapazitäten der vorhandenen Technik müssen dabei berücksichtigt werden. Darüber hinaus werden auf technischer Seite unterschiedliche Nebenbedingungen an das Layout gestellt. Dies umfasst neben einer zwingenden relativen Positionierung zweier Komponenten

zueinander auch die Möglichkeit, Bereiche zu trennen, beispielsweise damit Sicherheitsbereiche nicht von Menschen betreten werden. Ein aus Sicht der Layoutplanung weiteres wichtiges Merkmal der Anlage ist die Tatsache, dass die einzelnen Räume oder Komponenten sehr unterschiedliche Größe besitzen, die im Layout berücksichtigt werden müssen. Das bedeutet auch, dass nicht immer für das gesamte Gebäude gültige Ebenen vorhanden sind. Abbildung 3.2 zeigt das bisher geplante integrierte Layout.

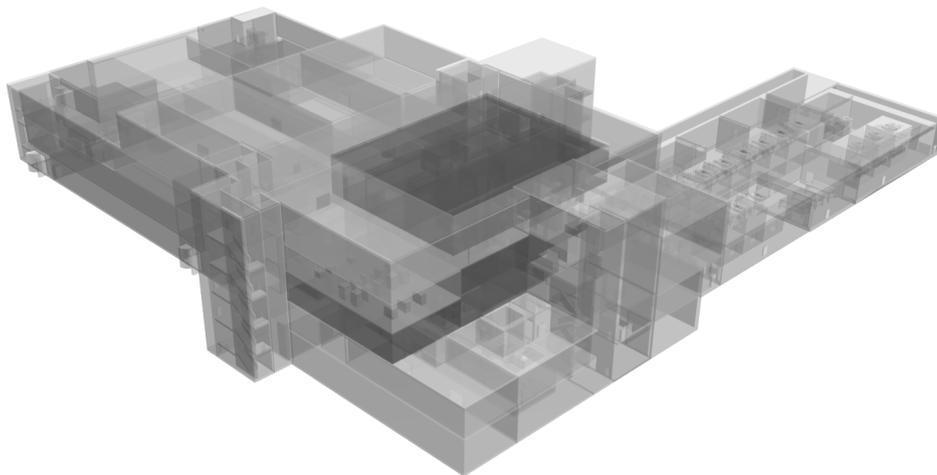


Abbildung 3.2.: Aktuelles Design von IFMIF

Die Design-Aktivitäten an IFMIF wurden im Juni 2013 vorerst eingestellt. Ob sie in Zukunft weitergeführt werden, ist bisher noch offen. Auch wenn eine Umsetzung eines verbesserten Designs in diesem Bereich unsicher ist, eignet sich IFMIF trotzdem als Beispiel für das entwickelte Modell, da für das aktuelle Layout eine umfangreiche Dokumentation vorliegt. Diese Dokumentation begründet bestimmte Entscheidungen und zeigt notwendige Bedingungen für den Entwurf auf. Das jetzige Layout der Anlage ignoriert weitestgehend Materialflüsse innerhalb des Gebäudes und baut allein auf den gegebenen Bedingungen auf. Insofern wäre bei einer Fortsetzung der Designaktivitäten sicherlich Optimierungspotential vorhanden.

## 3.2. Grundlagen des Modells

Ziel des Modells ist es, Layoutprobleme, wie sie im Rahmen des Designs von technischen Anlagen auftreten, abbilden zu können. Die im vorigen Abschnitt

genannte Forschungseinrichtung IFMIF ist dabei ein mögliches Beispiel. Daneben sollen sich aber beispielsweise ebenso Chemieanlagen und vergleichbare industrielle Anlagen abbilden lassen. Dazu sollen nun charakteristische Eigenschaften identifiziert werden, so dass diese als Anforderungen an das Modell umgesetzt werden können. Dabei soll besonderes Augenmerk auf die Unterscheidungsmerkmale, die in Abschnitt 2.2 genannt wurden, gelegt werden: beispielsweise Form und Größe der Einheiten, oder die Differenzierung nach statischen oder dynamischen Modellen. Anschließend muss betrachtet werden, wie die entsprechenden Eigenschaften sich im Modell umsetzen lassen. Dabei wird es notwendig sein, einige weniger wichtige Aspekte zu vernachlässigen, um die Komplexität des Problems zu begrenzen.

#### 3.2.1. Eigenschaften der zu planenden Anlagen

##### Komponenten

Für die Modellauswahl sind bei den zu platzierenden Komponenten im Wesentlichen die Form, Größe sowie die Orientierung von Interesse. Bei der betrachteten Art von Anlagen kann allgemein von Einheiten stark unterschiedlicher Größe ausgegangen werden. In Chemieanlagen gilt das beispielsweise für Reaktoren, Filter, Wärmetauscher oder Kolonnen, die in vielen verschiedenen Größen verbaut werden und auch zueinander sehr unterschiedliche Größen innerhalb einer Anlage haben können. Die Größe kann typischerweise für einen Filter bei etwa  $20 \text{ m}^3$  liegen und für Kolonnen bei einer Höhe von 10 bis 50 Meter Werte zwischen 10 und  $700 \text{ m}^3$  annehmen. Bei IFMIF liegen die Raumgrößen ebenso in einem breiten Spektrum von knapp  $40 \text{ m}^3$  bis  $4000 \text{ m}^3$ .

Für nahezu alle Objekte kann im Rahmen der Planung eine rechteckige Grundform angenommen werden. Während dies für die Form der Räume bei IFMIF offensichtlich ist, sind in der chemischen Industrie fast ausschließlich Komponenten mit runden Grundformen zu finden. Allerdings kann durch die Montage innerhalb eines rechtwinkligen Stahlbaus der freie Raum um ein rundes Objekt häufig nicht effizient genutzt werden. Deshalb kann auch dort von quaderförmigen Objekten ausgegangen werden. Das Seitenverhältnis der Objekte kann wie die Größe selbst in einem sehr weiten Rahmen schwanken. Für IFMIF reicht dieses von etwa eins für die Testzelle bis zu knapp neun für die beiden Räume mit den Linearbeschleunigern. Im Bereich der chemischen Industrie hätte der angesprochene Filter ebenso ein Seitenverhältnis von etwa eins; für die Kolonnen kann hingegen ein Wert von etwa zehn beobachtet

werden.

Alle Objekte können in 90°-Schritten um ihre Mittelachse gedreht werden. Es ist aus technischer Sicht nicht möglich, Komponenten um eine der beiden anderen Achsen zu drehen. Somit existieren vier mögliche Positionierungen im Raum.

### **Transporte**

Zwischen den einzelnen Komponenten innerhalb des Layouts muss Material transportiert werden, um den Betrieb der Anlage aufrecht zu erhalten. Dabei kann eine Anlage natürlich nicht als autarkes System betrachtet werden, sondern ist über außerbetriebliche Transporte mit der Umwelt verbunden. Demnach existieren auch Übergabepunkte für externe Transporte, die in einem Modell berücksichtigt werden müssen. Dies können neben Lieferzonen für LKW beispielsweise auch Rohrverbindungen zu benachbarten Anlagen sein.

Je nach Art des zu fördernden Gutes, können unterschiedliche Transportmittel zum Einsatz kommen. Für diese sind zum Teil weitere feste Einrichtungen notwendig, wie beispielsweise Schienen, Rollenbahnen oder auch nur freie Flächen zur Fortbewegung. Je nach verwendetem Fördermittel benötigen diese unterschiedlich viel Platz. Die Auswahl des Fördermittels kann zudem Einfluss auf das Layout haben, beispielsweise um sich überkreuzende Kranschienen zu vermeiden. Ebenso muss berücksichtigt werden, dass das Fördern eines Gutes oder von Personen (zum Beispiel beim Aufzug) je nach Richtung unterschiedliche Kosten verursachen kann.

Ferner kann davon ausgegangen werden, dass die Summe der ein- und ausgehenden Flüsse bei unterschiedlichen Objekten unter Umständen stark schwanken kann. So gibt es beispielsweise im Rahmen von IFMIF zentrale Elemente wie die Testzelle, die wesentlich mehr Materialflüsse aufweisen als beispielsweise Einrichtungen zur Aufnahme von Kühlkreisläufen oder anderem technischem Equipment.

### **Technische Bedingungen**

Eine weitere Gruppe von Merkmalen stellen die im Rahmen der Planung auftretenden technischen Bedingungen dar. Diese besitzen eine besondere Bedeutung, da sie für die Funktionserfüllung der Anlage entscheidend sind. Gemeinsam sind den im Rahmen dieser Arbeit betrachteten Anlagen sehr hohe Sicherheitsanforderungen. Relevant können in diesem Bereich unter anderem Sicherheitsbereiche um Gefahrenzonen sein, beispielsweise aufgrund

giftiger oder radioaktiver Substanzen, aus Feuerschutzgründen oder wegen Explosionsgefahr. Je nach Art der Gefahr und dem Schadensrisiko kann es notwendig sein, unterschiedliche Schutzmaßnahmen des Personals zu treffen: zum Beispiel feuerfeste Kleidung, Atemschutz oder bestimmte Sicherheitsunterweisungen; es kann auch erforderlich sein, den Einsatz von Menschen in Teilbereichen auszuschließen und auf Roboter zurückzugreifen. Neben den Gefahren für den Menschen muss ebenso auf Wechselwirkungen zwischen einzelnen Komponenten geachtet werden, beispielsweise einen Mindestabstand bei der Lagerung brennbarer Materialien von einer entsprechenden Gefahrenquelle.

Ein weiterer Aspekt kann die Sicherheit gegen Sabotage oder Terroranschläge sein. Dies könnte bedeuten, dass sensible Teile einer Anlage eher zentral angeordnet werden und von Sicherheitszonen umgeben sind. Auch wenn dies ein wichtiger Teil der Planung ist, wird er eher in einer späteren Planungsphase Berücksichtigung finden, in der das Layout bereits weitgehend festgelegt ist. Für den sicheren Betrieb spielt auch die Planung von Materialflüssen eine Rolle. So sollte darauf geachtet werden, dass die hochfrequentierten Wege sich möglichst wenig überschneiden, um im Betrieb Kollisionen zu verhindern. Im Bereich der Produktion ist dies unter anderem für den Staplerbetrieb von Interesse. Bei größeren Anlagen können darüber hinaus sich überschneidende Arbeitsbereiche von Kränen oder Manipulatoren zu Problemen führen. Ebenso müssen natürlich die Arbeitsbereiche des Personals beachtet werden, die in diesem Umfeld arbeiten.

Eine weitere technische Aspekt, der zum Teil auch Sicherheitsfragen berührt, ist die Zugänglichkeit der Anlage, beziehungsweise einzelner Komponenten. Diese muss sowohl während der Bauphase als auch während des Betriebs und der Wartung gewährleistet sein. Hierbei muss auch wiederum berücksichtigt werden, auf welche Weise die Zugänglichkeit gegeben sein muss: Das heißt, ob ein Austausch gegebenenfalls per Kran erfolgen muss oder ob die Zugänglichkeit für das Bedienpersonal ausreicht. Für die Zugänglichkeit spielt es ebenso eine Rolle, ob eine bestimmte Tätigkeit regelmäßig, zum Beispiel im Rahmen der jährlichen Wartungsarbeiten, erfolgt oder ob außerordentliche Operationen auftreten, beispielsweise wegen eines Bauteilversagens. Während in Chemieanlagen, die wirtschaftlich arbeiten müssen, solche Fehler auf ein absolutes Minimum reduziert werden, ist es bei IFMIF aufgrund der Komplexität und der Einzigartigkeit der Anlage sehr schwer, sämtliche möglichen Fehler auszuschließen. In diesem Fall würde ein nicht unwesentlicher Teil der Fehler bei Auftreten analysiert und eine spezifische Lösung ausgearbeitet werden, ohne auf eine Standardprozedur zurückzugreifen.

Auch der Prozessablauf an sich stellt technische Anforderungen, die bei der Planung berücksichtigt werden müssen. So wirken manche Komponenten aufgrund von Temperatur oder Vibrationen störend auf andere Prozesse, die deshalb räumlich zu trennen sind.

Charakteristisch für die betrachteten Großanlagen ist ebenso die Betriebsform: Fertigungsanlagen, beispielsweise im Automobilbau, laufen möglichst konstant und kleinere Störungen werden im laufenden Betrieb behoben; Stillstände sind in der Regel zwar kurz, treten aber häufiger auf. Großanlagen laufen hingegen über einen längeren Zeitraum (etwa ein bis fünf Jahre) durch, benötigen aber zwischen diesen Betriebsphasen längere Wartungsperioden, in denen alle auftretenden Mängel behoben werden. Entsprechend müssen alle Systeme auf diesen Zeitraum ausgelegt werden. Eine Wartungsphase lässt sich häufig wiederum in mehrere Abschnitte unterteilen:

- Vorbereitung während des Betriebs
- Herunterfahren der Anlage
- Wartungsarbeiten
- Hochfahren der Anlage
- Nacharbeiten im laufenden Betrieb

Zu den Vorbereitungen zählt zunächst eine Aufnahme sämtlicher auszuführenden Arbeiten, die entweder routinemäßig anfallen oder aufgrund von Verschleißerscheinungen notwendig sind. Dabei kommt auch bei funktionsfähigen Anlagenteilen, die jedoch eine erhöhte Ausfallwahrscheinlichkeit während der nächsten Betriebsphase haben, ein Austausch in Betracht. Neben diesen planerischen Tätigkeiten muss die Wartung auch organisatorisch vorbereitet werden. Dabei muss unter anderem sichergestellt sein, dass alle notwendigen Ersatzteile für die Wartung bereit stehen und auch dass das Personal für die anfallenden Aufgaben geschult ist.

Im Gegensatz zu Fertigungsanlagen kann das Herunterfahren einer Großanlage ein komplexerer Prozess sein, der eine bestimmte Sequenz erfordert. Auch in dieser Phase sowie zu Beginn der eigentlichen Wartung kann sich der Umfang der Tätigkeiten noch ändern, da der Zustand vieler Komponenten im Betrieb nicht ermittelt werden kann.

Nach dem Wiederanfahren der Anlage sind weitere Tätigkeiten notwendig, wie beispielsweise die Entsorgung ausgetauschter Teile oder eine Dokumentation der vorgenommenen Arbeiten.

Demnach weisen die betrachteten Anlagen sehr unterschiedliche Betriebsphasen auf, die auch an das Layout verschiedene Anforderungen stellen können.

Da fast immer die Anforderung besteht, die Wartung in möglichst kurzer Zeit abzuschließen, sollte auch das Layout auf einen schnellen Ablauf ausgerichtet sein. Demnach wird ein Layout stets eine Kompromisslösung für Phasen des Betriebs und der Wartung darstellen.

#### **Wirtschaftliche Bedingungen**

Neben den technischen Bedingungen ist auch die Wirtschaftlichkeit des zu planenden Layouts von Bedeutung. Dabei spielen vor allem die innerhalb der Anlage notwendigen Transporte eine Rolle. Im Rahmen der Einrichtung der Anlage fallen zunächst Kosten für die Einrichtung von Transportwegen (Rohre, Kräne und Kranbrücken etc.) als auch für das Bereitstellen der notwendigen Transportmittel mit ausreichender Kapazität an. Darüber hinaus müssen für jeden Transport Kosten in Abhängigkeit von der Menge und der Distanz berücksichtigt werden. Je nach Transportmittel werden diese im Vergleich zu den Einrichtungskosten einen größeren (z.B. Kran) oder kleineren (z.B. Gabelstapler) Anteil ausmachen.

Für die Wirtschaftlichkeit spielt bei der Layoutplanung ebenso die benötigte Fläche beziehungsweise der benötigte Raum eine Rolle. Je nach Grundstückspreis werden also unterschiedliche Layouts zu bevorzugen sein.

Ähnlich zu den technischen Anforderungen kann der Prozessablauf auch in wirtschaftlichen Fragen von Bedeutung sein. So können bestimmte Anordnungen von Maschinen mit Wechselwirkungen wie der Temperatur zwar technisch noch möglich sein, sich jedoch negativ auf die Produktqualität auswirken.

Die verschiedenen Gruppen von Einflussgrößen und Bedingungen sind nochmals in Tabelle 3.1 zusammengefasst.

Auch wenn nicht alle der betrachteten Anlagen diese Eigenschaften in vollem Maße aufweisen, so sind sie doch für den Großteil charakteristisch und sollten bei der Planung berücksichtigt werden. Natürlich ist es nicht möglich, alle Eigenschaften in einem Modell abzubilden. Daher sollen im folgenden Abschnitt die für das endgültige Layout wichtigsten Charakteristika identifiziert werden.

#### **3.2.2. Resultierende Anforderungen an das Modell**

Für die Entscheidung welche Eigenschaften tatsächlich berücksichtigt werden sollen, sind zwei Kriterien wesentlich. Dabei wird der Einfluss einer Eigenschaft auf das resultierende Layout im Verhältnis zu dem dafür notwendigen

Tabelle 3.1.: Übersicht wichtiger Eigenschaften technischer Anlagen

Kategorie	Eigenschaft
Technische Bedingungen	Zugänglichkeit
	Konstruktion
	Prozessstörungen
	Sicherheit
	Betriebsphase
Wirtschaftliche Bedingungen	Länge der Transportwege
	Platzbedarf
	Prozessgüte
	Transportaufwände
	Transportkapazitäten
Transporte	Externe Transporte
	Innerbetriebliche Transporte
Komponenten	Form
	Größe der Komponenten
	Orientierung

Modellierungsaufwand betrachtet. Dabei können durchaus unterschiedliche Modellierungsalternativen in Betracht kommen, die, je nachdem wie stark sie die Realität abstrahieren, mehr oder weniger schwierig umzusetzen sind.

Von sehr hoher Bedeutung sind dabei die Eigenschaften, die die Gestalt der zu platzierenden Einheiten betreffen. Bedingt durch die stark schwankenden Größen (zwischen 40 und 4000  $m^3$ ) und Formen der Komponenten wäre die Annahme einer identischen Gestalt, wie sie in der Grundform des quadratischen Zuordnungsproblems getroffen wird, sehr kritisch zu sehen. Auch die übliche Annahme in Layoutmodellen, dass alle Materialflüsse einer Komponente in einem einzelnen Punkt beginnen beziehungsweise enden, ist aufgrund der heterogenen Größe und Form nicht zutreffend. Da dies eine der wesentlichen Eigenschaften der betrachteten Anlagen ist, soll in diesem Punkt eine möglichst große Realitätsnähe erreicht werden.

Ebenso typisch für derartige Anlagen ist die Vielfalt der auftretenden Nebenbedingungen, die die Position einer Komponente innerhalb der Anlage oder

auch relativ zueinander bestimmen. In vielen Fällen sind diese unabdingbar, so dass ein Layout, das diese Bedingung verletzt, nicht umsetzbar ist. Da eine nachträgliche Anpassung des Layouts an diese Bedingung einen erheblichen Mehraufwand darstellt und auch nicht zwangsläufig aus dem berechneten Layout generiert werden kann, sollte das Modell in seinen Nebenbedingungen so flexibel sein, dass diese technischen Bedingungen abbildbar sind.

Ebenso interessant ist dabei der Aspekt, dass ein Planungsmodell auf verschiedenen Entwicklungsstufen eingesetzt werden könnte und in diesem Rahmen in der Regel voneinander abgegrenzte Planungsaufgaben existieren. Im Rahmen von IFMIF kann sich das beispielsweise zunächst auf das Design einzelner Komponenten oder Bereiche beziehen, aber auch in einer späteren Planungsphase beim Zusammenspiel der unterschiedlichen Bereiche bei der Aufteilung des Gebäudes. Dabei sollte berücksichtigt werden, dass innerhalb der Bereiche zwar die relative Positionierung eingehalten werden muss, es für die Einbindung in das Gebäude jedoch nicht nur die Möglichkeit gibt, eine Komponente in verschiedenen Orientierungen einzuplanen, sondern dass auch eine Spiegelung des Bereichs an einer vertikalen Ebene in der Regel keine Restriktionen verletzt. Für Komponenten, auf die dies zutrifft, sollten demnach unter Vernachlässigung einer Rotation um eine Horizontalachse nicht nur vier sondern acht Orientierungen möglich sein.

Der Aspekt der unterschiedlichen Betriebsphasen ließe sich auf unterschiedliche Weise darstellen. Einerseits ließe sich ein dynamisches Modell verwenden, wobei die verschiedenen Zeitpunkte den Betriebsphasen entsprechen. Dies ist allerdings eine sehr rechenaufwendige Modellierung, die zudem dem ansonsten starren Charakter der Anlage widerspricht. Da alle Komponenten innerhalb der Anlage fest montiert sind und somit in einem solchen Modell sehr hohe Standortwechselkosten verursachen würden, wäre ein Positionswechsel sehr unwahrscheinlich. Vorstellbar wäre allein der Fall, dass Komponenten bewegt werden müssen, um andere Anlagenteile zu erreichen. Da es sich dabei aber ausschließlich um bewegliche Komponenten handeln kann, käme in diesem Fall ebenso ein Entfernen aus dem Raster in Frage. Zudem werden in den verschiedenen Phasen in der Regel unterschiedliche Transportmittel beansprucht, sodass Wechselwirkungen zwischen den einzelnen Betriebsphasen nicht beachtet werden müssen. Dies gilt beispielsweise in einer Chemieanlage für Rohrverbindungen während des Betriebs oder Kräne in der Wartungsphase. Auch einige wirtschaftliche Aspekte wie die Berücksichtigung begrenzter Transportkapazitäten sollte über Nebenbedingungen erfasst werden können.

Ziel der Layoutplanung ist immer das Finden einer wirtschaftlichen Lösung, auch wenn „wirtschaftlich“ auf verschiedene Weise interpretiert werden kann.

Für das vorliegende Problem soll dabei die Minimierung des Transportaufwands im Vordergrund stehen. Betrachtet man die Planung von Chemieanlagen wie bei Köpcke und Schröder (2004) oder die hohen Transportkosten, die aufgrund der speziellen Umgebung im Rahmen von IFMIF entstehen, ist dies auch eine sinnvolle Zielgröße. Im Vergleich dazu stellt der Platzbedarf eines Layouts eine eher untergeordnete Größe dar.

Auch die Verwendung unterschiedlicher Transportmittel sollte in dem Modell berücksichtigt werden. Für das betrachtete Problem können zeitgleich sehr unterschiedliche Transportmittel eingesetzt werden, so dass die Annahme von identischen Kosten für jeden Transport diesem Sachverhalt nicht gerecht werden würde. Auch die unterschiedlichen Bewegungsräume und Kosten für die Einrichtung der Wege können nicht ohne weiteres vereinheitlicht werden.

Zusammenfassend lassen sich an das Modell also folgende Anforderungen stellen:

- Heterogene Raumgrößen und -formen müssen abbildbar sein.
- Materialflüsse können an unterschiedlichen Punkten einer Komponente beginnen oder enden.
- Vielfältige technische Bedingungen müssen erfasst werden können.
- Die Charakteristik der Transportmittel soll berücksichtigt werden; die Transportaufwände sind zu minimieren.

Auf Basis dieser Gesichtspunkte sollen nun im folgenden Abschnitt verschiedene Modellierungsansätze diskutiert werden. Dabei wird zunächst auf die in Kapitel 2 besprochenen Modelle eingegangen.

### 3.3. Eignung bestehender Ansätze

In seiner Grundform berücksichtigt das *quadratische Zuordnungsproblem* nur Komponenten gleicher Größe (vergleiche Abschnitt 2.3.1). Da dies den Anforderungen widerspricht, ist es in seiner Grundform nicht geeignet. Dies ließe sich jedoch mit einer Erweiterung lösen, in der sich eine Komponente über mehrere Rasterplätze erstrecken kann. In diesem Fall könnten die Materialflüsse in jedem der belegten Rasterplätze beginnen oder enden. Zwar würde das Ursprungsproblem dadurch etwas vereinfacht, je nach der gewählten Raster-einteilung ließe sich jedoch die Lösung präzisieren. Gerade für die Planung von Chemieanlagen würde die Einteilung in ein Raster der Aufteilung eines Stahlbaus sehr nahe kommen. Technische und wirtschaftliche Anforderungen ließen sich über entsprechende Nebenbedingungen, die beispielsweise Mindest- oder

### 3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen

---

Maximalabstände oder eine lineare Anordnung von Komponenten erzwingen, realisieren. Insofern würde sich das quadratische Zuordnungsproblem flexibel an die Gegebenheiten von Großanlagen anpassen lassen. Beachtet werden muss in diesem Fall jedoch, dass gerade bei wachsender Problemgröße oder einer feineren Rastereinteilung die Komplexität stark ansteigt und somit effiziente Lösungsmethoden erforderlich sind. Abbildung 3.3 zeigt einen solchen Ansatz mit unterschiedlich großen Komponenten.

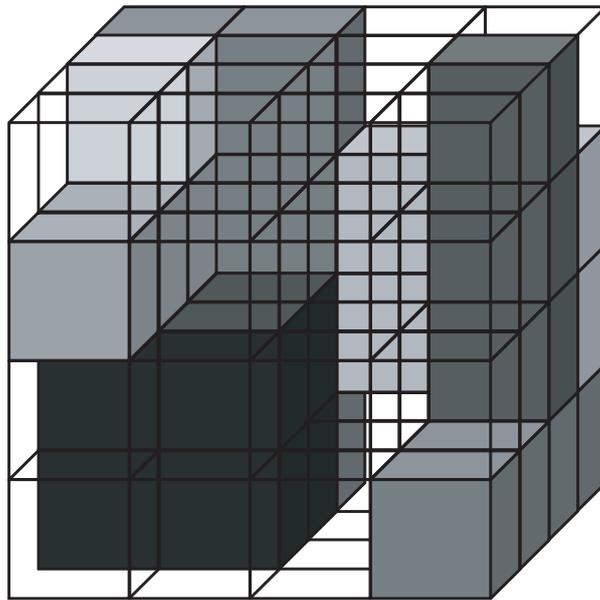


Abbildung 3.3.: Anordnung von Objekten unterschiedlicher Größe über eine Variante des quadratischen Zuordnungsproblems

Eine weitere Möglichkeit zur Layoutgenerierung waren die in Abschnitt 2.3.2 vorgestellten *Space Filling Curves*. Diese sind jedoch aus mehreren Gründen für den betrachteten Anwendungsfall ungeeignet. Dies liegt unter anderem daran, dass die Form der Komponenten nicht gesteuert werden kann und tendenziell eher kompakte Einheiten entstehen. Die Anfangs- und Endpunkte der Materialflüsse bleiben mehr oder weniger unberücksichtigt. Es kann lediglich die Reihenfolge der Einheiten entlang der Kurve beeinflusst werden. Auch weitere Nebenbedingungen lassen sich kaum abbilden, da hier nicht die Reihenfolge entlang der Kurve, sondern vielmehr die konkrete Position im Raster ausschlaggebend ist.

Ebenso lassen sich *graphenbasierte Modelle* nur schlecht für dreidimensionale Probleme anwenden (vergleiche Abschnitt 2.3.3). In diesen Fällen wäre immer ein zweistufiges Vorgehen notwendig, wobei erst eine Einteilung in Ebenen und anschließend eine davon unabhängige Layoutplanung der einzelnen Ebenen stattfinden würde. In diesen Fällen wäre es nicht ohne weiteres möglich, mehrere Komponenten in die Planung einzubeziehen, die sich über mehrere Ebenen erstrecken, da sich die Ebenen dann nicht mehr unabhängig voneinander optimieren lassen.

*Schnittmodelle*, wie in Abschnitt 2.3.4 vorgestellt, könnten durchaus ein interessanter Ansatz auch für dreidimensionale Probleme sein. In ihnen lassen sich unterschiedliche Raumgrößen gut abbilden und es liegt auch nicht zwangsläufig eine Bindung an vorgegebene Ebenen vor. Prinzipiell lässt sich der gesamte zur Verfügung stehende Raum durch Schnitte parallel zu den Koordinatenebenen einteilen. Gleichzeitig wäre dies ein sehr effizientes Verfahren zur Layoutplanung. Zudem bieten die Schnitte den Vorteil, dass das Layout in einer Weise geteilt wird, dass Verkehrswege sich gut entlang der Schnitte ausrichten ließen (Scholz 2010, S.22f). Problematisch ist jedoch, dass das Rotieren von Einheiten um ihren Mittelpunkt aufwendiger zu realisieren ist und sämtliche auftretenden Nebenbedingungen bereits beim Erzeugen des Schnittmusters berücksichtigt werden müssen. Auch die unterschiedlichen Anfangs- und Endpunkte der Materialflüsse wären in Komponenten mit flexibler Größe schwer zu realisieren.

Eine Alternative dazu wäre ein stetiges Modell mit Räumen starrer Größe. Dies würde dem Modell entsprechen, dessen Zielfunktion in Gleichung 2.33 angegeben ist. Darin können quaderförmige Komponenten abgebildet werden, die einen oder mehrere Punkte enthalten, von denen Materialflüsse ausgehen beziehungsweise enden. Die Orientierung im Raum wird über zwei Binärvariablen bestimmt. Über diese Art der Formulierung ist es darüber hinaus auch möglich Abstandsbedingungen zu definieren, oder die Positionierung einzelner Komponenten auf bestimmte Bereiche einzuschränken. Da bei Nebenbedingungen, die die relative Position zweier Komponenten zueinander einschränken, jeweils alle vier Orientierungsmöglichkeit berücksichtigt werden müssen, nehmen diese eine relativ komplexe Form an. Trotz dessen können Nebenbedingungen sehr flexibel hinzugefügt werden. Da es sich bei dieser Art der Formulierung jedoch um ein nichtlineares Problem handelt, ist das Finden von Lösungen sehr schwer. Ein entsprechendes Modell, das auf Nebenbedingungen zur Vermeidung von Überschneidungen beschränkt war, wurde innerhalb der Modellierungsumgebung Aimms implementiert und mit vorhandenen Standardmethoden gelöst. Da selbst ein sehr kleines Problem

mit lediglich 3 Komponenten und maximal je zwei Punkten pro Komponente bereits zwei Tage zur Bestimmung einer optimalen Lösung benötigte, wurde der Ansatz nicht weiter verfolgt. Zwar ließe sich das Modell durch angepasste Methoden sicherlich deutlich schneller lösen, allerdings scheint das Verfahren für größere Instanzen, die darüber hinaus eine Reihe von Nebenbedingungen enthalten, gänzlich unpraktikabel.

Aus diesen Gründen erscheint eine Variante des quadratischen Zuordnungsproblems als der erfolgversprechendste Ansatz. Dafür ist allerdings ein effizientes Lösungsverfahren notwendig, das in akzeptabler Zeit zulässige Lösungen findet. Je nach Art kann dieses Ziel durchaus auch variieren. Bei sehr stark restringierten Problemen kann bereits das Finden einer zulässigen Lösung, beziehungsweise einer Lösung, die möglichst wenige Bedingungen verletzt, ausreichen. Ist der Lösungsraum größer, wird die Suche nach einer möglichst guten oder der optimalen Lösung das Ziel sein.

## 3.4. Ansatz über Constraintprogrammierung

### 3.4.1. Grundlagen von CP

Eine Möglichkeit der Problemlösung ist die Verwendung der sogenannten Constraint-logischen Programmierung (CP). Darunter versteht man ein Lösungs- bzw. Optimierungsverfahren für kombinatorische Probleme. Diese haben dabei grundsätzlich einen ähnlichen Aufbau wie Probleme der mathematischen Optimierung. Während jedoch eine Nebenbedingung zum Beispiel eines linearen Optimierungsproblems lediglich eine Grenze darstellt, die bei der Lösung eingehalten werden muss, steht hinter einer Nebenbedingung im Rahmen der Constraintprogrammierung immer eine Logik, die im Rahmen des Lösungsverfahrens aktiv den Lösungsraum eingrenzt.

Unterschieden werden darüber hinaus im Allgemeinen sogenannte *Constraint-Satisfaction-Probleme* und *Constraint-Optimierungsprobleme*. Unter einem Constraint-Satisfaction-Problem (CSP) versteht man nach Hofstedt und Wolf (2007, S. 72):

**Definition 2** *Ein Constraint-Satisfaction-Problem (CSP) ist eine Konjunktion  $C = C' \wedge E_1 \wedge \dots \wedge E_n$  von Constraints aus  $\mathcal{CS}_{\mathcal{FD}}$  über den Variablen  $x_1, \dots, x_n \in X_{\mathcal{FD}}$ , so dass*

- *die Constraint-Konjunktion  $C'$  keine Constraints der Form  $x \in D$  enthält,*

- $\text{var}(C) \subseteq \{x_1, \dots, x_n\}$  gilt und
- die Constraints  $E_i$  die Form  $x_i \in \{e_{i,1}, \dots, e_{i,k_i}\}$  haben. Das Constraint  $E_i$  ordnet der Variablen  $x_i$  eine endliche Menge von Werten zu, diese Menge  $D(x_i)$  wird als Domäne der Variablen  $x_i$  bezeichnet.

Das bedeutet, dass die Bedingungen eines CSP sich aus den Definitionen des Wertebereichs aller Variablen (entspricht den Constraints  $E_i$ ) sowie den „echten“ Constraints  $C'$  zusammensetzt. Diese sind jeweils über einer Teilmenge der Variablen des CSP definiert. Bei einem CSP steht somit der Aspekt im Vordergrund, im Rahmen der gesetzten Bedingungen eine zulässige Lösung zu finden. Von einem Constraint-Optimierungsproblem spricht man in Anlehnung an Hofstedt und Wolf (2007, S. 299f), wenn ein CSP um eine Zielfunktion ergänzt wird, die maximiert oder minimiert werden soll:

**Definition 3** Gegeben sei ein CSP  $C = C' \wedge E_1 \wedge \dots \wedge E_n$ , sowie eine Zielfunktion  $f(x_1, \dots, x_n)$ . Hinsichtlich der Funktion  $f$  spricht man von einem Constraint-Optimierungsproblem, wenn eine oder alle Lösungen gesucht sind, die die Zielfunktion minimieren bzw. maximieren.

Der Aufbau und das Lösungsverfahren eines CSP soll im Folgenden anhand eines kleinen Zahlenbeispiels veranschaulicht werden. Gegeben seien die Variablen  $x \in \{2, 3, 4\}$ ,  $y \in \{2, 3, 4\}$  und  $z \in \{4, 5, 6, 7, 8\}$ , sowie die beiden Constraints  $x < y$  und  $x \cdot y = z$ .

Zunächst soll die Bedingung  $x < y$  auf die Variablen angewendet werden. Es folgt daraus einerseits, dass  $x$  nicht den Wert 4 annehmen kann, da kein Wert der Domäne  $D_y$  diesen Wert übersteigt und andererseits, dass  $y$  nicht den Wert 2 annehmen kann, da dies ebenfalls die Bedingung verletzen würde. Folglich können die beiden Domänen auf  $D_x = \{2, 3\}$  sowie  $D_y = \{3, 4\}$  eingegrenzt werden. Betrachtet man nun im nächsten Schritt die Bedingung  $x \cdot y = z$ , grenzt diese den Wertebereich von  $z$  auf die Alternativen  $D_z = \{6, 7, 8\}$  ein.<sup>5</sup> Dies folgt aus der Tatsache, dass kleinere oder größere Werte den bereits eingegrenzten Wertebereichen von  $x$  und  $y$  widersprechen würden. Des Weiteren lässt sich aus dieser Bedingung ableiten, dass  $x$  nicht den Wert 3 annehmen kann. In diesem Fall würde das Produkt aus  $x$  und

---

<sup>5</sup>Je nach Betrachtungsweise könnte an dieser Stelle natürlich auch der Wert 7 bereits ausgeschlossen werden. Dabei ist jedoch entscheidend, ob bei der Auswertung des Constraints lediglich die Randwerte der Domänen berücksichtigt werden, oder tatsächlich alle Kombinationen der möglichen Werte. Nur im zweiten Fall würde der Wert sieben nicht auftauchen und die Domäne von  $z$  entsprechend auf  $D_z = \{6, 8\}$  eingegrenzt werden.

### 3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen

---

$y$  in jedem Fall den Wertebereich  $D_z$  übersteigen, folglich gilt  $x = 2$ . Dieser Prozess des logischen Schließens und die damit verbundene Eingrenzung des Wertebereichs der Variablen wird als Propagieren bezeichnet.

Für die weitere Lösungsfindung muss eine Fallunterscheidung für die verbliebenen Variablen vorgenommen werden. Dabei muss zunächst eine Variable ausgewählt und anschließend der Wertebereich für diese Variable aufgeteilt werden. Eine Möglichkeit wie der sich ergebende Suchbaum aussehen könnte, zeigt Abbildung 3.4.

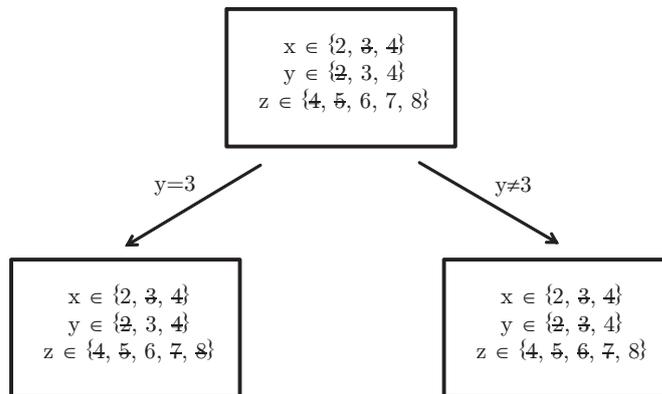


Abbildung 3.4.: Suchbaum eines CSP mit drei Variablen

In diesem Fall wird eine Fallunterscheidung für den Wertebereich von Variable  $y$  durchgeführt. Wendet man anschließend wieder die Nebenbedingungen auf die beiden Möglichkeiten an, werden dabei die Wertebereiche aller Variablen auf einen Wert eingegrenzt, das bedeutet es wurde für beide Zweige eine Wertebelegung gefunden, die den Constraints entspricht.

Beim Propagieren werden, ausgehend von der im letzten Schritt mit einem Wert belegten Variablen, alle Nebenbedingungen geprüft, in denen die betreffende Variable enthalten ist. Werden durch das Propagieren der Nebenbedingungen die Wertebereiche weiterer Variablen eingeschränkt, so werden auch für diese Variablen alle verknüpften Nebenbedingungen geprüft. Dieser Prozess erfolgt solange, bis keine weitere Einschränkung der Domänen auf diesem Weg mehr möglich ist.

Je nachdem, wie die Auswahl der Variablen und die anschließende Wertebelegung erfolgt, lässt sich eine Lösung früher oder später finden. Entsprechend fällt der Suchbaum dabei kleiner oder größer aus. Deshalb besitzen diese Ele-

mente eine hohe Bedeutung für die Lösungsfindung und sind wesentlicher Teil der Suchstrategie. Darüber hinaus ist auch eine geschickte Problemformulierung von Vorteil, wobei über entsprechende Constraints möglichst früh im Lösungsprozess die Wertebereiche eingegrenzt werden und damit Lösungen ausgeschlossen werden können.

Das lässt sich beispielsweise an einem Beispiel der Personaleinsatzplanung erkennen. Es werde angenommen, es soll ein Schichtplan für ein Call-Center erstellt werden, in dem mehrere Voll- und Teilzeitkräfte an 6 Tagen der Woche eine definierte Besetzungstärke abdecken. Zusätzlich muss dabei berücksichtigt werden, dass Mitarbeiter nicht länger als ihre jeweilige Sollstundenzahl eingesetzt werden können und regelmäßig freie Tage haben. Es soll des Weiteren angenommen werden, dass keine nennenswerten Überkapazitäten hinsichtlich der Mitarbeiterzahl bestehen.

Die Sollstundenzahl kann als Kapazitätsgrenze für jeden Mitarbeiter betrachtet werden und in einer entsprechenden Nebenbedingung umgesetzt werden. Ebenso lässt sich die Forderung nach freien Tagen als Kapazitätsgrenze definieren, beispielsweise, indem der Mitarbeiter an sieben aufeinander folgenden Tagen maximal fünfmal eingesetzt werden kann. Zusammen mit der Nebenbedingung, die die Besetzungstärke für die einzelnen Tage festlegt, ließe sich nun ein Schichtplan erstellen.

Je nachdem, wie die Variablen nach und nach mit Werten belegt werden, kann es zu der Situation kommen, dass ein Mitarbeiter über einen längeren Zeitraum nicht eingesetzt wird, da dies prinzipiell zunächst nicht den Nebenbedingungen widerspricht. Allerdings kann es dabei passieren, dass am Ende der Planungsperiode die anderen Mitarbeiter ihre Sollstundenzahl bereits erreicht haben und unter Berücksichtigung der notwendigen freien Tage die Besetzungstärke nicht mehr erreicht werden kann. Dies würde bedeuten, dass auf diesem Pfad keine Lösung existiert. Die Ursache dieses Konflikts lag bereits in einer frühen Phase, nämlich der langen freien Phase des betreffenden Mitarbeiters. Dies könnte durch ein zusätzliches Constraint verhindert werden, das eine Untergrenze für die Einsatzzeiten eines Mitarbeiters in einem begrenzten Zeitraum definiert. Das könnte beispielsweise durch die Forderung umgesetzt werden, dass ein Mitarbeiter an sieben aufeinanderfolgenden Tagen mindestens drei Mal arbeiten muss. Jede Lösung, die die ursprünglichen Bedingungen erfüllt, würde zwangsläufig auch diese Bedingung erfüllen. So gesehen, wäre das zusätzliche Constraint redundant. Da es jedoch hilft den Suchraum bereits frühzeitig einzugrenzen, ist es trotzdem sinnvoll für die Lösungsfindung.

Im Allgemeinen wird der Auswahl der Variablen eine größere Bedeutung bei-

gemessen als der darauf folgenden Wertauswahl (Arbelaez et al. 2009, S. 2). Für den in dieser Arbeit behandelten Anwendungsfall gilt dies jedoch nicht unbedingt. Aufgrund des potenziell sehr großen Lösungsraums kann durch eine geschickte Wertauswahl das Finden zulässiger und guter Lösungen erheblich beschleunigt werden. Die Wertauswahl besitzt hier auch einen höheren Stellenwert als in der Literatur beschrieben, da durch das Propagieren nur in begrenztem Maße Lösungen ausgeschlossen werden können. So kann zwar noch gefolgert werden, welche Positionen ein bestimmtes Objekt im Raster annehmen kann und dessen Wertebereich entsprechend eingegrenzt werden; für die folgenden Objekte hängt dies jedoch wiederum von der Entscheidung auf der vorigen Stufe ab. Entsprechend kann für einen Suchpfad nur schwer eine Aussage getroffen werden, ob am Ende tatsächlich eine zulässige Lösung gefunden wird, beziehungsweise ab welchem Entscheidungspunkt dies nicht mehr der Fall ist. Aus diesem Grund kann auch die Wertauswahl nicht vernachlässigt werden. Die für das behandelte Modell entwickelten und betrachteten Ansätze werden in Abschnitt 4.1.3 beschrieben.

Bei der Variablen- und Wertauswahl wird jeweils zwischen statischen und dynamischen Verfahren unterschieden. Unter einer statischen Auswahl versteht man dabei eine vorgegebene Reihenfolge, in der die Entscheidungsvariablen mit Werten belegt werden. Bei dem vorigen Beispiel wäre dies der Fall, wenn zunächst die Mitarbeiter für den ersten Tag des Planungsmonats eingeteilt würden und anschließend sequentiell von Tag zu Tag weitergegangen würde. Dem stehen dynamische Verfahren gegenüber. Eines der häufigsten Verfahren ist dabei die Auswahl der Variablen nach der Domänengröße. Dahinter steht der Gedanke, dass Variablen, die nur wenige unterschiedliche Werte annehmen können, im Lösungsprozess möglichst früh berücksichtigt werden sollten, um ein späteres Scheitern eines Lösungswegs zu verhindern. Des Weiteren kann auch untersucht werden, welche Variable im vorigen Schritt, relativ betrachtet, die größte Domänenreduktion erfahren hat, da diese offenbar sensibler reagiert als andere Variablen.

Neben den genannten Mechanismen kann im Rahmen eines Constraint-Optimierungsproblems auch noch eine untere Schranke eingeführt werden, die dabei hilft, die Qualität einer Lösung oder eines Lösungszweigs hinsichtlich eines Optimums einzuschätzen. Wie bei anderen Optimierungsproblemen kann diese Einschätzung natürlich nur so gut sein, wie die untere Schranke selbst. Dabei steht die Berechnung einer unteren Schranke auch im Rahmen von Constraint-Problemen im Spannungsfeld zwischen der Dichte der Schranke und der Komplexität ihrer Berechnung. Die Wahl einer unteren Schranke wird in Abschnitt 4.2 weiter behandelt.

### 3.4.2. Anwendung auf das Modell

Die Anwendung der Constraint Programmierung auf eine Variante des quadratischen Zuordnungsproblems bietet in Hinblick auf die in Abschnitt 3.2.2 formulierten Anforderungen eine Reihe von Vorteilen.

Um die Forderung erfüllen zu können, dass Materialflüsse an unterschiedlichen Punkten innerhalb eines Objektes beginnen oder enden können, werden diese Objekte ihrer Größe entsprechend aus mehreren Variablen aufgebaut. Diese sind über Nebenbedingungen miteinander verbunden. Die höhere Zahl an Variablen bedeutet dabei aber nicht, dass sich der Suchbaum im gleichen Maß vergrößert. Da je nach räumlicher Ausdehnung eines Objekts, dessen Lage bereits durch bis zu drei Variablen definiert ist, werden lediglich diese im Suchbaum berücksichtigt; die restlichen Variablen werden durch die Nebenbedingungen beim Propagieren gesetzt. Somit lassen sich unterschiedliche Raumgrößen bei hoher Flexibilität bezüglich der Materialflüsse abbilden.

Auch für die Umsetzung von Nebenbedingungen bezüglich der Positionierung der Objekte innerhalb des Layouts ist das Konzept gut geeignet. Gerade bei Nebenbedingungen, die die relative Position zweier Objekte zueinander betreffen (zum Beispiel ein Objekt über dem anderen) lassen sich im Zuge des Propagierens viele Lösungen ausschließen. Dies wäre innerhalb eines gemischt-ganzzahligen Optimierungsproblems in Verbindung mit Branch-and-Bound Ansätzen oder Schnittebenenverfahren aufwendiger zu realisieren.

Für die Umsetzung von Constraint-Problemen existiert eine Reihe von Bibliotheken. Die Implementierung des Modells erfolgte mit Hilfe des IBM ILOG CP Optimizers sowie zum Teil mit der zugehörigen Modellierungssprache OPL.

## 3.5. Modellformulierung und Nebenbedingungen

Im Folgenden sollen nun der Aufbau des Modells und die in dem Modell integrierten Nebenbedingungen erörtert werden. Dazu ist es zunächst notwendig, die im Weiteren verwendeten Begriffe voneinander abzugrenzen.

Innerhalb des Layouts soll eine Reihe von *Objekten* positioniert werden. Darunter versteht man eine quaderförmige Einheit, die über ihre Länge, Breite und Höhe definiert ist. Da es sich um ein diskretes Modell handelt, müssen diese Maße ebenfalls ganzzahlige Werte (größer eins) annehmen.

Ein Objekt ist aus ein oder mehreren *Blöcken* aufgebaut. Ein Block bildet

innerhalb des Modells die kleinste Einheit und ist somit ein Würfel mit der Kantenlänge eins. Im Modell entspricht jedem Block eine Variable, deren Wert die Position des Blocks angibt. Die einzelnen Blöcke, beziehungsweise die sie repräsentierenden Variablen eines Objekts, stehen über Nebenbedingungen miteinander in Verbindung.

Alle Objekte müssen während der Lösungsfindung innerhalb des dreidimensionalen *Rasters* platziert werden. Dieses ist wie auch die Objekte durch seine (ganzzahligen) Außenmaße gekennzeichnet. Je nach Planungsaufgabe kann es sich dabei beispielsweise um das umgebende Gebäude handeln.

Unter einer *Position* versteht man einen einzelnen Platz innerhalb des Rasters, der als Gegenpart zu einem Block ebenfalls einem Würfel mit Kantenlänge eins entspricht. Äquivalent zu den einzelnen Blöcken, entspricht auch jeder Position im Raster jeweils eine Variable, deren Wert den Block kennzeichnet, der dieser Position zugeordnet ist. Die Tatsache, dass die Zuordnung von Block zu Position somit doppelt existiert, vereinfacht die Formulierung der Nebenbedingungen.

Die Objekte seien im Folgenden mit  $o \in \{1, 2, \dots, O\}$  indexiert, ihre Maße werden mit den Parametern  $l_o$ ,  $b_o$  und  $h_o$  bezeichnet. Das Objekt  $o$  umfasst  $l_o \cdot b_o \cdot h_o = n_o$  einzelne Blöcke. Die Maße des zur Verfügung stehenden Rasters werden dem gegenüber mit  $L$ ,  $B$  und  $H$  bezeichnet. Sowohl die Indizes der Blöcke eines Objekts, als auch die Indizes der Positionen im Raster werden nacheinander in x-, y- und z-Reihenfolge vergeben (vergleiche Abbildung 3.5). Die Anzahl der Positionen im Raster ergibt sich zu  $L \cdot B \cdot H = N$ . Da die Anzahl der Positionen und Blöcke für eine Zuordnung identisch sein muss, ist es gegebenenfalls notwendig, weitere Einzelblöcke ohne Materialflüsse oder Beziehungen zu anderen Blöcken hinzuzufügen. Die Variablen, die die Position eines Blocks  $i$  angeben, werden mit  $\pi_i$  bezeichnet, die entgegengesetzten Variablen, die den Block auf einer bestimmten Position  $j$  kennzeichnen, mit  $\rho_j$ . Der Wertebereich beider Gruppen von Variablen liegt jeweils zwischen  $1, \dots, N$ . Des Weiteren seien die Koordinaten der  $j$ -ten Position mit  $x_j$ ,  $y_j$  und  $z_j$  gegeben. Die relative Position eines Blocks  $i$  innerhalb des zugehörigen Objekts  $o$  sei mit den Koordinaten  $x'_{i,o}$ ,  $y'_{i,o}$  und  $z'_{i,o}$  gegeben. Ferner seien die Koordinaten einer Position  $j$  oder eines Blocks  $i$  verkürzt mit dem Tupel  $p_j = (x_j, y_j, z_j)$  beziehungsweise  $p'_{i,o} = (x'_{i,o}, y'_{i,o}, z'_{i,o})$

Basierend auf diesen Definitionen lässt sich zunächst ein grundlegendes Modell formulieren. Weitere Nebenbedingungen, die zum einen realistischere Planungsoptionen bieten und zum anderen den Suchprozess beschleunigen, werden in Abschnitt 3.5.1 vorgestellt. Da hinter den Nebenbedingungen häufiger eine komplexere Logik steht, lassen sie sich häufig nicht mehr formal aus-

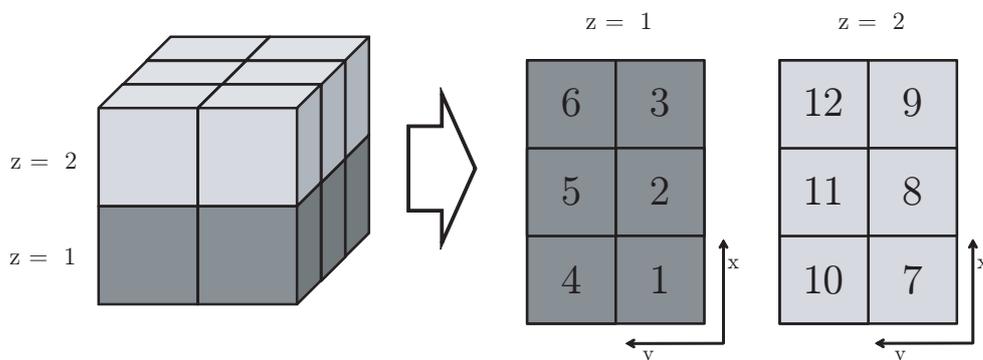


Abbildung 3.5.: Fortlaufende Indizes der Blöcke innerhalb von Objekten

formulieren. Aus diesem Grund sollen die Nebenbedingungen hier und im Folgenden nur verbal formuliert werden; die dahinter stehende Logik wird getrennt erläutert. Mit den Transportmengen  $t_{h,i}$  zwischen zwei Blöcken  $h$  und  $i$  sowie der Distanz  $d_{j,k}$  zwischen zwei Positionen  $j$  und  $k$ , lässt sich das Problem wie folgt formulieren:

$$\min f(\pi) = \sum_{h=1}^N \sum_{i=1}^N t_{h,i} \cdot d_{\pi_h, \pi_i} \quad (3.1)$$

u.d.N.:

$$\begin{aligned} & allDifferent(\pi) \\ & inverse(\pi, \rho) \\ & object(\pi_h, \dots, \pi_{h+n_o}) \quad \forall o \in 1, \dots, O \\ & \pi_i \in \{1, \dots, N\} \quad \forall i \in 1, \dots, N \\ & \rho_j \in \{1, \dots, N\} \quad \forall j \in 1, \dots, N \end{aligned}$$

Dabei bedeutet die Bedingung „*allDifferent()*“, dass alle darin berücksichtigten Variablen unterschiedliche Werte annehmen müssen. Auf das Problem übertragen, heißt das, dass allen Blöcken unterschiedliche Positionen zugewiesen werden müssen und dabei kein Wert mehrfach auftritt. Da die Kardinalität der Menge der Blöcke und Positionen identisch ist, folgt daraus, dass es sich bei der Zuordnung um eine bijektive Abbildung handelt. Diese Nebenbedingung lässt sich im Rahmen eines CP-Problems effizient handhaben

und entspricht den ersten beiden Nebenbedingungen des in Abschnitt 2.3.1 vorgestellten quadratischen Zuordnungsproblems.

Die Nebenbedingung „*inverse()*“ stellt sicher, dass die Zuordnungsvariablen  $\pi$  und  $\rho$  konsistent bleiben. Demnach muss gelten:

$$i = \rho_{\pi_i} \quad \forall i \in 1, \dots, N \quad (3.2)$$

Aufgrund der Bijektivität gilt die Umkehrung entsprechend:

$$j = \pi_{\rho_j} \quad \forall j \in 1, \dots, N \quad (3.3)$$

Um die Objekte in der Form zu platzieren, in der sie angegeben sind, ist die Nebenbedingung „*object()*“ notwendig. Bei der Platzierung von Objekten im Rahmen der in Abschnitt 4.1.2 beschriebenen Variablenauswahl wird zunächst eine ausreichende Zahl an Blöcken eines Objekts ausgewählt und positioniert, durch die die Lage eines Objekts eindeutig bestimmt ist. Dafür werden die Eckpunkte mit den Koordinaten  $p'_{j,o} = (0, 0, 0)$ ,  $p'_{j+l_o-1,o} = (l_o, 0, 0)$ ,  $p'_{j+l_o \cdot b_o-1,o} = (l_o, b_o, 0)$  und  $p'_{j+l_o \cdot b_o \cdot h_o-1,o} = (l_o, b_o, h_o)$  ausgewählt. Auf Basis dieser Eckpunkte werden den restlichen Blöcken eines Objekts ihre Positionen durch Propagieren zugewiesen. Demnach muss die Nebenbedingung „*object()*“ zunächst mögliche Positionen der Blöcke mit den angegebenen Koordinaten in Bezug auf bereits platzierte Eckpunkte eingrenzen. Im Anschluss erfolgt die Platzierung der restlichen Blöcke. Berücksichtigt werden müssen dabei auch die Optionen, ein Objekt um eine horizontale Achse zu drehen oder es zu spiegeln.

#### 3.5.1. Berücksichtigte Nebenbedingungen

Bei den meisten der weiteren zur Lagebestimmung der Blöcke verwendeten Nebenbedingungen, handelt es sich um unäre oder binäre Constraints. Das bedeutet, dass die Bedingungen sich entweder auf ein oder zwei Variablen beziehen. Es handelt sich folglich immer um eine Beschränkung der Lage im Raster oder der relativen Position zueinander.

Des Weiteren kann zwischen Bedingungen unterschieden werden, die eine Anforderung des Benutzers darstellen und solchen, die implizit hinzugefügt werden, beispielsweise für das Formen der Objekte oder zur Unterstützung einer effizienten Suchstrategie.

### Die Constraints *inDomain()*, *inLevel()* und *onPosition()*

Diese Gruppe von unären Nebenbedingungen begrenzt den Wertebereich einer Variablen  $\pi_i$  auf eine oder mehrere Positionen im Raster. Dabei kann es sich bei dem Bereich, der der Bedingung „*inDomain()*“ zugrunde liegt, um eine beliebige Auswahl von Positionen innerhalb des Rasters handeln. So gesehen handelt es sich bei den anderen beiden Nebenbedingungen „*inLevel()*“ und „*onPosition()*“ lediglich um Spezialfälle dieses Constraints, das auf bereits vordefinierte Bereiche (wie zum Beispiel eine bestimmte Ebene des Rasters) zurückgreift. In allen Fällen handelt es sich um Bedingungen, die Anforderungen des Benutzers abbilden sollen. Im Falle von IFMIF ließen sich über diese Constraints beispielsweise voneinander getrennte Sicherheitsbereiche definieren und Objekte diesen Bereichen jeweils zuordnen. Ebenso denkbar wäre der Fall, dass in einem Gebäude technische Einrichtungen wie Heizung, Lüftung und ähnliches im Untergeschoss platziert werden sollen. Dies ließe sich über solche Constraints ebenso erreichen.

### Die Constraints *maxDistance()*, *minDistance()* und *exactDistance()*

Diese binären Constraints dienen dem Festlegen von Abstandsbeziehungen innerhalb des Rasters. Dabei könnte eine Maximalentfernung beispielsweise für die Verteilung von Antennen sinnvoll sein, um eine lückenlose Netzabdeckung zu erreichen. Eine Mindestentfernung könnte andererseits sinnvoll sein, um sich störende Objekte, zum Beispiel aufgrund von Vibrationen, zu trennen. An sich ließe sich aus diesen beiden Bedingungen die Forderung nach einer genauen Entfernung nachbauen. Trotzdem ist es sinnvoller, dafür eine eigene Nebenbedingung zu verwenden, um ein effizienteres Propagieren zu ermöglichen.

### Die Constraints *sameLevel()* und *above()*

Mit diesen Bedingungen kann die relative Position zweier Blöcke zueinander beeinflusst werden. Die Nebenbedingung „*sameLevel()*“ wird verwendet, um zwei Blöcke in einer zwar beliebigen, aber identischen Ebene zu platzieren. Das kann zum Beispiel sinnvoll sein, wenn mehrere Objekte über ein flurgebundenen Materialflusssystem miteinander verbunden werden sollen. Die Bedingung „*above()*“ legt hingegen fest, dass ein Block immer mindestens eine Ebene über dem anderen platziert sein muss. Dies kann in Kombination mit „*sameLevel()*“ beispielsweise sinnvoll sein, um eine aufrechte Positionierung

### 3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen

---

eines Objekts zu erreichen, da ein Objekt theoretisch in allen Orientierungen platziert werden kann.

#### Die Constraints *linear()*, *notLinear()*, *planar()* und *notPlanar()*

Diese Gruppe von Constraints sind für die Ausrichtung von Blöcken zueinander notwendig. Berechnet man die Anzahl  $s_{\pi_h, \pi_i}$  gleicher Koordinatenwerte zweier Blöcke  $h$  und  $i$  über die Formel

$$s_{\pi_h, \pi_i} = 1_{\{x_{\pi_h} = x_{\pi_i}\}} + 1_{\{y_{\pi_h} = y_{\pi_i}\}} + 1_{\{z_{\pi_h} = z_{\pi_i}\}}, \quad (3.4)$$

so sind die Bedingungen bei folgenden Werten von  $s_{\pi_h, \pi_i}$  erfüllt:

$s_{\pi_h, \pi_i}$	<i>linear()</i>	<i>notLinear()</i>	<i>planar()</i>	<i>notPlanar()</i>
0	falsch	wahr	falsch	wahr
1	falsch	wahr	wahr	falsch
2	wahr	falsch	wahr	falsch

Aus dieser Definition folgt, dass beispielsweise „*planar()*“ nicht unbedingt heißen muss, dass zwei Blöcke sich in der gleichen Ebene befinden, sondern lediglich, dass sie eine Koordinate teilen müssen. Insofern kann es sich auch um eine vertikale Ebene handeln, in der sich beide befinden. Das gleiche gilt ebenso für die Bedingung „*linear()*“. Soll eine solche Unterscheidung stattfinden, muss zusätzlich die Bedingung „*sameLevel()*“ verwendet werden.

#### Das Constraint *freeSpace()*

Während der Lösungssuche werden die einzelnen Blöcke und Objekte sukzessiv im Raster positioniert. Um die Anzahl der Suchpfade zu reduzieren, ist es sinnvoll zu prüfen, welche Freiräume oder Freiflächen im Raster noch existieren und welche Objekte an diesen Stellen positioniert werden können.

Der Grundgedanke dieser Bedingung ist in Abbildung 3.6 dargestellt. In diesem Fall ist ein einzelner Block bereits in einem zweidimensionalen Raster auf der Position (2,2) platziert worden. Als nächstes soll in dem gegebenen Raster die mit einem Fragezeichen markierte Ecke des rechts stehenden Objekts positioniert werden. Aufgrund der Größe des Rasters sowie des bereits platzierten Blocks kommen die durchgestrichenen Positionen für eine Positionierung nicht in Frage.

Es kann dabei in drei Schritten vorgegangen werden:

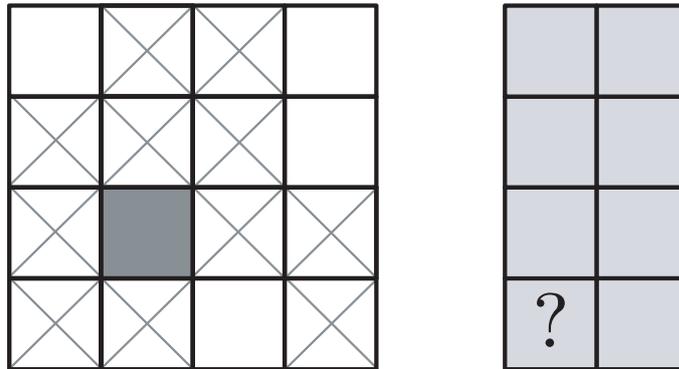


Abbildung 3.6.: Reduktion des Wertebereichs aufgrund der Maße eines Objekts

- Bestimmen, welchen freien Raum das zu einem Block gehörende Objekt mindestens benötigt
- Freie Räume im Raster identifizieren
- Abgleich der Anforderungen und des verfügbaren Raums und Einschränkungen der jeweiligen Domänen.

Da die Objekte feste Maße haben, die sich über die Laufzeit nicht ändern, muss der erste Schritt lediglich einmal zu Beginn der Lösungssuche durchgeführt werden. Relevant für die Frage, ob ein bestimmter Block  $i$  auf einem Rasterplatz positioniert werden kann, sind dabei die Mindestmaße  $x'_{min,i,o}$ ,  $y'_{min,i,o}$  und  $z'_{min,i,o}$ , die dieser jeweils benötigt. Diese ergeben sich dabei gemäß:

$$x'_{min,i,o} = \max \{ |l_o - x'_{i,o} + 1|, x'_{i,o} \} \quad (3.5)$$

Entsprechend werden ebenso die Mindestmaße für die  $y$ - und die  $z$ -Richtung bestimmt. Diese bilden zusammen das Tupel  $p'_{min,i,o} = (x'_{min,i,o}, y'_{min,i,o}, z'_{min,i,o})$ . Da nicht davon ausgegangen werden kann, dass die Ausrichtung der Objektachsen und Rasterachsen miteinander übereinstimmt, gibt das Tupel  $\tilde{p}'_{min,i,o}$  die aufsteigend sortierten Mindestmaße des Blocks  $i$  an. Die Sortierung stellt im Weiteren sicher, dass eine Positionierung eines Blocks auf einer Position nur dann ausgeschlossen wird, falls das zugehörige Objekt selbst in der günstigsten Orientierung nicht platziert werden kann. Als Beispiel zeigen Abbildung 3.7 sowie Tabelle 3.2 die Berechnung der Mindestmaße einiger Blöcke

des angegebenen Objekts.

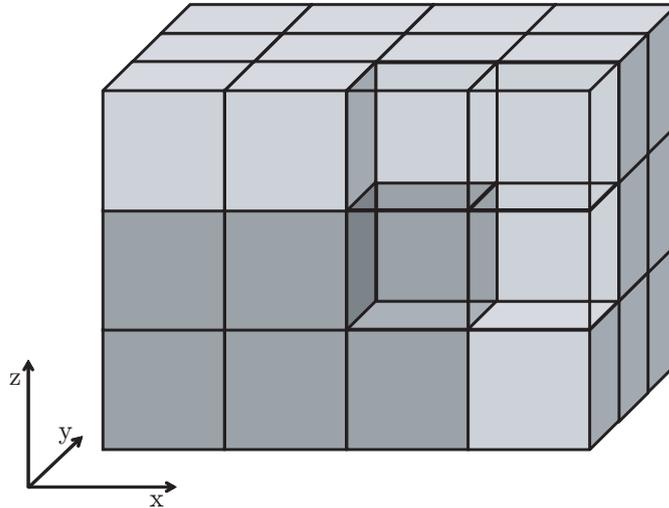


Abbildung 3.7.: Objekt der Größe 4x3x3 mit ausgewählten Blöcken zur Bestimmung von Mindestmaßen

Der zweite Schritt, das Identifizieren freier Räume im Raster, hängt von der aktuellen Zuordnung von Blöcken zu Positionen ab und muss deshalb im Laufe der Lösungssuche wiederholt werden. Um eine Aussage treffen zu können, ob ein Block mit seinen zugehörigen Mindestmaßen einer Position zugeordnet werden kann, müssen zunächst alle freien Räume im Raster bestimmt werden. Da die zu positionierenden Objekte die Form eines Quaders besitzen, sind alle quaderförmigen Freiräume relevant, die in keine Richtung weiter vergrößert werden können. Diese Freiräume sollen im folgenden als *maximale Unterräume* bezeichnet werden. Abbildung 3.8 zeigt für ein 5x4x3 Raster mit bereits einem platzierten Objekt alle maximalen Unterräume.

Wichtig dabei ist, dass tatsächlich alle derartigen Freiräume in einem Iterationsschritt bestimmt werden, da sonst gegebenenfalls Zuordnungsmöglichkeiten ausgeschlossen werden, die jedoch zu einer gültigen Lösung führen würden. Wie in Abbildung 3.8 zu sehen ist, kann auf diese Weise eine Position zu mehreren maximalen Unterräumen gehören. Betrachten wir beispielhaft Position  $p_{43} = (3, 1, 3)$ , so gehört diese zu den Unterräumen A und B. Zu berücksichtigen ist nun allerdings auch die relative Position innerhalb des Unterraums. Wir betrachten diese Position nun auf die gleiche Weise wie es bei den Blöcken

Tabelle 3.2.: Beispielhafte Bestimmung der Mindestmaße innerhalb eines 4x3x3 Objekts

$p'_{i,o}$	$x'_{min,i,o}$	$y'_{min,i,o}$	$z'_{min,i,o}$	$P'_{min,i,o}$	$\tilde{P}'_{min,i,o}$
(1,1,1)	4	3	3	(4,3,3)	(3,3,4)
(2,1,1)	3	3	3	(3,3,3)	(3,3,3)
(3,1,1)	3	3	3	(3,3,3)	(3,3,3)
(1,1,2)	4	3	2	(1,1,2)	(1,1,2)
(2,1,2)	3	3	2	(3,3,2)	(2,3,3)
(3,2,2)	3	2	2	(3,2,2)	(2,2,3)
-					

in Formel 3.5 geschieht. Dabei stellt man fest, dass die Maximalmaße für diese Position  $p_{max_{A,j}} = (3, 4, 1)$  (bzw.  $\tilde{p}_{max_{A,j}} = (1, 3, 4)$ ) für Unterraum A und  $p_{max_{B,j}} = (3, 4, 3)$  (bzw.  $\tilde{p}_{max_{B,j}} = (3, 3, 4)$ ) für Unterraum B betragen. Relevant für die weitere Betrachtung wäre in diesem Fall nur  $\tilde{p}_{max_{B,j}}$ , da dessen Elemente paarweise größer als  $\tilde{p}_{max_{A,j}}$  sind.

Wenn die Mindestanforderungen gemäß der relativen Position der Blöcke innerhalb eines Objekts sowie das maximale Platzangebot innerhalb des Rasters für jede Position bestimmt sind, kann anschließend im dritten Schritt ein Abgleich erfolgen, und die Domänen der Blockvariablen können entsprechend eingeschränkt werden.

Betrachtet man als Beispiel einen Eckpunkt des Objekts aus Abbildung 3.7, stellt man fest, dass es mit seinen Mindestmaßen  $\tilde{p}'_{min,i,o}$  jeweils einen der Eckpunkte des maximalen Unterraums B einnehmen kann. Folglich ließe sich die Domäne des Eckpunkts  $i$  auf  $\pi_i \in \{3, 5, 18, 20, 43, 45, 58, 60\}$  eingrenzen. Die beiden Unterräume A und C kommen von ihren Abmaßen nicht in Frage. Algorithmus 1 zeigt, auf welche Weise beim Propagieren zunächst die Unterräume bestimmt werden, um anschließend mit deren Hilfe die Domänen einzuschränken. Dabei werden, ausgehend von jeder Position, sämtliche maximalen Unterräume erzeugt und anschließend für die betroffenen Variablen gespeichert. Diese lassen sich finden, indem man, ausgehend von einer bestimmten Position, die Richtungsachsen in allen möglichen Reihenfolgen durchläuft. Daraus kann anschließend das maximale Platzangebot für jede Position bestimmt werden und mit dessen Hilfe können die Domänen der Variablen  $\pi$  und  $\rho$  eingeschränkt werden.

Dieses Verfahren kann prinzipiell für sämtliche Variablen durchgeführt werden. Im Vergleich zu einfacheren Nebenbedingungen wie beispielsweise eine

### 3. Layoutplanung unter Berücksichtigung heterogener Raumgrößen

---

```

input :  $\pi, \rho, \tilde{p}'_{min,i,o}$ 
output : Reduzierte Domänen der Variablen  $\pi$  und  $\rho$ 

// Bestimmt die maximalen Unterräume und ordnet sie allen
// Reihenfolgen und Variablen zu, die sie erzeugen
for  $j = 1$  to  $N$  do
    for  $a = 1$  to  $6$  do
        if reihenfolge  $[j][a] = false$  then
            MaxUnterraum dummy = bestimmeUnterraum( $j, a$ );
            foreach block  $\in$  dummy do
                for  $b = 1$  to  $6$  do
                    if reihenfolge  $[j][b] = false$  then
                        if prüfeReihenfolge(dummy,  $b$ ) = true then
                            |_ unterräume  $[j].add$ (dummy);

// Bestimmt die Maximalmaße für die jeweiligen Unterräume
// und sortiert diese
for  $j = 1$  to  $N$  do
    foreach  $u \in$  unterräume  $[j]$  do
        |_  $\tilde{p}_{max_u,j} =$  distanzMax( $u$ );
    sort( $\tilde{p}_{max,j}$ );

// Reduzieren der Domänen der Variablen  $\pi$  und  $\rho$ 
// Existiert kein passender Unterraum für  $i$  wird der Wert
// entfernt
for  $i = 1$  to  $N$  do
    while  $\pi_i.hasNextValue()$  do
        val =  $\pi_i.nextValue()$ ;
        while !(fitsIn( $\tilde{p}'_{min,i,o}, \tilde{p}_{max_u,val}$ )) do
            |_ next  $u$ ;
        if  $u = \tilde{p}_{max_u,val}.end()$  then
            |_  $\pi_i.removeValue$ (val);
            |_  $\rho_{val}.removeValue$ ( $i$ );

```

**Algorithmus 1** : Propagieren der Nebenbedingung *freeSpace()*

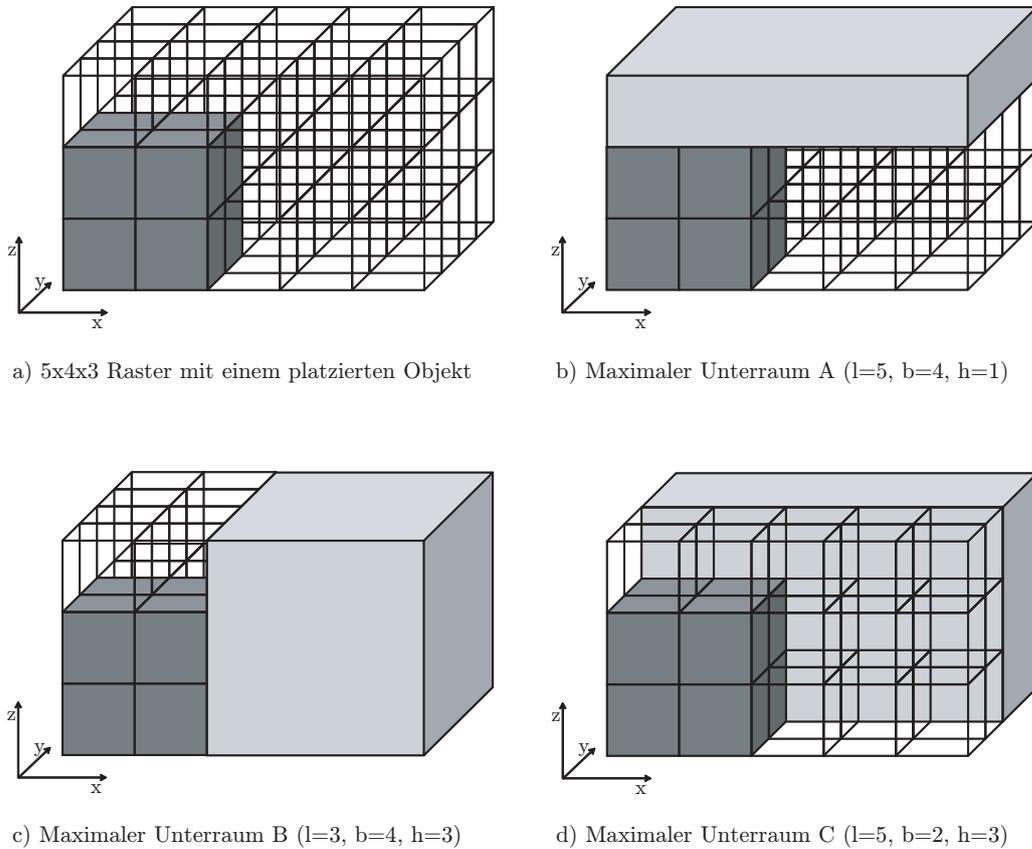


Abbildung 3.8.: Raster mit einem Objekt und allen zugehörigen maximalen Unterräumen

Abstandsrestriktion ist sie jedoch wesentlich aufwändiger zu berechnen. Deshalb ist es sinnvoll, sie nur für ausgewählte Variablen zu berücksichtigen und gegebenenfalls auch nicht unbedingt nach jeder Iteration der Lösungssuche durchzuführen. Insofern ist eine Abstimmung zwischen der Steuerung der Suche über die Variablen- und Wertauswahl, wie sie in Abschnitt 4.1.2 vorgestellt wird, notwendig. Zu beachten ist zudem, dass die Nebenbedingung, da sämtliche Variablen involviert sind, beim Prozess des Propagierens sehr häufig aufgerufen wird. Daher ist es sinnvoll, das Propagieren zunächst aufzuschieben und erst am Ende durchzuführen, wenn über die sonstigen Restriktionen kein weiteres Eingrenzen der Domänen mehr möglich ist (IBM 2012).

Das beschriebene Vorgehen vereinfacht die Situation jedoch insofern, dass

zwei Aspekte vereinfacht werden. Zum einen spielt die Orientierung des Objekts oder auch die Frage, ob ein Spiegeln zulässig ist, keine Rolle. Da sowohl die möglichen Maximaldistanzen einer Position als auch die Mindestdistanzen eines Blocks sortiert werden, wird lediglich geprüft, ob das Objekt auf irgendeine Art und Weise passen könnte, unabhängig davon ob diese Art der Positionierung zulässig wäre. Allerdings vereinfacht dieses Verfahren den Abgleich zwischen Maximalabständen und Minimalabständen, da beide sortiert werden können und in diesem Fall weniger Vergleiche der Distanzen durchgeführt werden müssen. Neben diesem Aspekt wird zudem, ausgehend von einem Block, nur eine Richtung, also entlang oder entgegen einer Achse, geprüft. Für die im Rahmen der Auswahl zuerst betrachteten Eckpunkte eines Objekts ist dies zwar unproblematisch, allerdings könnte auch die Einschränkung des Wertebereichs eines der anderen Blöcke eine Lösung ausschließen und sollte nicht unbedingt vernachlässigt werden.

Aus diesen beiden Gründen soll für dieses Constraint bei der Lösungssuche auch eine alternative Propagierung getestet werden, die die Distanzen in beide Richtungen berücksichtigt und bei der auch die zulässigen Orientierungen eines Objekts beachtet werden. Dies führt allerdings dazu, dass, insbesondere wenn schon eine Reihe von Objekten platziert ist, der Aufwand steigt, da die entsprechenden Mindestdistanzen nicht mehr sortiert vorliegen. Hierfür werden folglich die Mindestabstände in beide Achsenrichtungen verwendet, wobei für die Ausrichtung entlang der Achse der Index  $+$  und für die Ausrichtung entgegen der Achse ein  $-$  ergänzt wird (siehe auch Abbildung 3.9). Obwohl diese Art der Propagierung genauer wäre und die Wertebereiche stärker einschränken würde, kann aufgrund des größeren Rechenaufwands zunächst keine eindeutige Aussage getroffen werden, welches Vorgehen vorzuziehen ist. Dies soll in Kapitel 4 betrachtet werden.

#### **Das Constraint *demandForPosition()***

Durch das Einschränken der Wertebereiche kann der Fall eintreten, dass einzelne Objekte nur noch eine bestimmte Lage im Raster einnehmen können, wenn auch die Orientierung des Objekts noch nicht festgelegt ist. In diesem Fall würde beispielsweise bereits feststehen, dass eine Position durch einen der acht Eckpunkte eingenommen werden muss; andere Blöcke können ausgeschlossen werden. Daher sollte geprüft werden, ob die Nachfrage nach einer Menge an Positionen (die beispielsweise für die Eckpunkte reserviert sind) das Angebot an Positionen übersteigt und ob in diesem Fall ausreichend Ausweichmöglichkeiten für andere Objekte bestehen. Das generelle Vorgehen dazu

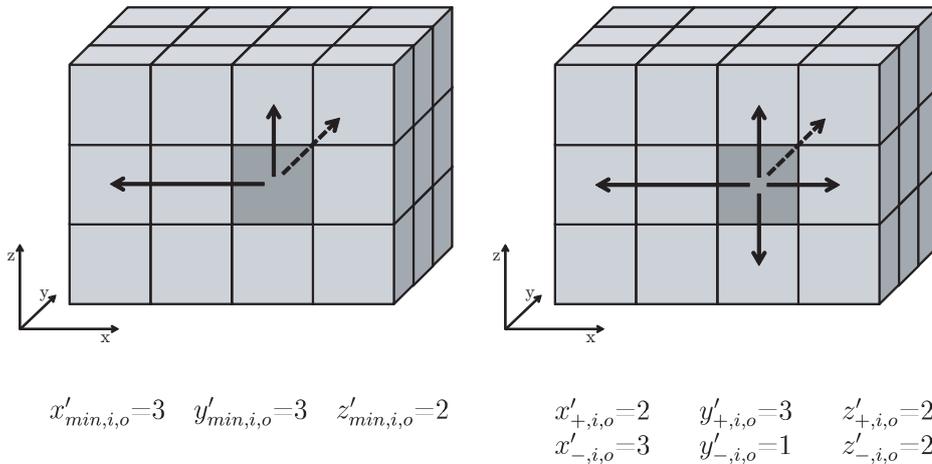


Abbildung 3.9.: Bestimmung der Mindestmaße mit und ohne Berücksichtigung der Orientierung eines Objekts

zeigt Algorithmus 2.

Problematisch bei diesem Vorgehen ist jedoch, dass es nicht zielführend sein kann, wahllos Teilmengen aus Blöcken zu bilden und deren Wertebereiche anschließend auf Überschneidungen zu überprüfen. Deshalb ist es wie bei der vorigen Nebenbedingung notwendig, diesen Ansatz nicht flächendeckend, sondern nur nach einigen Iterationsschritten bei Bedarf durchzuführen. Eine sinnvolle Teilmenge können zum Beispiel die Eckpunkte eines Objekts sein, die im Anschluss an das vorige Constraint „*freeSpace()*“ geprüft werden. Dieses Vorgehen wird äquivalent zu den für das Constraint "*object()*" betrachteten Eckpunkten auch angewandt.

```
input :  $\pi, \rho$ 
output : Reduzierte Domänen der Variablen  $\pi$  und  $\rho$ 
// Bestimme mögliche Positionen einer Teilmenge S
foreach  $S \subseteq \{1, \dots, N\}$  do
  foreach  $i \in S$  do
    while  $\pi_i.hasNextValue()$  do
      if  $(t = \pi_i.nextValue()) \notin T$  then
         $T = T \cap t$ ;
    // Schränke Domänen ein, falls Anzahl Positionen gleich
    // Anzahl Blöcke der Menge S
    if  $|T| < |S|$  then
       $constrainedFailed()$ ;
    if  $|T| = |S|$  then
      foreach  $i \in S$  do
         $\pi_i.removeAllOtherValues(T)$ ;
      foreach  $j \in \{1, \dots, N\} \setminus T$  do
         $\rho_j.removeAllOtherValues(S)$ ;
```

**Algorithmus 2** : Propagieren der Nebenbedingung *demandForPosition()*

## 4. Lösungsverfahren

In diesem Kapitel steht die effiziente Suche nach Lösungen mit niedrigen Materialflusskosten im Vordergrund. Dafür soll in Abschnitt 4.1 zunächst eine Reihe von Ansätzen, insbesondere im Hinblick auf die Variablen- und Wertauswahl, vorgestellt werden. Zudem wird in Abschnitt 4.2 ein Berechnungsverfahren zur Bestimmung einer unteren Schranke des Optimierungsproblems skizziert.

Im Weiteren steht dann die Auswahl passender Lösungsansätze in Abhängigkeit der Eingabeparameter im Vordergrund. Dazu werden einerseits charakteristische Größen der betrachteten Layoutprobleme vorgestellt und die Eignung verschiedener Ansätze in Bezug darauf erläutert.

### 4.1. Mögliche Lösungsstrategien

Für die Modellierung des Problems wurde CP Optimizer von IBM ILOG verwendet. Das Modell wurde zum größten Teil mit der zugehörigen Modellierungssprache OPL geschrieben. Die Logik der komplexeren Nebenbedingungen wurde über die zugehörige C++-Schnittstelle implementiert.

Die Eingabe der Daten erfolgt über eine in Java geschriebene grafische Oberfläche. Die Eingabedaten sind dabei

- die Rahmendaten des Rasters
- die zu platzierenden Objekte
- die gewünschten Nebenbedingungen
- die Fluss- und die Distanzmatrix
- sowie Einstellungen zur Steuerung der Lösungssuche.

Darüber hinaus kann über die Oberfläche der Suchprozess verfolgt und ebenso eine Serie von Berechnungen mit variierenden Suchparametern gestartet werden. Es erfolgt eine Darstellung der sich ergebenden Lösung. Die Oberfläche ist in Abbildung 4.1 dargestellt.

## 4. Lösungsverfahren

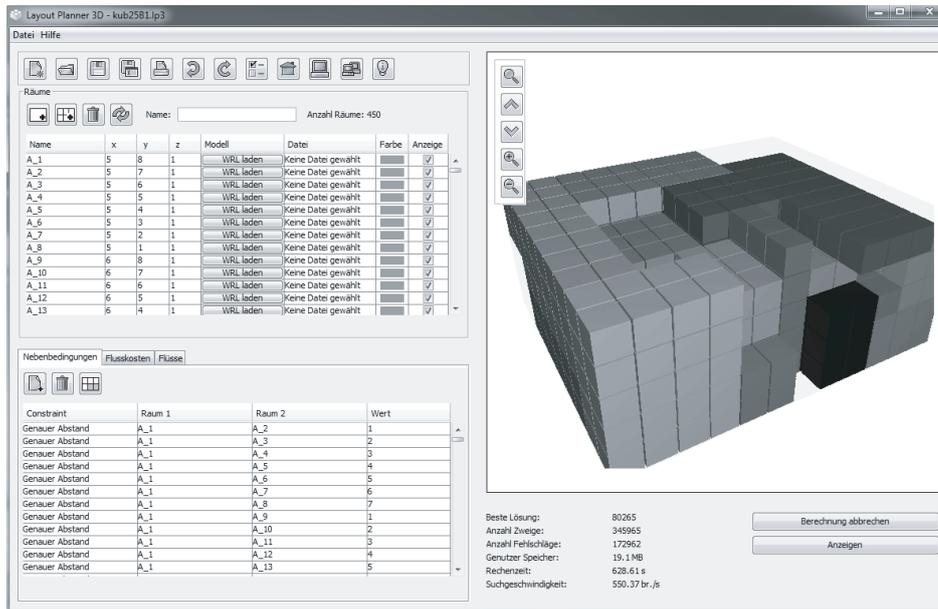


Abbildung 4.1.: Grafische Oberfläche zur Steuerung des Modells

Für die Steuerung der Suche ist es möglich, unterschiedliche Varianten der Variablen- und Wertauswahl zu bestimmen und die Verwendung einiger Nebenbedingungen sowie der unteren Schranke einzuschränken. Neben einem Zeitlimit kann ebenso eine Suchstrategie gewählt werden. Dabei stehen die vom CP Optimizer zur Verfügung gestellten Varianten zur Auswahl. Neben einer Tiefensuche kann auf eine Möglichkeit zurückgegriffen werden, die die Suche nach einer bestimmten Anzahl von Fehlschlägen neu startet. Ein Fehlschlag kann darin bestehen, dass keine weitere Wertebelegung ohne eine Verletzung der Nebenbedingungen mehr möglich ist, oder dass die aktuelle Teillösung bereits höhere Kosten verursacht als eine bereits ermittelte Lösung. Damit kann verhindert werden, dass bei einem frühen Entscheidungsfehler im Suchprozess die Suche in einem bestimmten Zweig stecken bleibt. Diese Möglichkeit wird ebenso in anderen Frameworks außer dem CP Optimizer angeboten und wird in der Literatur auch häufig empfohlen (Refalo (2004), Malapert et al. (2012)).

### 4.1.1. Einflussgrößen der Suchstrategie

Für die Auswahl der passenden Strategie spielen unterschiedliche Faktoren eine Rolle, da je nach Daten andere Aspekte im Vordergrund stehen können. Dabei muss unterschieden werden zwischen der Frage, ob ein bestimmtes Merkmal für die gewählte Suchstrategie überhaupt beachtet werden soll, und der Frage, wie das Merkmal konkret die Entscheidungen während der Suche beeinflusst. Betrachtet man die Objektgröße, so ist für die erste Frage beispielsweise interessant, wie stark sich die Größen der Objekte unterscheiden. Liegt eine eher homogene Größe vor, so ist der Einfluss dieses Merkmals gering. Entsprechend sollte es während der Suche nicht oder nur mit einem geringen Gewicht berücksichtigt werden. Soll das Merkmal dann als Kriterium während der Suche eingesetzt werden, müssen die einzelnen Objekte anhand einer Maßzahl bewertet werden. Diese sollte über alle Merkmale normiert sein, um eine vergleichbare Bewertung aus den unterschiedlichen Kriterien zu bestimmen. Für den Bewertungsvektor  $\vec{v}_a$  des Merkmals  $a$  wird dazu die euklidische Norm herangezogen, so dass aus dem Einheitsvektor

$$\vec{e}_{v_a} = \frac{\vec{v}_a}{\|\vec{v}_a\|}$$

eine normierte Bewertung für alle Objekte im Bereich zwischen 0 und 1 resultiert. Wie in Abschnitt 3.5, wird die Lage eines Objekts über bis zu vier Eckpunkte festgelegt, die eine Position im Raster zugewiesen bekommen. Die Position der Blöcke eines Objekts ist dadurch ebenso festgelegt und wird anschließend propagiert. Da die Entscheidung über die weitere Suche nicht auf Objekt- sondern auf Variablen-Ebene getroffen wird, bekommen alle Variablen eines Objekts die Bewertung des Objekts zugeordnet. Als zweites Kriterium werden anschließend die Eckpunkte nochmal gesondert bewertet, so dass die Positionierung des Objekts über diese abläuft. Ausgehend von den Eingabedaten wurden die folgenden Merkmale, die für eine Steuerung der Suche in Frage kommen, betrachtet.

#### Domänengröße

Ein häufig verwendetes Kriterium für die Auswahl der Variablen ist die Größe der jeweiligen Domäne. Dabei sollen Variablen bevorzugt werden, deren Wertebereich bereits stark eingeschränkt wurde und bei denen ein Scheitern der

Suche im weiteren Verlauf deshalb wahrscheinlich ist. Dieses Merkmal kann sich zum Teil mit der Bewertung der Constraints überschneiden, da auch ein Teil der Nebenbedingungen bereits zu Beginn zu einer Einschränkung des Wertebereichs führt und somit über zwei Kriterien berücksichtigt würde.

### **Objektgröße**

Ein Kriterium für die Variablenauswahl kann die Größe der einzelnen Objekte sein. Dabei ist es sinnvoll, größere Objekte möglichst früh im Suchprozess zu platzieren, da es unwahrscheinlich ist, dass gegen Ende ein entsprechender Freiraum zur Verfügung steht. Als Bewertung eines Blocks  $\pi_i$  wird dafür die Anzahl der Blöcke des zugehörigen Objekts verwendet. Dies entspricht also dem Produkt aus Länge, Breite und Höhe des Objekts.

### **Beschränkungen über Constraints**

Constraints beeinflussen die Zahl der Möglichkeiten, einen Block zu positionieren. Da nicht alle Constraints unmittelbar zu einer Reduktion des Wertebereichs führen, sollte deren Auswirkung bereits zu einem frühen Zeitpunkt berücksichtigt werden. Dies gilt für binäre Constraints wie beispielsweise Abstandsbedingungen oder das Erzwingen linearer oder planarer Anordnungen. Die Bewertung eines Constraints erfolgt über die Anzahl möglicher Positionen der einzelnen Blöcke. Für ein binäres Constraint liegt der Bewertungsrahmen folglich im Bereich zwischen 1 und  $N^2$ . Um pro Objekt eine normierte Bewertung zu erhalten, werden im Anschluss die Bewertungen der Constraints dem jeweiligen Objekt zugerechnet und für dieses jeweils aufsummiert. Der daraus resultierende Bewertungsvektor wird wiederum, wie in Abschnitt 4.1.1 beschrieben, normiert.

### **Bewertung der Materialflüsse**

Die ein- und ausgehenden Materialflüsse eines Objekts können ebenso als Kriterium zur Steuerung der Suche in Betracht gezogen werden. Dabei steht allerdings weniger die Zulässigkeit der Lösung als das möglichst frühe Auffinden guter Lösungen im Vordergrund. Aus diesem Grund sollen Objekte mit hohen ein- und ausgehenden Materialflüssen während der Suche früher platziert werden, um eine günstige Ausrichtung zueinander zu ermöglichen. Entsprechend erfolgt eine Bewertung anhand der Materialflüsse aller zu einem Objekt zugehörigen Variablen.

### 4.1.2. Variablenauswahl

Abhängig von den Eingangsdaten kommen für die Variablenauswahl unterschiedliche Strategien in Betracht. Neben den Standardansätzen der Auswahl nach Domänengröße sowie einer zufälligen Auswahl, werden dabei auch die im vorigen Abschnitt angesprochenen Größen wie eine Auswertung der Constraints oder der Objektgröße berücksichtigt. Dabei ist es auch möglich neben einer Entscheidung, die allein auf einem Kriterium basiert, eine gemischte Strategie zu nutzen, bei der die Faktoren mit unterschiedlichem Gewicht berücksichtigt werden.

Die Suche anhand der Domänengröße ist ein weit verbreiteter Standardansatz und berücksichtigt nicht die Gruppierung der einzelnen Blöcke in Objekten. So kann es durchaus der Fall sein, dass Blöcke aus unterschiedlichen Objekten aufeinanderfolgend platziert werden und sich unter Umständen gegenseitig blockieren können. Praktisch tritt dieser Fall jedoch nur in Ausnahmefällen auf. In der Regel führt das Positionieren eines Blocks dazu, dass ebenso die restlichen Blöcke des Objekts im Anschluss ausgewählt werden, oder über das Propagieren folgen. Die Ursache hierfür ist, dass nach dem Platzieren des ersten Blocks eines Objekts die Wertebereiche der restlichen Blöcke durch die Abstandsbedingungen und die geforderte Ausrichtung der Eckpunkte sehr stark eingeschränkt werden. Das führt im Anschluss dazu, dass diese meistens die nächstkleinere Domäne aufweisen und deshalb im folgenden Schritt ausgewählt werden.

Bei einer zufälligen Variablenauswahl ist dies jedoch nicht der Fall. Die Variablen werden unabhängig von einer Bewertung oder sonstigen Kriterien ausgewählt; auch die Gruppierung in Objekten bleibt dabei unberücksichtigt. Dieser Ansatz wird deshalb nur in wenigen Fällen erfolgreich sein können, beispielsweise wenn die Objekte zum Großteil nur aus einem einzelnen Block bestehen. Alternativ dazu wurde noch eine Variablenauswahl implementiert, bei der das nächste Objekt zufällig ausgewählt wird; dieses wird jedoch anschließend vollständig platziert. Eine zufällige Variablenauswahl kann beispielsweise als zweites Kriterium bei identischen Bewertungen eingesetzt werden. Dann kann bei einer zufälligen Variablenauswahl und gleichzeitig sporadischem Neustarten der Suche das Verfolgen unterschiedlicher Lösungszweige begünstigt werden, falls bei einer festen Strategie keine Lösung gefunden wird.

Bei der Auswahl der Variablen über eine gemischte Strategie werden zunächst die genannten Kriterien, sofern sie berücksichtigt werden sollen, ausgewertet und mit dem gewählten Gewicht zu einer Gesamtbewertung verrechnet. Für alle Kriterien gilt, dass die Bewertung der einzelnen Variablen eines Objekts

zusammengerechnet werden und dem Objekt als ganzes zugeordnet werden. Demnach besitzen alle Variablen eines Objekts die gleiche Bewertung. Da die Positionierung der Objekte über bis zu vier Eckpunkte erfolgen soll, erhalten diese eine höhere Bewertung als die restlichen Variablen, um bevorzugt ausgewählt zu werden. Es wird darauf geachtet, dass die Bewertung dieser Eckpunkte nicht über derjenigen anderer Objekte liegt. Daraus folgt, dass bei einer Sortierung alle Variablen eines Objekts blockweise gruppiert bleiben.

Das Kriterium der Domänengröße kann innerhalb der gemischten Strategie ebenso berücksichtigt werden. Da aus einer eingeschränkten Domäne eines einzelnen Blocks eine ähnliche Einschränkung für das gesamte Objekt folgt, werden alle Variablen eines Objekts mit der jeweils kleinsten Domäne des Objekts bewertet. Dies ist auch sinnvoll, um eine einheitliche Bewertung eines Objekts zu erreichen und dessen Variablen bei der Auswahl gruppiert zu berücksichtigen.

### 4.1.3. Wertauswahl

Auch für die Wertauswahl werden je nach der Charakteristik der Eingangsdaten des Problems unterschiedliche Ansätze verfolgt. Daneben existieren wie für die Variablenauswahl auch in diesem Bereich Standardansätze, die in Kapitel 5 ebenfalls untersucht werden sollen. Dabei werden die Auswahl des kleinstmöglichen Wertes als auch eine zufällige Wertauswahl betrachtet. Die Zuweisung des kleinstmöglichen Wertes füllt das Raster, ausgehend vom Ursprung, nach und nach auf. Wird dieses Verfahren mit einer Variablenauswahl nach der Höhe der ein- und ausgehenden Flüsse kombiniert, führt dieses offensichtlich zu verhältnismäßig schlechten Lösungen, da Objekte mit vielen Materialflüssen eine ungünstige Position erhalten. Auf der anderen Seite wird ein relativ kompaktes Layout erzeugt, was bei einem hohen Füllgrad des Rasters von Vorteil sein kann. Eine zufällige Wertauswahl führt bei heterogenen Objektgrößen zu Schnitten im Layout, die die weitere Positionierung von Objekten erschwert. Auf der anderen Seite kann es zu Situationen kommen, in denen eine zufällige Auswahl hilft, falls die Suche in einem Lösungszweig feststeckt, der gegebenenfalls nicht erfolgreich ist (IBM 2012). Drei weitere Strategien zur Wertauswahl sollen im Folgenden näher erläutert werden.

## Äquivalenz von Variablen- und Rasterbewertung

Ein für die Wertauswahl entwickelter Ansatz ist das Erreichen einer möglichst ähnlichen Bewertung von Positionen im Raster und Bewertungen von Variablen. Ziel dabei ist, Objekten mit einer hohen Bewertung, die zum Beispiel viele ein- und ausgehende Flüsse aufweisen, eine verhältnismäßig gute Position zuzuordnen. Entsprechend sollen Objekte mit schlechter Bewertung eher schlechtere Positionen (beispielsweise am Rand) erhalten. Dazu ist es zunächst notwendig, für die Variablen und Rasterpositionen jeweils eine Bewertung durchzuführen und diese bei der Wertauswahl der Variablen anschließend miteinander zu vergleichen.

Für die Bewertung der Variablen wurden dabei zwei Aspekte betrachtet. Zum einen sollten Objekte mit vielen Flüssen gut erreichbar sein, zum anderen behindern zentral platzierte große Objekte unter Umständen die Positionierung weiterer Objekte. Demnach wurde für die Bewertung der einzelnen Blöcke die Flussbewertung sowie die Bewertung der Objektgröße berücksichtigt. Beide Bewertungen sind (abhängig von der gewählten Strategie) bereits im Rahmen der Variablenauswahl ermittelt worden. Da eine hohe Bewertung der Objektgröße nun allerdings zu einem tendenziell eher schlechteren Platz führen soll, muss die Bewertung umgekehrt werden. Da diese zwischen 0 und 1 liegt, werden alle Werte jeweils von 1 abgezogen und der sich ergebende Vektor wiederum normiert um mit der Bewertung der Flüsse vergleichbar zu sein. Die beiden Bewertungen können für die endgültige Wertauswahl mit einer Gewichtung berücksichtigt werden. Bei dieser Gewichtung sollte beispielsweise die relative Größe eines Objekts in Bezug auf das gesamte Raster berücksichtigt werden. Eine Variation dieses Gewichtungsfaktors wird in Kapitel 5 betrachtet.

Die Positionen des Rasters werden anhand ihrer Lage innerhalb des Rasters bewertet. Die Bewertung erfolgt je Position  $j$  gemäß

$$L + B + H - |2x_j - L| - |2y_j - B| - |2z_j - H|.$$

Die Bewertung des Rasters sowie der Objekte ist nochmal in Abbildung 4.2 veranschaulicht. Die Bewertung einer Position anhand der Lage ist in der Abbildung links veranschaulicht. Zentrale Positionen haben durchschnittlich kürzere Wege und damit eine hohe Bewertung. Diese Positionen sind in der Abbildung dunkler dargestellt.

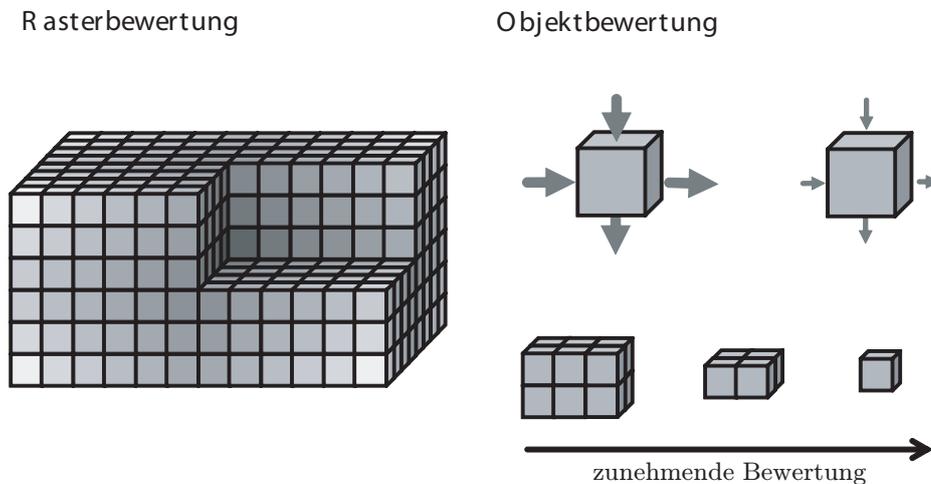


Abbildung 4.2.: Bewertung des Rasters sowie Bewertung der Objekte anhand der Merkmale Flüsse und Objektgröße

Nach der Bewertung muss noch eine Zuordnung zwischen Rasterpositionen und Objekten beziehungsweise deren Variablen getroffen werden. Dazu werden beide Bewertungen aufsteigend sortiert und anschließend einander zugeordnet. Auf diese Weise erhält jeder Block eine Idealposition im Raster, auf der er bei der Wertauswahl platziert werden sollte. Falls eine Position mit dieser Bewertung nicht mehr frei ist oder außerhalb der Domäne der Variablen liegt, wird die Position gewählt, deren Bewertung möglichst nah an diesem Ziel ist.

#### **Konstruktive Wertauswahl auf Basis bereits positionierter Objekte**

Bei diesem Ansatz steht weniger das Erreichen eines flussminimalen Layouts im Vordergrund, als vielmehr das Finden einer zulässigen Lösung. Daher soll dieses Verfahren primär eingesetzt werden, falls der Füllgrad des Rasters hoch ist und gleichzeitig sehr unterschiedliche Objektgrößen berücksichtigt werden müssen. In diesen Fällen kann es schwierig sein, die vorhandenen Objekte innerhalb des Rasters zu positionieren.

Daher soll bei dieser Form der Wertauswahl eine kompakte Form der Objekte erreicht werden, bei der möglichst wenige Lücken entstehen, die die weitere Suche erschweren würden.

Zu diesem Zweck wird der Wertebereich einer Variablen dahingehend analysiert, wie gut das zu der gewählten Variablen gehörende Objekt an bereits platzierten Einheiten ausgerichtet ist. Es sollen daher Platzierungen bevorzugt werden, die innerhalb des Layouts zu möglichst wenigen Kanten führen. Auf diese Weise sollen größere Blöcke mit gemeinsamen Maßen gebildet werden, an denen sich weitere Objekte leichter ausrichten lassen. Dabei werden sowohl Hindernisse berücksichtigt (das können sowohl die Grenzen des Rasters als auch andere Objekte sein) als auch die Grenzen von Objekten in der näheren Umgebung. Eine positive Bewertung bedeutet dabei, dass die zur Verfügung stehenden Maße, denen des zu platzierenden Objekts entsprechen.

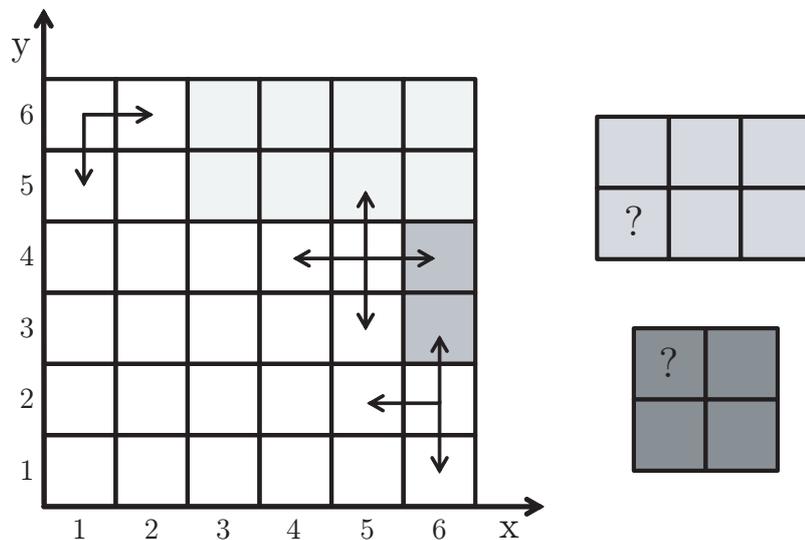


Abbildung 4.3.: Bewertung einzelner Rasterpositionen anhand der Äquivalenz von Maßen

Die Bewertung soll kurz anhand des in Abbildung 4.3 dargestellten Beispiels erläutert werden. In dem dargestellten 6x6 Raster sind bereits zwei Objekte platziert. Für die beiden markierten Blöcke der rechts dargestellten Objekte sollen nun die drei Positionen  $(1, 6)$ ,  $(5, 4)$  und  $(6, 2)$  bewertet werden. Betrachten wir zunächst das größere der beiden Objekte. Ausgehend von der ersten zu platzierenden Variablen werden zwei freie Positionen in Längs-Richtung und eine freie Position in Quer-Richtung benötigt. Entgegen dieser Richtungen ist der Platzbedarf jeweils null. Eine Platzierung der Variable auf Position  $(1, 6)$  ist nur bei einer Drehung des Objekts um  $90^\circ$  im Uhrzeigersinn möglich; das Objekt erstreckt sich dann auf den Positionen im Bereich

$x \in \{1, 2\}$  und  $y \in \{4, 5, 6\}$ . Ausgehend von der betrachteten Position endet die kürzere Seite des Objekts genau an dem angrenzenden Objekt auf den Positionen ( $x \in \{3, \dots, 6\}$  und  $y \in \{5, 6\}$ ). Ebenso positiv würde in die Bewertung eingehen, dass auch der Platzbedarf entgegen der Längs- und Querrichtung des Objekts dem Angebot im Raster, nämlich jeweils null Einheiten entspricht.

Günstiger als diese Position wäre hingegen noch Position (5, 4). In diesem Fall würde das platzierte Objekt in Längs- und Querrichtung zwar nicht durch ein anderes Objekt begrenzt, allerdings würde es bestehende Kanten schließen, da es bündig mit den bereits platzierten Objekten abschließen würde. Das Verfahren wurde so implementiert, dass die Ausrichtung an anderen Objekten stärker gewichtet wird als eine entsprechende Begrenzung durch das Raster. Der Grund dafür ist, dass im Zweifelsfall eher eine Ausrichtung an einem bestehenden Objekt erfolgen soll, als einen neuen Bereich zu beginnen. Im Extremfall könnte dies andernfalls dazu führen, dass den ersten Objekten jeweils eine Eckposition im Raster zugewiesen wird. Im konkreten Beispiel würde das obere Objekt idealerweise im Bereich  $x \in \{3, 4, 5\}$  und  $y \in \{3, 4\}$  platziert werden. Bei einer Entscheidung für das untere Objekt sollte der Bereich  $x \in \{1, 2\}$  und  $y \in \{5, 6\}$  gewählt werden. Zu beachten ist hierbei, dass die Bewertung für die Objekte nicht simultan erfolgt. Sind für ein Objekt auf einer Position mehrere Ausrichtungen möglich, wird als Bewertung das Maximum der unterschiedlichen Möglichkeiten verwendet.

#### Gemischte Wertauswahl

Wie bei der Variablenauswahl können auch bei der Wertauswahl mehrere, gewichtete Kriterien als Entscheidungsgrundlage verwendet werden. Neben den beiden vorgestellten Varianten steht darüber hinaus noch die Möglichkeit zur Verfügung, einem Block immer den zu diesem Zeitpunkt optimalen Platz zuzuweisen; das heißt, die Position die unter Berücksichtigung der bereits platzierten Objekte die geringsten Flusskosten aufweist.

Eine Mischung der Kriterien kann beispielsweise Sinn machen, wenn andernfalls keine eindeutige Entscheidung getroffen werden kann. Betrachtet man eine Variablenauswahl anhand der ein- und ausgehenden Flüsse einer Variablen und wendet darauf eine flussoptimale Wertauswahl an, so kann zu Beginn keine Entscheidung getroffen werden, da alle Positionen eine identische Bewertung hätten. Gemäß der Implementierung der Entscheidungsregel würde in diesem Fall bei der Position mit dem kleinsten Index begonnen. Dies würde allerdings bedeuten, dass ein Block mit hoher Flussdichte eine ungün-

stige Position zugewiesen bekommt. Wird die Äquivalenz von Variablen- und Rasterbewertung als zusätzliches Kriterium mit berücksichtigt, könnte dies vermieden werden.

## 4.2. Bestimmung unterer Schranken

Untere Schranken eines Optimierungsproblems dienen dem Beweis der Optimalität einer Lösung. Dabei wird eine Schranke in der Regel anhand von zwei Kriterien bewertet. Zunächst sollte eine untere Schranke möglichst dicht an der optimalen Lösung liegen. Dadurch können suboptimale Lösungen möglichst früh ausgeschlossen werden und die Lösungszeit verkürzt sich entsprechend. Zum anderen muss betrachtet werden, welcher Aufwand zur Berechnung der Schranke notwendig ist.

Im vorliegenden Fall muss beachtet werden, dass je nach Problemstellung das Finden einer zulässigen Lösung bereits relativ schwierig sein kann und in diesem Fall gegenüber der Optimalität im Vordergrund steht. Deshalb wäre es in diesen Fällen nicht wünschenswert, wenn eine Lösung, die die Constraints erfüllt, aufgrund der unteren Schranke nicht gefunden wird. In diesem Fall widerspricht die Forderung nach der Dichtheit der Schranke offensichtlich dem Ziel, eine zulässige Lösung zu finden.

Über das in Kapitel 3 formulierte Modell ist es möglich, eine Vielzahl unterschiedlicher Problemstellungen abzubilden. Dabei können unterschiedliche Parameter variieren, die auch auf die Auswahl einer unteren Schranke Einfluss haben. Dies kann die Frage sein, wie stark unterschiedlich die Größen der einzelnen Objekte sind, wie viele zusätzliche Nebenbedingungen existieren oder auch die Frage, wie groß die Varianz der Materialflüsse ist und ob eine Reihe der Objekte gar keine Flüsse aufweisen.

Da der Fokus bei der Problemlösung stärker auf dem Finden zulässiger Lösungen steht, spielt die untere Schranke eine eher untergeordnete Rolle, die bei nur schwach restringierten Problemen die Optimalität schneller zeigen kann.

Untere Schranken stellen im Bereich der mathematischen Optimierung Lösungen eines relaxierten Problems dar. Dabei werden beispielsweise Ganzzahligkeitsbedingungen vernachlässigt, um die Problemlösung zu beschleunigen. Problematisch im vorliegenden Fall ist die Vielzahl der Nebenbedingungen und die unterschiedliche Form der einzelnen Objekte. Ein relaxiertes Modell, das wie das Ursprungsproblem auf CP basiert, ist natürlich nicht sinnvoll, da in diesem Bereich Ganzzahligkeit eine explizite Forderung ist und die Methodik stärker auf das Auffinden zulässiger Lösungen als optimaler Lösungen

ausgelegt ist. Aus diesem Grund ist es sinnvoll, die untere Schranke über ein lineares Problem zu bestimmen. Dabei ist es allerdings nicht möglich, Nebenbedingungen bezüglich einer linearen Ausrichtung oder ähnlichem zu berücksichtigen. Entsprechend wird die Qualität der unteren Schranke in diesem Fall allerdings sinken. Vor diesem Hintergrund sollen zunächst einige existierende Schranken des quadratischen Zuordnungsproblems betrachtet werden.

### 4.2.1. Untere Schranken des quadratischen Zuordnungsproblems

Für das quadratische Zuordnungsproblem existiert neben einer Vielzahl alternativer Problemformulierungen auch eine Reihe unterschiedlicher Berechnungsweisen für untere Schranken. Ein Überblick über diese Ansätze ist beispielsweise bei Li et al. (1994), Pardalos et al. (1994), Commander (2003) oder Loiola et al. (2007) zu finden. Unterschieden werden dabei häufig die Gilmore-Lawler-Schranke und darauf aufbauende Ansätze, eigenwertbasierte Schranken und Ansätze über alternative Problemformulierungen. Im Folgenden werden zunächst diese Ansätze mit ihren Eigenschaften kurz vorgestellt. Im Anschluss erfolgt die Auswahl der für das Modell verwendeten Schranke und die notwendigen Anpassungen werden erläutert.

#### Gilmore-Lawler-Schranke

Die sogenannte Gilmore-Lawler-Schranke wurde Anfang der 60er Jahre entwickelt und wird bis heute in unterschiedlichen Variationen verwendet sowie ebenso als Vergleichswert für andere Schranken des quadratischen Zuordnungsproblems verwendet. Die Schranke beruht auf einem linearen Zuordnungsproblem mit der Matrix  $K$ , wobei diese Matrix auf der Kostenmatrix  $G$  und der Flussmatrix  $F$  des ursprünglichen quadratischen Zuordnungsproblems beruht (Pardalos et al. 1994). Damit ergibt sich die Schranke nach Gilmore (1962) und Lawler (1963) durch folgende Definition:

**Definition 4** *Es kennzeichne  $f_i$  die  $i$ -te Zeile der Flussmatrix  $F$  und  $g_i$  die  $i$ -te Zeile der Kostenmatrix  $G$ . Ferner seien  $\hat{f}_i$  und  $\hat{g}_i$  die Zeilen der Matrizen  $F$  und  $G$  ohne die Einträge  $f_{ii}$  beziehungsweise  $g_{ii}$ . Mit dem minimalen Vektorprodukt  $\langle \hat{f}_i, \hat{g}_i \rangle_-$  ergeben sich die Matrixeinträge von  $K$  mit*

$$k_{ij} = f_{ii}g_{jj} + \langle \hat{f}_i, \hat{g}_i \rangle_-, \quad i, j = 1, \dots, n.$$

Die Gilmore-Lawler-Schranke (GLB) ist dann das Minimum des linearen Zuordnungsproblems:

$$GLB(F, G) = \min_{\pi \in \Pi} \sum_{i=1}^n k_{i\pi(i)}$$

Das minimale Vektorprodukt  $\langle \hat{f}_i, \hat{g}_i \rangle_-$  ergibt sich durch das Skalarprodukt des aufsteigend sortierten Vektors  $\hat{f}_i$  und des absteigend sortierten Vektors  $\hat{g}_i$ . Die Einträge  $k_{ij}$  der Matrix  $K$  können so interpretiert werden, dass  $k_{ij}$  den geringst möglichen Flusskosten für Objekt  $i$  entspricht, falls dieses der Position  $j$  zugeordnet wird. Durch das Sortieren der Vektoren wird implizit eine Zuordnung der restlichen Objekte zu Positionen vorgenommen, die jedoch mit den anderen Einträgen der Matrix nicht konsistent ist. Das lineare Problem sucht anschließend eine Permutation, die über die Matrix  $K$  zu den geringsten Gesamtkosten führt.

Die Bestimmung der Matrix  $K$  erfordert dabei einen Aufwand von  $O(n^2 \log(n))$ , die Berechnung des linearen Problems einen Aufwand von  $O(n^3)$ . Insgesamt ergibt sich somit eine Komplexität von  $O(n^3)$  (Xia 2008). Die Gilmore-Lawler-Schranke lässt sich somit vergleichsweise effizient bestimmen; der Abstand zur optimalen Lösung ist allerdings oftmals groß.

### Eigenwertbasierte Schranken

Voraussetzung für die Anwendung eigenwertbasierter Schranken sind symmetrische Fluss- und Kostenmatrizen. Eine erstmalige Formulierung findet sich bei Finke et al. (1987). Diese basiert auf der sogenannten Spur-Formulierung, die Ende der 70er Jahre von C.S. Edwards entwickelt wurde.<sup>6</sup> Die durch Finke et al. (1987) vorgeschlagene Eigenwertschranke ist definiert durch:

**Definition 5** *Es sei  $F$  die Fluss- und  $G$  die Kostenmatrix eines quadratischen Zuordnungsproblems und  $\lambda$  und  $\mu$  kennzeichnen die aufsteigend sortierten Eigenwerte der Matrizen  $F$  und  $G$ . Dann gilt für eine beliebige Permutation  $\pi \in \Pi$  die Ungleichung*

$$\sum_{i=1}^n \lambda_i \mu_{n-i+1} \leq \sum_{i=1}^n \sum_{j=1}^n f_{ij} g_{\pi(i)\pi(j)} \leq \sum_{i=1}^n \lambda_i \mu_i,$$

und die Eigenwertschranke  $EVB(F, G)$  ist gegeben durch

---

<sup>6</sup>Auf diese Art der Formulierung soll hier nicht näher eingegangen werden; sie ist beispielsweise bei Loiola et al. (2007) zu finden.

$$\sum_{i=1}^n \lambda_i \mu_{n-i+1}.$$

Neben dieser Schranke existiert eine Vielzahl weiterer eigenwertbasierter Schranken, bei denen die Ausgangsmatrizen weiteren Rechenoperationen unterzogen werden, bevor die Eigenwerte jeweils bestimmt werden (Li et al. 1994). Der Rechenaufwand zur Bestimmung dieser Schranke beträgt je nach Verfahren im besten Fall  $O(n^3)$  für die Bestimmung der Eigenwerte. Da die anschließenden Operationen (Sortierung der Vektoren und Bilden des Skalarprodukts) eine geringere Komplexität aufweisen, ist dies auch der zeitbestimmende Schritt. Schranken, die nach diesem Verfahren berechnet werden, liegen in der Regel näher am Optimum als die Gilmore-Lawler-Schranke (Anstreicher und Brixius 2001). Unter Umständen können die Werte jedoch auch negativ sein. Dies wird begünstigt durch eine hohe Varianz der Eigenwerte, was wiederum aus stark schwankenden Flüssen zwischen den einzelnen Objekten resultiert. Weitergehende Ansätze vermeiden dieses Problem; sie sind jedoch komplexer zu berechnen und sind nach Burkard et al. (1998) für Branch-and-Bound-Verfahren ungeeignet.

#### **Schranken, basierend auf alternativen Problemformulierungen**

Neben den beiden bereits genannten Gruppen von unteren Schranken existiert noch eine Vielzahl von alternativen Formulierungen und Relaxierungen, die eine untere Schranke für das quadratische Zuordnungsproblem liefern (zum Beispiel Hahn und Grant (1998), Adams et al. (2007), Povh und Rendl (2009) oder Hahn et al. (2012)). Da diese Ansätze zum Teil sehr unterschiedlich sind, lassen sich wenige allgemeine Aussagen über sie treffen. In der Regel liegen die Berechnungszeiten jedoch über denen der Gilmore-Lawler-Schranke oder der beschriebenen Eigenwertschranke (Pardalos et al. 1994); es lassen sich andererseits jedoch auch bessere Ergebnisse erzielen.

#### **4.2.2. Untere Schranke für das vorliegende Modell**

Für das gegebene Modell ist es sinnvoll, eine Schranke zu wählen, die sich effizient berechnen lässt, um auch eine Abschätzung von einzelnen Zweigen während der Lösungssuche zu ermöglichen. Dies bietet darüber hinaus den Vorteil, dass ein Teil der Nebenbedingungen zumindest nach einer Neuberechnung der Schranke in diesem Wert abgebildet ist. Wie bereits erwähnt, könnte es sinnvoll sein, je nach Eingabedaten des Modells, eine Auswahl bezüglich der Schranke zu treffen. Da der Fokus jedoch auf der Art und Weise

der Suche nach zulässigen Lösungen liegen soll, ist es sinnvoll eine Schranke zu wählen, die sich effizient berechnen und nach Möglichkeit in einem gewissen Rahmen an das spezielle Problem anpassen lässt.

Aus diesem Grund sind Schranken über alternative Formulierungen aufgrund ihrer in der Regel längeren Rechenzeit eher ungeeignet. Des Weiteren besteht das Problem, dass gerade bei großen Instanzen eine linearisierte Form einen hohen Speicherbedarf für Nebenbedingungen und Kostenmatrix aufweist.<sup>7</sup>

Gegen eine Schranke über die Berechnung der Eigenwerte der Matrizen spricht das Verhalten der Qualität bei schwankenden Flussgrößen. Da mit solchen Schwankungen durchaus gerechnet werden sollte, ist diese Art von Schranke eher ungeeignet. Bei weiterführenden Ansätzen, die dieses Problem beheben, ist die Rechenzeit für die Anwendung unzureichend.

```
// Erzeugen der Vektoren  $\hat{f}_i$  und  $\hat{g}_i$ , sowie Eintragen
// fester Abstände in die Matrix  $K$ 
foreach  $h \in \{1, \dots, N\}$  do
  foreach  $i \in \{1, \dots, N\}$  do
    if inSameObject(i, j) then
      foreach  $s \in \{1, \dots, N\}$  do
         $k[h][s] += t_{h,i} \cdot \text{distance}(h, i);$ 
        speicher [h][s].push(distance(h, i));
      else if  $\pi_h$ .isFixed() then
        foreach  $s \in \{1, \dots, N\}$  do
           $k[h][s] += t_{h,i} \cdot d_{\pi_j, s};$ 
          speicher [h][s].push( $d_{\pi_j, s}$ );
        else
           $\hat{f}_i \cdot \text{push}(t_{h,i});$ 
           $\hat{g}_i \cdot \text{push}(d_{h,i});$ 
```

<sup>7</sup>Beispielsweise besteht die in Abschnitt 2.3.1 dargestellte lineare Form des quadratischen Zuordnungsproblems aus  $n^4 + n^2$  Variablen,  $n^4 + 2 \cdot n + 1$  Nebenbedingungen sowie, entsprechend der Zahl der Variablen,  $n^4 + n^2$  Binärbedingungen.

```

// Vektoren  $\hat{f}_i$ ,  $\hat{g}_i$  und  $\text{speicher}[h][s]$  aufsteigend sortieren
foreach  $i \in \{1, \dots, N\}$  do
    sort( $\hat{f}_i$ );
    sort( $\hat{g}_i$ );
    foreach  $j \in \{1, \dots, N\}$  do
        sort( $\text{speicher}[i][j]$ );
// Matrix  $K$  erzeugen
foreach  $i, j \in \{1, \dots, N\}$  do
    if  $\pi_i.\text{isFixed}()$  then
         $k[i][j] = M$ ;
    index =  $\hat{f}_i.\text{end}()$ ;
    foreach  $s \in \hat{g}_i$  do
        if  $\text{speicher}[i][\pi_i].\text{begin}() == \hat{g}[\pi_i][s]$  then
             $\text{speicher}[i][\pi_i].\text{remove}(\text{speicher}[i][\pi_i].\text{begin}());$ 
        else
             $k[i][\pi_i] + = \hat{g}[\pi_i][s] \cdot \hat{f}[i][\text{index} - -];$ 
    else
        if  $\pi_j.\text{isFixed}()$  then
             $k[i][j] = M$ ;
        else
            index =  $\hat{f}_i.\text{end}()$ ;
            foreach  $s \in \hat{g}_i$  do
                if  $\text{speicher}[h][i].\text{begin}() == \hat{g}[i][s]$  then
                     $\text{speicher}[h][i].\text{remove}(\text{speicher}[h][i].\text{begin}());$ 
                else
                     $k[h][\pi_h] + = \hat{g}[i][s] \cdot \hat{f}[h][\text{index} - -];$ 
                    if index == -1 then
                         $k[h][i] = M$ ;
foreach  $h, i \in \{1, \dots, N\}$  do
    if  $\text{speicher}[h][i].\text{end}() \neq 0$  then
         $k[h][i] = M$ ;

```

**Algorithmus 3 :** Bestimmung der Eingangsdaten zur Berechnung der angepassten Gilmore-Lawler-Schranke

### 4.3. Eignung von Modell und Lösungsstrategie in Bezug auf die Anforderungen

---

Aus diesem Grund wurde eine angepasste Form der Gilmore-Lawler-Schranke für die Suche verwendet. Dabei wird berücksichtigt, dass, ausgehend von einem bestimmten Block, andere Blöcke feste Abstände besitzen, die eingehalten werden müssen. Entsprechend werden diese Distanzen mit ihren entsprechenden Transportmengen aus den Vektoren vor dem Sortieren herausgenommen. Sollte die Menge der geforderten festen Abstände, ausgehend von einer bestimmten Position, nicht erreicht werden können, wird für die entsprechenden Kosten  $k_{ij}$  ein hinreichend hoher Wert  $M$  angenommen, der eine Zuordnung des Blocks auf diese Position verhindert. Das Vorgehen für die Berechnung der Eingangsdaten des linearen Problems ist in Algorithmus 3 skizziert. Dabei werden in dem Datenelement „speicher[ ]“ jeweils die innerhalb eines Objekts vorgegebenen Distanzen gespeichert. Gleiches gilt für bereits festgesetzte Entfernungen zu bereits platzierten Blöcken. Nach dem Sortieren werden die Skalarprodukte aus Entfernungsvektoren und Flussvektoren gebildet. Dabei werden umgekehrt Einträge der Vektoren ausgelassen, die bereits über den Speicher berücksichtigt wurden. Falls der Speicher für einen Eintrag der Matrix  $K$  nicht vollständig geleert wird, ist eine solche Belegung nicht möglich und der Eintrag wird auf den Wert  $M$  gesetzt.

### 4.3. Eignung von Modell und Lösungsstrategie in Bezug auf die Anforderungen

Im Folgenden sollen nochmal kurz die Hauptanforderungen des Modells und die entwickelten Lösungsansätze gegenübergestellt werden. Da in Abhängigkeit der Eingangsdaten unterschiedliche Modellanforderungen im Vordergrund stehen können, mussten verschiedene Ansätze verwendet werden, die sich zum Teil auch widersprechen. Somit kommt der Entscheidung für eine konkrete Lösungsstrategie eine hohe Bedeutung zu. Diese Entscheidung steht dabei im Spannungsfeld der unterschiedlichen Anforderungen, die dabei gegeneinander abgewogen werden müssen.

Dazu seien zunächst kurz nochmal die wesentlichen Eigenschaften nach Abschnitt 3.2.2 genannt:

- Heterogene Raumgrößen und -formen
- Schwankende Materialflüsse, die an unterschiedlichen Punkten eines Objekts anschließen
- Technische Nebenbedingungen
- Charakteristik von Transportmitteln

Der letzte Punkt wurde in Form spezifischer Nebenbedingungen, die beispielsweise eine lineare Anordnung bedingen, abgebildet. Der Aspekt heterogener Raumgrößen und unterschiedlicher Formen kann über das zugrunde liegende Raster erfasst werden. Weitere Raumformen können durch das Verbinden mehrerer Objekte über Nebenbedingungen erreicht werden. Zur Abbildung technischer Randbedingungen wurde eine Reihe von Constraints definiert, die für die Modellbildung genutzt werden können.

Im Hinblick auf die Lösungsansätze wurden mehrere Strategien entwickelt, die auf unterschiedliche Eingangsdaten ausgerichtet sind.

Die schwankenden Raumgrößen finden dabei in mehreren der Ansätze Berücksichtigung. Einerseits werden durch die implementierten Constraints die Wertebereiche der Variablen soweit eingegrenzt, dass Objekte nur in den jeweils passenden Freiräumen platziert werden können. Zudem wird, sollte für ein Objekt nur noch ein Freiraum in Frage kommen, dieser für dieses Objekt reserviert. Für diese beiden Einschränkungen sind die Constraints *freeSpace()* und *demandForPosition()* verantwortlich.

Darüber hinaus kann die Objektgröße sowohl bei der Variablen- als auch bei der Wertauswahl berücksichtigt werden. Dabei wird jeweils darauf geachtet, dass die Platzierung großer Objekte möglichst geringe Auswirkungen hat. Dies kann dabei einerseits durch eine frühe Positionierung erreicht werden als auch durch eine Position im Raster, die verhältnismäßig geringe Einschränkungen für die weitere Suche bietet. Bei der Wertauswahl anhand der Äquivalenz von Objekt- und Rasterbewertung werden größere Objekte tendenziell eher weiter außen platziert. Bei der konstruktiven Wertauswahl werden Objekte so in das Layout integriert, dass möglichst zusammenhängende Blöcke von Objekten entstehen und die Anzahl der Schnitte durch das Layout reduziert wird.

Durch die Aufteilung der Objekte in eine Reihe von Blöcken können Materialflüsse in unterschiedlichen Punkten beginnen und enden. Die schwankende Größe der Flüsse hat, wie auch die Raumgröße, sowohl auf die Variablen- als auch Wertauswahl Einfluss. Dazu sollen die Objekte mit hohen Flüssen früh ausgewählt werden, um eine gute relative Positionierung zueinander zu ermöglichen. Die Wertauswahl kann dabei einerseits statisch aufgrund der Äquivalenz von Objekt- und Rasterbewertung als auch dynamisch erfolgen. Eine dynamische Auswahl würde dabei bedeuten, dass die Position, die zu diesem Zeitpunkt zu den geringsten Flusskosten führt, ausgewählt wird.

Die Constraints die zur Abbildung technischer Nebenbedingungen dem Modell zugefügt werden, beeinflussen die Domänengröße der Variablen und fin-

### 4.3. Eignung von Modell und Lösungsstrategie in Bezug auf die Anforderungen

den auf diese Weise Eingang in die Suchstrategie. Da im Fall von binären Constraints eine Reduktion sich erst nach der Platzierung einer der beiden beteiligten Variablen auswirkt, werden diese Constraints gesondert auf Basis ihrer Restriktivität bewertet. Dies ist beispielsweise bei einem Maximalabstand von eins zwischen zwei Blöcken der Fall. Solange keiner der beiden Blöcke platziert wurde, besteht kein Einfluss auf die Domänengröße. Sobald jedoch eine der beiden Variablen mit einem Wert belegt wurde, stellt die Bedingung eine sehr starke Restriktion dar, die unter Umständen nicht mehr erfüllt werden kann. Dies soll durch die gesonderte Bewertung dieser Gruppe von Constraints verhindert werden.



## 5. Evaluierung der Ergebnisse

Im folgenden Kapitel werden in Abschnitt 5.1 zunächst die verwendeten Testinstanzen erläutert und auf die Problematik einer geeigneten Auswahl eingegangen. Dabei wird auf Instanzen des quadratischen Zuordnungsproblem und davon abgeleitete Testdaten, auf Instanzen des Single-Row-Facility-Layout-Problems und speziell für das Modell erstellte Layoutprobleme zurückgegriffen. Abschnitt 5.2 geht anschließend auf die Ergebnisse der Berechnungen ein. Dabei wird jeweils auf die im vorigen Kapitel genannten Möglichkeiten zur Reduktion des Lösungsraums, als auch auf die Ansätze zur Variablen- und Werteauswahl eingegangen. Abschließend werden in Abschnitt 5.3 auf Basis der Beobachtungen Kenngrößen der Datensätze bestimmt, die zur Auswahl einer effizienten Strategie geeignet sind.

### 5.1. Verwendete Testdaten

Die Bewertung der im vorigen Kapitel vorgestellten Rechenansätze erfolgt unter der zusätzlichen Schwierigkeit, dass kaum Testinstanzen existieren, die zum Vergleich herangezogen werden könnten. Es ist lediglich möglich, Testinstanzen vergleichbarer Probleme zu verwenden und so in einem begrenztem Rahmen Aussagen treffen zu können. Daher sollen im Folgenden zunächst bereits optimal gelöste Instanzen des quadratischen Zuordnungsproblems betrachtet werden. In einem zweiten Schritt werden einige dieser optimalen Lösungen in Blöcke unterschiedlicher Größe zerlegt und ebenso mit dem Programm berechnet. Somit kann auch ein abgewandeltes Problem, für das jedoch die gleiche optimale Lösung gilt wie für das ursprüngliche quadratische Zuordnungsproblem, betrachtet werden. Darüber hinaus existieren noch einige weitere Instanzen alternativer Formulierungen von Layoutproblemen, deren Ergebnisse sich bis zu einem gewissen Grad übertragen lassen. In einem letzten Schritt sollen selbst erstellte Testgrößen berechnet werden, die explizit den ursprünglich formulierten Anforderungen an das Modell entsprechen. Auch wenn in diesen Fällen kein direkter Vergleich bezüglich der Optimalität mehr möglich ist, lässt sich doch zeigen, dass auch für umfangreichere Pro-

bleme mit einer großen Zahl an Nebenbedingungen Lösungen in akzeptabler Zeit gefunden werden können.

### 5.1.1. Anwendung auf das quadratische Zuordnungsproblem

Das quadratische Zuordnungsproblem ist eines der am besten untersuchten Probleme des Operations Research. Aus diesem Grund finden sich auch zahlreiche Testinstanzen unterschiedlicher Größe, deren optimale Lösung bekannt ist.<sup>8</sup> Zur Überprüfung der Rechenergebnisse wurde eine Reihe von Datensätzen ausgewählt, die jeweils eine rechtwinklige Entfernungsmessung verwenden und zum Teil auch eine symmetrische Flussmatrix aufweisen. Die verwendeten Instanzen sowie ihr jeweiliges Optimum sind in Tabelle 5.1 aufgelistet.

Tabelle 5.1.: Verwendete Testinstanzen des quadratischen Zuordnungsproblems

Bezeichnung	Optimum	Bezeichnung	Optimum
nug12	578	sko56	34.458*
nug14	1.014	sko64	48.498*
nug15	1.150	sko72	66.256*
nug16a	1.610	sko81	90.998*
nug18	1.930	sko90	115.534*
nug20	2.570	sko100	152.002*
nug24	3.488	tho30	149.936
nug27	5.234	tho40	240.516*
nug30	6.124	tho150	8.133.398*
sko42	15.812*	wil50	48.816*
sko49	23.386*	wil100	273.038*

Die Optimalität der mit \* markierten Lösungen wurde bisher nicht bewiesen. Die entsprechenden Werte stellen die bisher beste bekannte Lösung dar. Die Instanzen stammen von Nugent et al. (1968), Skorin-Kapov (1990), Blte und Thonemann (1996) und Wilhelm und Ward (1987).

---

<sup>8</sup>Eine von R.E. Burkard, S.E. Karisch und F. Rendl erstellte Online-Bibliothek mit Instanzen sowie weiteren Informationen zum quadratischen Zuordnungsproblem ist unter <http://www.seas.upenn.edu/QZPlib/> zu finden. Die Seite wird seit 2002 an der University of Pennsylvania von Peter Hahn betreut.

Für diese Testdaten kommt nur eine begrenzte Auswahl an Suchstrategien in Frage. In diesen Fällen liegt die Schwierigkeit nicht im Ermitteln einer zulässigen Lösung, sondern im Finden möglichst guter Lösungen, idealerweise des Optimums innerhalb eines sehr großen Lösungsraums. Da alle Objekte die gleiche Größe haben und auf allen Positionen im Raster platziert werden können, kommen Strategien, die die Objektgröße oder etwaige Nebenbedingungen berücksichtigen, nicht in Betracht. Auch eine Betrachtung der Domänengröße liefert keine verwertbaren Erkenntnisse, da diese während des Suchprozesses für alle verbleibenden Objekte identisch ist. Somit kommen für die Variablenauswahl lediglich eine Selektion anhand der ein- und ausgehenden Flüsse in Betracht. Für die Werteauswahl wird einerseits die flussoptimierte Werteauswahl verwendet, die jedem Objekt die zu diesem Zeitpunkt bestmögliche Position zuweist. Als Alternative wird ebenso auf die Positionierung anhand der Äquivalenz der Fluss- und Rasterbewertung zurückgegriffen. Die in diesem Ansatz ebenso enthaltene Objektgröße wird aus dem bereits genannten Grund nicht berücksichtigt. Ebenso in Betracht kommt eine Mischung dieser beiden Strategien.

### 5.1.2. Aus dem quadratischen Zuordnungsproblem abgeleitete Instanzen

Wie zu Beginn des letzten Abschnitts beschrieben, lassen sich die in Kapitel 4 beschriebenen Lösungsansätze nur begrenzt auf das quadratische Zuordnungsproblem anwenden. Da ansonsten nur wenige weitere Vergleiche möglich sind, sollen im Weiteren nun aus dem quadratischen Zuordnungsproblem abgeleitete Instanzen betrachtet werden. Zwar entsprechen diese auch nur eingeschränkt den Zielvorstellungen, die dem beschriebenen Modell zugrunde liegen, jedoch lassen sich zumindest einige Eigenschaften des Modells darauf anwenden.

Deshalb wurden zu einigen der in Tabelle 5.1 genannten Instanzen mit zusätzlichen Nebenbedingungen weitere Testdatensätze generiert, die einen stärker eingeschränkten Lösungsraum aufweisen, jedoch zur gleichen optimalen Lösung führen. Dazu wurde die bekannte optimale Lösung in mehrere rechteckige Objekte aufgeteilt, die jedoch die gleichen Flüsse aufweisen wie in dem ursprünglichen Problem. Da dessen optimale Lösung nach wie vor erreichbar ist, muss diese offensichtlich ebenso das Optimum des dabei entstehenden Problems sein. Ein Beispiel für dieses Vorgehen ist in Abbildung 5.1 zu sehen, die Maße der daraus resultierenden Objekte ist in Tabelle 5.2 angegeben. Mit Hilfe dieses Verfahrens wurden aus den Instanzen *sko64*, *sko81*, *sko100*, *wil50*

## 5. Evaluierung der Ergebnisse

und *wil100* jeweils drei zusätzliche Instanzen kreiert. Die Aufteilung dieser Instanzen ist im Anhang zu finden.

10	3	35	21	29	23	48	44	7	76	17
9	45	50	12	72	61	38	92	36	53	41
8	58	70	66	100	26	47	99	46	64	90
7	10	24	40	67	85	30	73	51	69	20
6	42	59	68	8	93	56	81	82	84	87
5	60	9	86	98	96	89	19	49	2	28
4	88	62	16	52	4	22	32	34	57	15
3	27	80	54	94	75	31	43	14	77	6
2	39	5	13	55	74	95	25	63	79	91
1	65	83	11	33	71	97	1	78	37	18
	1	2	3	4	5	6	7	8	9	10

10	3	35	21	14	29	23	48	44	7	76	20	
9	45	12	50	12	72	15	38	16	2	36	53	19
8	58	70	66	100	26	47	99	46	17	90		
7	10	24	10	67	85	30	11	51	69	20		
6	42	59	68	8	93	56	81	82	84	87		
5	60	9	86	98	96	89	19	49	2	8	28	
4	88	62	16	52	4	22	32	34	57	15		
3	27	80	54	94	75	31	43	14	77	6		
2	39	5	13	55	74	95	25	63	79	91		
1	65	83	11	33	71	97	1	78	37	18		
	1	2	3	4	5	6	7	8	9	10		

Abbildung 5.1.: Aufteilung in Blöcke am Beispiel der Instanz *sko100a*

Tabelle 5.2.: Maße der Objekte nach der Aufteilung gemäß Abbildung 5.1

	l	b		l	b		l	b		l	b
<b>1</b>	6	4	<b>6</b>	2	2	<b>11</b>	3	1	<b>16</b>	2	3
<b>2</b>	2	4	<b>7</b>	3	2	<b>12</b>	2	3	<b>17</b>	3	1
<b>3</b>	2	2	<b>8</b>	2	1	<b>13</b>	2	2	<b>18</b>	2	2
<b>4</b>	2	2	<b>9</b>	2	2	<b>14</b>	2	1	<b>19</b>	1	1
<b>5</b>	3	2	<b>10</b>	5	1	<b>15</b>	1	3	<b>20</b>	1	1

Auf diese Weise können in begrenztem Rahmen auch heterogene Raumgrößen getestet werden. Problematisch ist in diesem Fall jedoch, dass alle Rasterpositionen belegt sind und somit nur geringe Freiräume bestehen. In diesen Fällen tritt die Zulässigkeit der Lösung immer stärker in den Fokus, die Optimalität hat nur einen geringen Stellenwert. Aus diesem Grund wurde bei der dritten kreierten Instanz eine Einteilung vorgenommen, die das Finden zulässiger Lösungen erleichtert und bei der der Schwerpunkt stärker auf dem Finden möglichst guter Lösungen liegen soll. Dafür wurde die Einteilung in einer Weise vorgenommen, bei der sich die einzelnen Objekte möglichst viele Maße teilen, beziehungsweise es sich um Vielfache eines Maßes handelt. Eine solche Einteilung ist in Abbildung 5.2 zu sehen.

10	3	35	21	29	23	48	44	7	76	17
9	45	50	12	72	61	38	92	36	53	41
8	58	70	66	100	26	47	99	46	64	90
7	10	24	40	67	85	30	73	51	69	20
6	42	59	68	8	93	56	81	82	84	87
5	60	9	86	98	96	89	19	49	2	28
4	88	62	16	52	4	22	32	34	57	15
3	27	80	54	94	75	31	43	14	77	6
2	39	5	13	55	74	95	25	63	79	91
1	65	83	11	33	71	97	1	78	37	18
	1	2	3	4	5	6	7	8	9	10

10	3	35	21	29	23	48	44	7	76	17
9	15		12	72	61	38	92	36	53	41
8	58	70	16		26	47	99	46	64	90
7	14		40	67	85	30	73	51	69	18
6	42	4	59	68	8	93	56	81	82	84
5	60	3	9	86	98	9		19	49	2
4	88	62	16	52	4	22	32	34	57	15
3	2		54	94	75	31	43	14	77	6
2	39	5	13	55	74	95	25	63	79	91
1	1		11	33	71	97	6		1	78
	1	2	3	4	5	6	7	8	9	10

Abbildung 5.2.: Aufteilung in Blöcke mit ähnlichen Maßen

Trotz dieser Modifikationen lassen sich eine Reihe der im vorigen Kapitel vorgestellten Möglichkeiten nicht ausnutzen. Beispielsweise beschränken sich alle Beispiele auf zweidimensionale Probleme, es treten auch keinerlei Nebenbedingungen bezüglich der absoluten oder relativen Position der Objekte auf.

### 5.1.3. Vergleichbare Instanzen alternativer Formulierungen

Vereinzelt lassen sich Instanzen, die für Modelle mit einer anderen Ausrichtung aufgestellt wurden, zum Teil mit Anpassungen übernehmen. So lässt sich das von Bazaraa (1975) verwendete Modell mit geringfügigen Änderungen lösen.

Bazaraa baut dabei auf einer Variante des quadratischen Zuordnungsproblems auf. Ähnlich zu der blockweisen Einteilung in Abschnitt 5.1.2, platziert er Objekte unterschiedlicher Form und Größe innerhalb des Rasters. Jedem Objekt wird dabei eine Menge möglicher Positionen zugewiesen, wobei die Objekte zum Teil auch gespiegelt werden. Gemäß Bazaraa stammen diese möglichen Positionierungen aus einer Vorauswahl der beteiligten Planer. Ohne diese Einschränkung wären die entstehenden Instanzen mit ihren etwa 60 Positionen für die damalige Zeit nicht lösbar gewesen. In der Veröffentlichung werden zwei Instanzen mit 12 beziehungsweise 14 Objekten vorgestellt, die im Folgenden mit *baz12* und *baz14* bezeichnet werden. Die optimale Lösung von *baz12* mit dem Zielfunktionswert von 14079 ist in Abbildung 5.3 dargestellt.

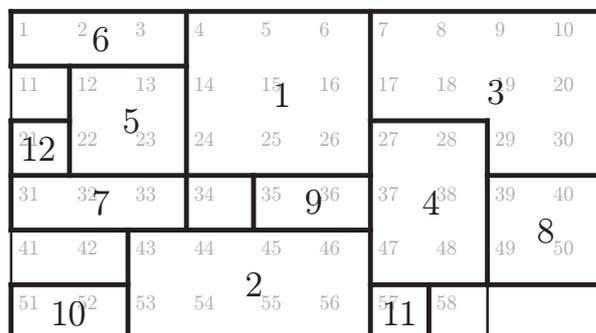


Abbildung 5.3.: Optimale Anordnung von 12 Objekten auf 58 Positionen für *baz12*

Bei der Lösung müssen zwei Anpassungen vorgenommen werden. Einerseits wird aus der Veröffentlichung nicht klar, wie die angegebenen Flüsse auf die einzelnen Blöcke der Objekte aufgeteilt werden. Hierfür wurde die Annahme getroffen, dass der Fluss zwischen zwei Objekten sich zu gleichen Teilen auf die beteiligten Objekte aufteilt; diese Annahme korrespondiert auch mit dem sich ergebenden Optimum.<sup>9</sup> Zudem werden in dieser Arbeit nicht direkt dem Objekt mögliche Positionen zugewiesen, sondern jeweils den zu einem Objekt gehörenden Blöcken. Dadurch kann es passieren, dass, falls sich mehrere Bereiche im Ursprungsmodell überschneiden oder berühren, das Objekt nicht diese konkreten Positionen einnimmt, sondern die Blöcke Positionen aus zwei unterschiedlichen Mengen einnehmen. Die Wahrscheinlichkeit für ein solches Verhalten ist bei kleineren Objekten offensichtlich größer als bei größeren Objekten. Betrachtet man nochmals Abbildung 5.3, tritt dieser Fall beispielsweise für Objekt zehn auf. Für dieses sollen unter anderem die Positionen {51, 52} sowie {42, 43} in Frage kommen. In der optimalen Lösung nach dem in dieser Arbeit vorgestellten Modell wird das Objekt den Positionen {42, 52} zugeordnet. Dies ist möglich, da nur eingegrenzt werden kann, dass für den ersten Block des Objekts die Positionen {42, 51} und für den zweiten Block die Positionen {43, 52} möglich sind. Dementsprechend resultiert daraus eine geringfügige Abweichung gegenüber der optimalen Lösung.

Neben dem genannten Beispiel wurden noch drei Instanzen unterschiedlicher

---

<sup>9</sup>Das bedeutet beispielsweise, dass der Fluss zwischen einem aus drei Blöcken bestehenden Objekt und einem aus vier Blöcken bestehenden Objekt durch zwölf geteilt wird und alle Verbindungen zwischen den einzelnen Blöcken jeweils diesen Fluss zugeordnet bekommen.

Größe des Single Row Facility Layout Problems getestet. Bei diesem wird eine Menge von Objekten unterschiedlicher Größe linear angeordnet. Die Flussrichtung spielt keine Rolle; das heißt, die Transportkosten entlang und entgegen der Linie sind identisch. Die Instanzen mit acht beziehungsweise elf Objekten stammen von Simmons (1969), die Instanz mit 20 Einheiten basiert auf den Daten des quadratischen Zuordnungsproblems von Nugent und stammen von Heragu und Kusiak (1991).<sup>10</sup>

#### 5.1.4. Testinstanzen entsprechend der Modellanforderungen

Um einige Beispiele zu berechnen, die den Modellanforderungen entsprechen, wurden neben den bestehenden Instanzen einige zusätzliche Instanzen entworfen. Hierzu wurden die Anzahl der Objekte und der Nebenbedingungen, die Größe des Rasters, der Füllgrad des Rasters sowie die Verteilung der Flüsse variiert. Die Flüsse wurden nach folgenden beiden Ansätzen zufällig generiert:

1. Alle Flüsse sind normalverteilt mit dem gleichen Mittelwert von 10 und einer Varianz von 50.
2. Alle Flüsse zwischen zwei Objekten sind jeweils mit einer Varianz von 50 um den gleichen Mittelwert normalverteilt. Dieser Mittelwert wird für je zwei Objekte zufällig bestimmt und entspricht der Klassenmitte einer Gleichverteilung zwischen 0 und 20 bei fünf Klassen. Der Mittelwert wird somit aus der Menge  $\{2, 6, 10, 14, 18\}$  zufällig gewählt.

Zudem ist der Fluss zwischen zwei Objekten mit einer Wahrscheinlichkeit von  $\alpha$  gleich null. Durch die beiden Varianten sollen unterschiedlich stark schwankende Flussgrößen erzeugt werden.

Mit den unterschiedlichen Einflussgrößen wurde eine Reihe von Instanzen erstellt, deren Eckdaten in Tabelle 5.3 angegeben sind. Die Anzahl der Nebenbedingungen entspricht jeweils der Anzahl der Objekte.

Alle Instanzen wurden mit dem Ansatz, der im jeweiligen Fall die besten Ergebnisse zeigte, nochmals bei längerer Laufzeit berechnet. Die Lösungen die dabei gefunden wurden, sollen als Referenzwert für einen Vergleich der Ansätze dienen und sind in Tabelle 5.4 angegeben. Die gewählte Rechenzeit lag dabei je nach Modellgröße zwischen drei und fünf Stunden.

---

<sup>10</sup>Die Instanzen werden im Folgenden mit  $sim8_{SRFLP}$ ,  $sim11_{SRFLP}$  und  $nug20_{SRFLP}$  bezeichnet.

Tabelle 5.3.: Kenngrößen der erstellten Instanzen

Bezeichnung	Raster	Objekte	Füllgrad	Mittelwert	$\alpha$
k20a1	10x8x5	20	90%	Gleich	0,99
k20a2	10x8x5	20	90%	Klassen	0,99
k20b1	10x8x5	20	75%	Gleich	0,99
k20b2	10x8x5	20	75%	Klassen	0,99
k25a1	12x10x5	25	90%	Gleich	0,995
k25a2	12x10x5	25	90%	Klassen	0,995
k25b1	12x10x5	25	75%	Gleich	0,995
k25b2	12x10x5	25	75%	Klassen	0,995
k30a1	16x10x5	30	90%	Gleich	0,999
k30a2	16x10x5	30	90%	Klassen	0,999
k30b1	16x10x5	30	75%	Gleich	0,999
k30b2	16x10x5	30	75%	Klassen	0,999

Tabelle 5.4.: Beste gefundene Lösungen der erstellten Instanzen

Instanz	Referenzlösung	Instanz	Referenzlösung
k20a1	93.716	k25b1	74.207
k20a2	91.075	k25b2	65.204
k20b1	59.325	k30a1	46.774
k20b2	65.204	k30a2	35.304
k25a1	114.052	k30b1	33.489
k25a2	123.320	k30b2	24.356

## 5.2. Ergebnisse der Testläufe

Bei der Betrachtung der Rechenergebnisse muss der unterschiedliche Charakter der Eingangsdaten berücksichtigt werden. Zwangsläufig wird nicht jede Methode unabhängig von den Daten immer brauchbare Ergebnisse liefern. Deshalb soll die Wirksamkeit der Ansätze in Anwendung auf jeweils passende Instanzen gezeigt werden. Im Anschluss sollen Kennzeichen der Datensätze identifiziert werden, anhand derer die Methodenauswahl erfolgen kann.

### 5.2.1. Auswirkungen der unteren Schranke

Die untere Schranke wurde in erster Linie auf das QZP und die davon abgeleiteten Instanzen angewendet. Der Hauptfokus lag dabei auf kleineren Datensätzen, da die Qualität der Gilmore-Lawler-Schranke mit wachsender Problemgröße deutlich abnimmt. Dieser Aspekt ist zwar allen Schranken gemein, tritt jedoch in diesem Fall besonders stark auf (Loiola et al. 2007).

Auf die Geschwindigkeit, mit der Lösungen gefunden beziehungsweise verbessert werden, wirkt sich die gewählte untere Schranke nicht positiv aus. Es kann lediglich eine geringfügige Reduktion des Suchbaums erreicht werden. Die Zeit zur Berechnung der unteren Schranke ist trotz ihrer verhältnismäßig geringen Komplexität so hoch, dass die Suche langsamer abläuft.

Dabei wurde neben einer einmaligen Berechnung der Schranke zu Beginn der Suche auch eine mehrfache Berechnung betrachtet. Dadurch besteht die Möglichkeit, dass eine Schranke, die berechnet wird, nachdem bereits einige Blöcke platziert wurden, dichter am Optimum liegt und somit mehr Suchzweige ausgeschlossen werden können. Allerdings zeigte auch dieser Ansatz keinen Erfolg.

Fraglich ist jedoch ebenso, ob die Adaption einer komplexeren Schranke für praktisch relevante Problemgrößen Erfolg versprechen würde, da deren Qualität ebenso mit der Problemgröße abnimmt. Für die gegebenen Instanzen mit 100 bis 150 Variablen liegt die beste bekannte untere Schranke drei bis sechs Prozent unter der besten bekannten Lösung.<sup>11</sup> Dadurch wird die Anwendbarkeit in dem vorliegenden Modell eingeschränkt.

---

<sup>11</sup>Basierend auf den Angaben unter <http://www.seas.upenn.edu/QZPLib/>.

### 5.2.2. Propagieren möglicher Positionen eines Objekts

Bei diesem Verfahren wurde der Wertebereich eines Blocks unter Berücksichtigung des zugehörigen Objekts und potentieller Positionen im Raster eingeschränkt. Dadurch lässt sich der Wertebereich des jeweiligen Blocks und somit auch der Lösungsraum zu einem bestimmten Suchzeitpunkt erheblich reduzieren. Dies ist jedoch mit einem gewissen Aufwand verbunden. Da der Ansatz bei Objekten, die aus nur einem Block bestehen, keine Wirkung haben kann, wurde die Methode auf die Instanzen des QZP (Abschnitt 5.1.1) nicht angewendet.

In der Regel lassen sich durch das Propagieren der in Frage kommenden Positionen wesentlich schneller zulässige Lösungen finden. Dies ist in Abbildung 5.4 beispielsweise deutlich bei den Instanzen *sko64A* und *sko64C* zu sehen. Für alle in der Abbildung gezeigten Berechnungen wurde die Variablenauswahl allein auf Basis der Domänengröße getroffen; als Wert wurde jeweils der kleinste Wert der Domäne gewählt.

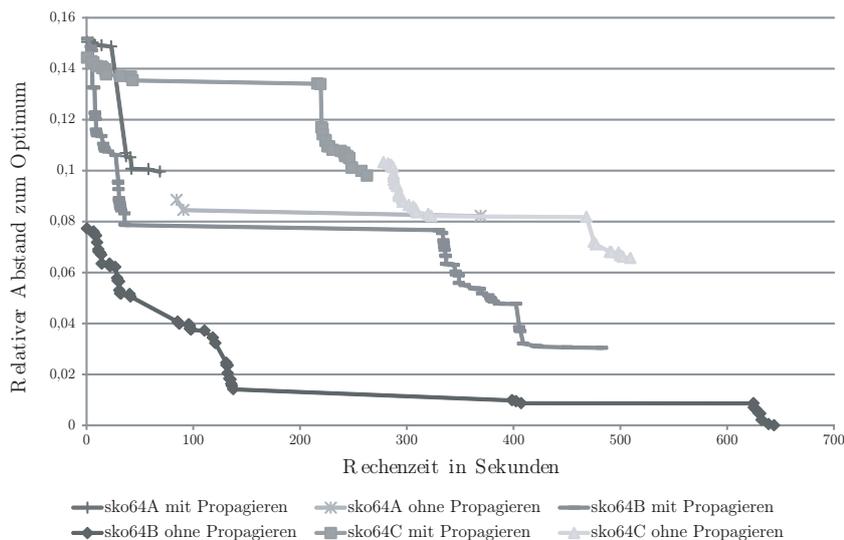


Abbildung 5.4.: Vergleich der Lösungssuche für die von *sko64* abgeleiteten Instanzen mit und ohne Propagieren möglicher Positionen

Diese Lösungen haben jedoch eine im Vergleich schlechtere Qualität. Der Grund dafür liegt in der verwendeten Suchstrategie, bei der die Suche nach einer gewissen Zahl von Fehlschlägen neu gestartet wird. Da durch das Verfahren viele Variablenbelegungen, die nicht zu einer gültigen Lösung führen,

ausgeschlossen werden, treten für genau diese Fälle weniger Fehlschläge auf. Dadurch dauert es länger, bis die Suche einen bestimmten Suchzweig wieder verlässt. Solange es verhältnismäßig einfach ist, zulässige Lösungen zu finden, steigt mit der Anzahl an unterschiedlichen Suchzweigen auch die Wahrscheinlichkeit, bessere Lösungen zu finden. Dieser Unterschied ließe sich jedoch durch eine Anpassung der Zahl der Fehlschläge, nach der ein Neustart erfolgt, leicht ausgleichen. Es sollte jedoch zunächst nur gezeigt werden, dass Lösungen sich mit dieser Art der Propagierung schneller finden lassen.

Im Fall von Instanz *sko64B* wird auch ohne das Propagieren nach weniger als einer Sekunde eine mögliche Lösung gefunden. Der Grund dafür liegt in der im Vergleich zur Rastergröße relativ hohen Objektzahl von 20. Dadurch existiert eine sehr große Zahl an möglichen Lösungen, sodass sich auch ohne das Propagierungsverfahren schnell Lösungen finden lassen. Entsprechend sollte für die Entscheidung, ob das Verfahren eingesetzt werden soll, sowohl die durchschnittliche Anzahl an Blöcken pro Objekt als auch deren Streuung berücksichtigt werden. Ebenso lassen sich mit sinkendem Füllgrad des Rasters leichter Lösungen finden. Diese Punkte können gegen eine Anwendung des Verfahrens sprechen.

Vergleichbare Ergebnisse lassen sich für Instanzen beobachten, die von *wil100* abgeleitet wurden (siehe Abbildung 5.5). Dabei wurde mit identischen Methoden wie bei den von *sko64* abgeleiteten Instanzen vorgegangen. Für *wil100A* wurde dabei in beiden Fällen keine Lösung gefunden. Bei der Instanz *wil100B* wurden nur mit dem Propagierungs-Ansatz bei *wil100C* wurden in beiden Fällen Lösungen gefunden; mit dem Propagieren von möglichen Positionen jedoch deutlich schneller.

Für die Instanzen, die gemäß den Modellanforderungen erstellt wurden (vergleiche Abschnitt 5.1.4), wurden ohne das Propagierungsverfahren in keinem Fall und mit Hilfe des Verfahrens nur in einem Fall innerhalb der jeweiligen Suchzeit eine Lösung gefunden. Die Suchzeit lag für diese Fälle je nach Modellgröße zwischen 15 und 30 Minuten. Problematisch ist dabei, dass bei zunehmender Anzahl an Objekten häufiger identische Bewertungen vorliegen und Objekte nicht mehr an einem Stück platziert werden. In diesen Fällen ist die Anwendung der im folgenden Abschnitt diskutierten Methode zwingend notwendig.

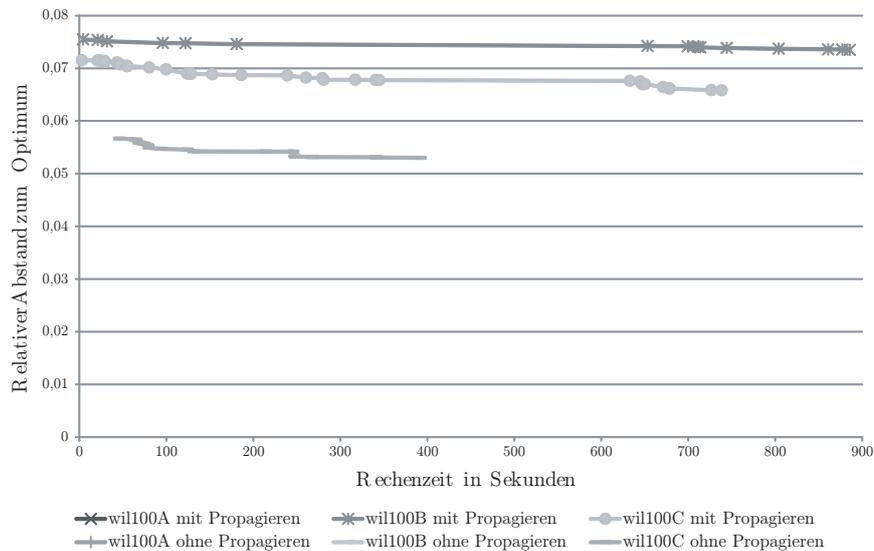


Abbildung 5.5.: Vergleich der Lösungssuche für die von *wil100* abgeleiteten Instanzen mit und ohne Propagieren möglicher Positionen

### 5.2.3. Propagieren der Objekte

Bei diesem Ansatz wird zunächst ein Objekt ausgewählt. Anschließend werden so viele Eckpunkte des Objekts platziert, bis die Lage des Objekts eindeutig bestimmt ist. Die Bedingung schränkt dabei einerseits die möglichen Positionen der jeweils folgenden Eckpunkte ein und platziert anschließend die restlichen Blöcke. Durch die Bedingung wird verhindert, dass Objekte unvollständig platziert werden oder zeitgleich mehrere Objekte positioniert werden. Verglichen wird dieses Vorgehen mit der Möglichkeit, dass ein Objekt nicht über eine einzelne Bedingung erstellt wird, sondern über eine Reihe von binären Constraints, die jeweils die Abstände und Ausrichtungen zwischen den einzelnen Blöcken eines Objekts festlegen. Da die Variablenauswahl über die Domänengröße erfolgt, werden auch in diesem Fall Objekte immer noch „am Stück“ platziert. Der Grund hierfür ist, dass nach dem Platzieren des ersten Blocks die Platzierungsmöglichkeiten der weiteren Blöcke eines Objekts über die besagten Abstandsbedingungen soweit eingeschränkt wird, dass diese Blöcke als nächstes gewählt werden.

Da in diesem Fall die Abstandsbedingungen nicht einzeln ausgewertet werden müssen, sondern der Abstand über ein einzige (dafür etwas komplexere) Bedingung definiert wird, läuft die Suche über diesen Ansatz in nahezu allen

Fällen deutlich effizienter ab. Eine Ausnahme bilden dabei wiederum kleinere Instanzen, die aus verhältnismäßig vielen Objekten bestehen, wie in Abbildung 5.6 für die Instanz *sko64B* zu sehen ist.

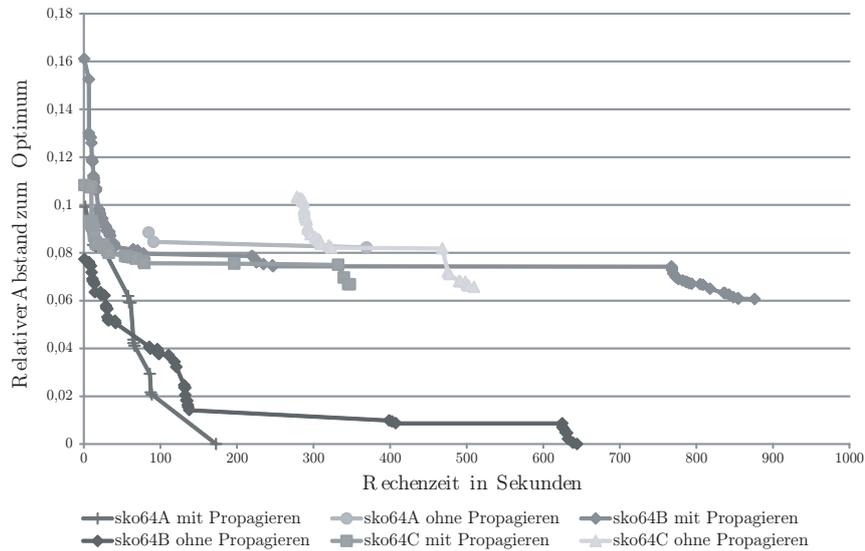


Abbildung 5.6.: Vergleich der Lösungssuche für die von *sko64* abgeleiteten Instanzen mit und ohne Propagieren der Objekte

Bei größeren Instanzen gilt wiederum, dass sich mit dem Propagieren über eine einzelne Bedingung sich schneller Lösungen finden lassen als ohne das Propagieren (Abbildung 5.7). In zwei Fällen (*wil100A* und *wil100B*) wird innerhalb der gegebenen Zeit von 15 Minuten keine zulässige Lösung gefunden wird. Bei den anhand der Modellanforderungen erstellten Instanzen wurde in keinem Fall ohne das Propagieren der Objekte eine Lösung innerhalb der gegebenen Zeit gefunden.

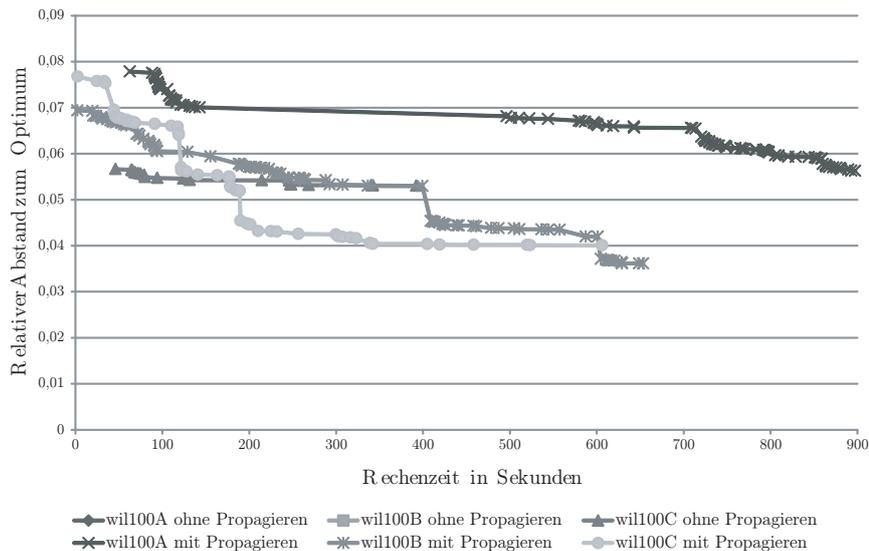


Abbildung 5.7.: Vergleich der Lösungssuche für die von *wil100* abgeleiteten Instanzen mit und ohne Propagieren der Objekte

### 5.2.4. Propagieren der Nachfrage nach einer Position

Um die Nachfrage nach einer Position abzubilden, wurde, aufbauend auf dem Propagieren der möglichen Positionen, überprüft, ob eine Position von mehr Blöcken nachgefragt wird als Ausweichpositionen für diese Blöcke zur Verfügung stehen. Dadurch soll erkannt werden, ob eine bestimmte Position einen Engpass darstellt oder nicht.

Um alle Engpässe zu erkennen, müsste das Verfahren für alle Positionen, unabhängig von der Zahl der in Frage kommenden Blöcke durchgeführt werden. Gerade bei Positionen, die vielen Blöcken als Wert zugewiesen werden können, benötigt dies viel Zeit. Aus diesem Grund wurden nur Positionen berücksichtigt, die im Wertebereich eines Objekts liegen, das sonst keine weitere Positionierungsmöglichkeit mehr besitzt.

Die Anwendung dieses Verfahrens erbrachte nur in vereinzelten Fällen und darüber hinaus vernachlässigbare Verbesserungen. In der Regel verlief die Suche aufgrund des zusätzlichen Aufwandes geringfügig langsamer. Als problematisch erwies sich dabei, dass der beschriebene Fall so selten auftritt, dass das Propagieren für diese Einzelfälle sich nicht lohnt. Es würde nur zu einem früheren Abbruch eines Suchpfades kommen, wenn zwei Objekte mit identischen Maßen beide nur noch in einem, identischen Bereich des Rasters

positioniert werden können. In diesem Fall würde jedoch die Suche bei Anwendung der Variablenauswahl anhand der Domänengröße auch im Folgeschritt auf den Fehler treffen, da anschließend eins der beiden Objekte ausgewählt würde und das andere keine gültige Position mehr hätte. Da dieser Fall nur selten eintritt und zudem nur geringe Auswirkungen hätte, ergibt sich für die Suche kein wesentlicher Unterschied.

### 5.2.5. Effizienz der Variablen- und Wertauswahl

Die Auswahl von Variablen und Werten hängt stark von den unterschiedlichen Instanzen ab. Wie bereits erwähnt kommen beispielsweise für die Instanzen des quadratischen Zuordnungsproblems eine Reihe der Ansätze wie die Auswahl anhand der Domänengröße oder der Objektgröße nicht in Frage. Aus diesem Grund sollen im Folgenden die Ansätze getrennt nach den unterschiedlichen Gruppen von Instanzen betrachtet werden.

#### Instanzen des QZP

In diesem Fall stellt lediglich die Variablenauswahl anhand der ein- und ausgehenden Flüsse eine sinnvolle Alternative dar. Als Vergleichswert wird noch eine zufällige Auswahl betrachtet. Für die Wertauswahl ist die Äquivalenz sowohl von Objekt- und Rasterbewertung als auch eine flussoptimierte Positionierung möglich. Auch für diesen Fall wird eine zufällige Positionierung als Vergleich betrachtet.

Für alle Instanzen zeigen sich ähnliche Ergebnisse. Am besten schneidet dabei eine Variablenauswahl anhand der Flussgröße in Kombination mit der flussoptimierten Wertauswahl ab. Für Instanzen bis zur Größe von 20 Objekten konnte dabei die optimale Lösung in der Regel innerhalb verhältnismäßig kurzer Zeit gefunden werden. Mit wachsender Instanzgröße wuchs mit der steigenden Zahl an möglichen Lösungen auch der Abstand zwischen der besten gefundenen Lösung und dem Optimum, beziehungsweise der besten bisher bekannten Lösung.<sup>12</sup>

Für die Wertauswahl anhand der Äquivalenz von Objekt- und Rasterbewertung wurden gemischte Ergebnisse beobachtet. Bei einigen Instanzen, die starke Schwankungen der ein- und ausgehenden Flüsse aufwiesen, schnitt der Ansatz verhältnismäßig gut ab. Bei geringerer Streuung der Flüsse fallen

---

<sup>12</sup>Im Folgenden sollen sowohl Lösungen mit bewiesener Optimalität als auch Lösungen, die den besten bisher bekannten Zielfunktionswert aufweisen, der Einfachheit halber als Optimum bezeichnet werden.

die Ergebnisse jedoch teilweise auch schlechter aus als bei einer zufälligen Variablen- und Werteauswahl (siehe auch Abbildung 5.8 und 5.9).

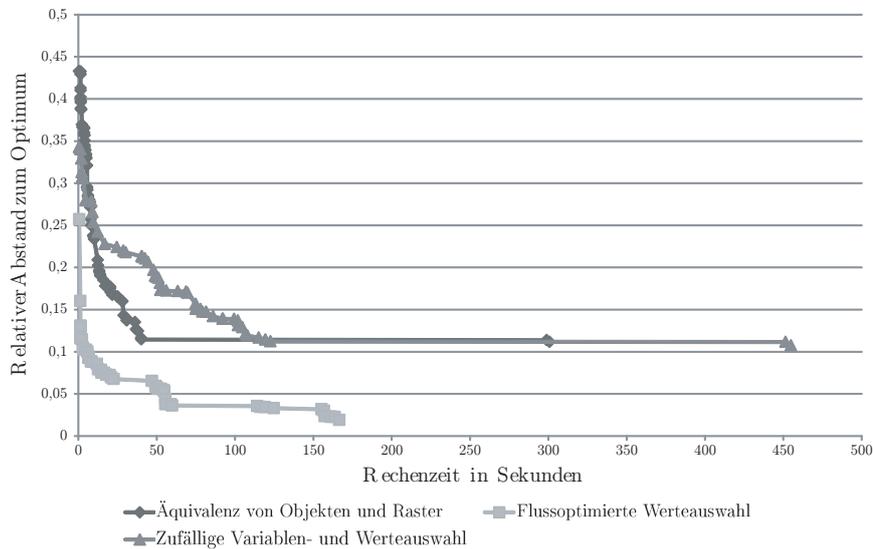


Abbildung 5.8.: Lösungssuche für die Instanz *tho40*

Dies lässt sich erklären, wenn man für die Optimallösung verschiedener Instanzen die Summe der ein- und ausgehenden Flüsse betrachtet. In Fällen mit starken Schwankungen lässt sich häufig beobachten, dass Objekte mit relativ hohen Flüssen eher zentral platziert werden, während Objekte an Rändern oder Eckpunkten eher weniger Flüsse aufweisen (siehe Abbildung 5.10). Da dies das Prinzip ist, das der Werteauswahl anhand der Äquivalenz von Objekt- und Rasterbewertung zugrunde liegt, sind in diesen Fällen positive Ergebnisse zu beobachten. In anderen Fällen mit geringeren Schwankungen ergibt sich für das Optimum kein Muster, so dass dieser Ansatz nicht geeignet ist. Problematisch ist dabei zudem, dass die Reihenfolge der Variablen- und Werteauswahl im Vorfeld bereits festliegt und somit auf die Anordnung eines Teillayouts während der Suche nicht reagiert wird. Ähnliches gilt für Instanzen, die verhältnismäßig viele Leerflüsse aufweisen. In diesen Fällen können sich gegebenenfalls mehrere Zentren mit hohen Flüssen im Layout bilden, sodass diese nicht mehr mit dem Zentrum des Rasters zusammenfallen. Ein derartiger Fall ließe sich beispielsweise mit Hilfe einer Clusteranalyse im Vorfeld erkennen. Entstehen dabei nur wenige Cluster, würde dies eher dafür sprechen dass die jeweiligen Cluster nur schwach zusammenhängen und gegebenenfalls mehrere Zentren ausbilden. Entsteht eine größere Zahl an Clustern,

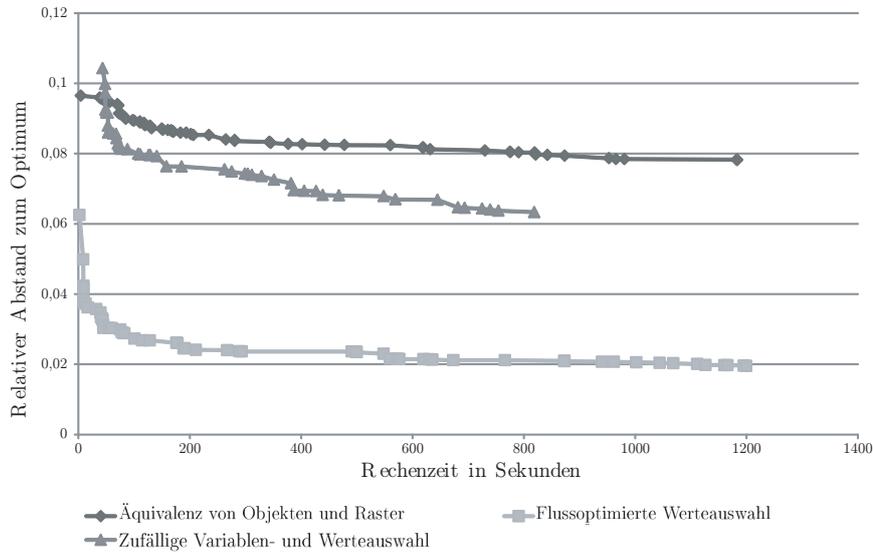


Abbildung 5.9.: Lösungssuche für die Instanz *wil100*

lassen sich solche Schwerpunkte nicht ausmachen und das Layout würde sich auch nicht in einzelne Bereiche aufteilen lassen.

5	3.088	3.724	3.720	4.286	3.698	4.850	3.808	3.032
4	3.562	3.376	5.892	5.370	3.784	4.406	3.886	2.690
3	2.270	4.694	5.462	5.066	5.924	5.908	4.478	2.484
2	3.968	4.632	4.026	5.338	5.524	4.616	3.974	2.712
1	2.556	4.228	2.082	3.954	3.724	2.060	3.078	1.694
	1	2	3	4	5	6	7	8

Abbildung 5.10.: Summe der ein- und ausgehenden Flüsse für *tho40*

### Vom QZP abgeleitete Instanzen

Diese Instanzen haben im Vergleich zur beabsichtigten Modellausrichtung die Schwierigkeit, dass das Raster durch die Blöcke vollständig gefüllt wird. In Abhängigkeit der Anzahl der Objekte und der Art und Weise der Aufteilung, kann der Lösungsraum unterschiedlich stark eingeschränkt sein und das Finden von Lösungen dadurch erschwert werden.

Für diese Instanzen kommt die Variablenauswahl auf Basis der Raumgröße und Domänengröße in Frage. Mit beiden Möglichkeiten lassen sich vergleichbare Ergebnisse erzielen. Dies lässt sich auch dadurch erklären, dass, solange der Wertebereich nicht durch zusätzliche Nebenbedingungen eingeschränkt wird, beide Ansätze auf eine ähnliche Reihenfolge hinauslaufen. Ausnahmen von dieser Regel treten beispielsweise für Objekte mit großem Seitenverhältnis auf. Diese können unter Umständen zwar die gesamte Länge des Rasters benötigen, haben aber gleichzeitig eine relativ geringe Anzahl an Blöcken, sodass sie bei einer Wertauswahl anhand der Raumgröße erst später berücksichtigt würden. Dieser Fall würde nach dem Propagieren durch eine Variablenauswahl aufgrund der Domänengröße früher erkannt und das Objekt entsprechend früh platziert.

Abbildung 5.11 zeigt unterschiedliche Varianten der Variablenauswahl unter Anwendung der konstruktiven Wertauswahl für die Instanz *sko64A*. Für diese Instanz war die Suche nach einer zulässigen Lösung verhältnismäßig einfach, für schwierigere Testdaten konnten sowohl mit zufälligen Variablen- als auch der zufälligen Objektauswahl innerhalb der gegebenen Zeit keine Lösungen mehr gefunden werden. Das die vorliegende Instanz relativ leicht zu lösen ist, ist insbesondere an der Tatsache zu erkennen, dass selbst mit der zufälligen Variablenauswahl, bei der mehrere Objekte gleichzeitig platziert werden, zumindest spät eine Lösung gefunden wurde. Zudem ist die Art und Weise der Wertauswahl ebenso hilfreich. Durch die Orientierung an bereits platzierten Objekten startet die Positionierung in den Ecken des Rasters und auch sich ergebende Eckpunkte innerhalb des Layouts werden bevorzugt gewählt. Da bei einer Platzierung in einem Eckpunkt der Rest des Objekts durch propagieren bestimmt werden kann, lassen sich selbst bei zufälliger Auswahl noch Lösungen finden.

Neben den in Abbildung 5.11 dargestellten Möglichkeiten der Variablenauswahl lassen sich über weitere Varianten einer gemischten Variablen- und Wertauswahl noch deutlich bessere Lösungen in kürzerer Zeit finden. Darauf wird bei der Diskussion der Ergebnisse für die Instanzen, die anhand der Modelanforderungen erstellt wurden, näher eingegangen. An den Ergebnissen aus Abbildung 5.11 lässt sich jedoch ferner erkennen, dass die zur Blockbildung führende konstruktive Wertauswahl unabhängig von der gewählten Strategie der Variablenauswahl relativ schnell zulässige Lösungen findet.

Für die Wertauswahl kommen jedoch prinzipiell alle im vorigen Kapitel diskutierten Möglichkeiten in Betracht. Eine flussoptimierte Platzierung ist jedoch problematisch, da sie verhältnismäßig oft zu Lücken im Layout führt, die gerade bei einer vollständigen Füllung des Rasters das Finden einer Lösung

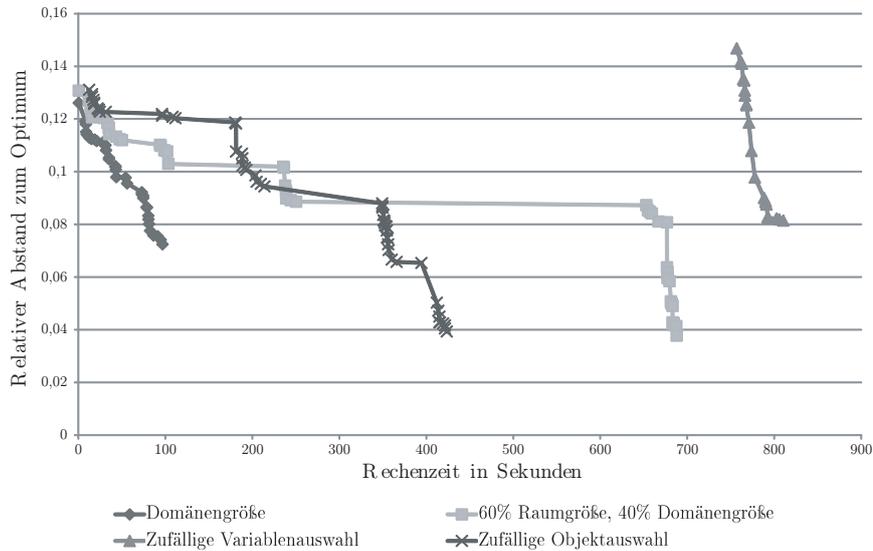


Abbildung 5.11.: Möglichkeiten der Variablenauswahl für *sko64A* bei Anwendung der konstruktiven Wertauswahl

erschweren. Entsprechend wird in diesen Fällen erst relativ spät oder gar keine zulässige Lösung gefunden. Ein ähnliches Problem tritt bei der Auswahl anhand der Äquivalenz von Objekt- und Rasterbewertung auf.

Sowohl die Auswahl des kleinsten Wertes der Domäne als auch die konstruktive Auswahl auf Basis bereits platzierter Objekte führen zu relativ kompakten Layoutformen, die das Platzieren weiterer Objekte erleichtern. Im zweiten Fall wurden Objekte so positioniert, dass ihre Maße möglichst mit bereits platzierten Objekten übereinstimmen und möglichst wenige Kanten entstehen. In der überwiegenden Zahl der Fälle erwies sich die genannte konstruktive Wertauswahl als die günstigste Variante.

Eine genaue Betrachtung der Eingangsdaten ist jedoch trotzdem notwendig. Dies lässt sich beispielsweise an den beiden Abbildungen 5.12 und 5.13 erkennen. Dort sind Ergebnisse mit unterschiedlichen Varianten der Wertauswahl für die Testdaten *sko64B* und *sko100A*, die aus 20 beziehungsweise 10 Objekten bestehen, dargestellt. Instanz *sko64B* enthält verhältnismäßig viele kleinere Objekte, die aus nur ein oder zwei Einheiten bestehen. Da diese bei der angewandten Variablenauswahl auf Basis von Objekt- und Domänengröße erst am Ende platziert werden, lassen sich damit eventuell entstandene Lücken im Layout auffüllen, so dass sich eine zulässige Lösung ergibt. Da

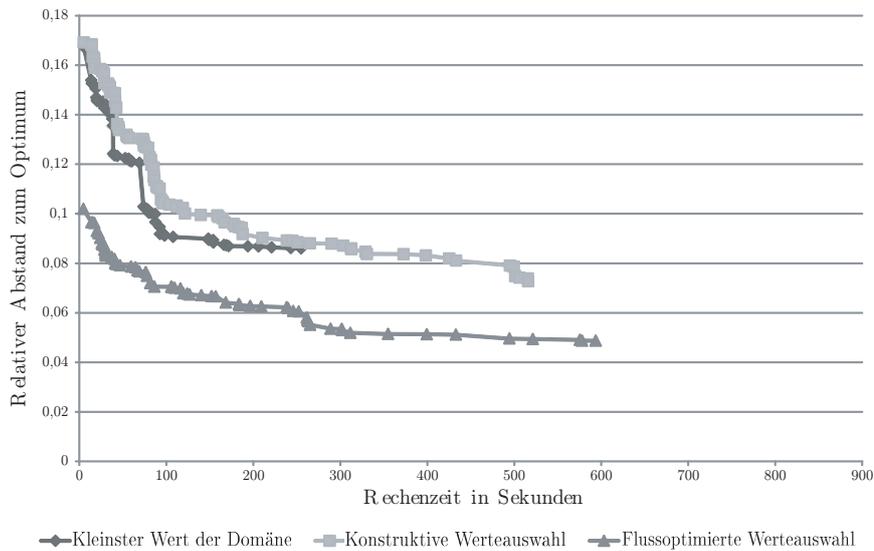


Abbildung 5.12.: Möglichkeiten der Werteauswahl für *sko64B* bei Variablenauswahl auf Basis von Objekt- und Domänengröße

es in diesen Fällen folglich relativ einfach ist, zulässige Lösungen zu bestimmen, ist die Anwendung der flussoptimierten Werteauswahl vorteilhaft, da Materialflüsse zwischen den Objekten stärker berücksichtigt werden und die gefundenen Lösungen daher eine höhere Qualität aufweisen.

### Instanzen alternativer Formulierungen

Instanzen des Single Row Facility Layout Problems lassen sich mit den vorgestellten Methoden nicht zufriedenstellend lösen. Zwar werden in kurzer Zeit Lösungen gefunden, jedoch weisen diese einen sehr großen Abstand zum Optimum auf (siehe Abbildung 5.14).

Die schlechten Ergebnisse lassen sich über die Art und Weise erklären, mit der Lösungen generiert werden. Nach der Positionierung des ersten Objekts, werden im Falle einer flussoptimierten Werteauswahl alle Objekte in einer Reihe an dieses Objekt anschließend platziert. Wird in einem Teilzweig keine zulässige Lösung gefunden oder existiert bereits eine bessere Lösung, wird für die gleiche Variable der zunächst gewählte Wert ausgeschlossen und der nächstbeste wird gewählt. Dies wird als Fehlschlag bezeichnet; es entsteht eine wachsende Lücke im Layout. Wenn bereits der vorige Zweig fehlgeschlagen ist, wird auch das neue Layout nicht erfolgreich sein. Dies gilt solange, bis die

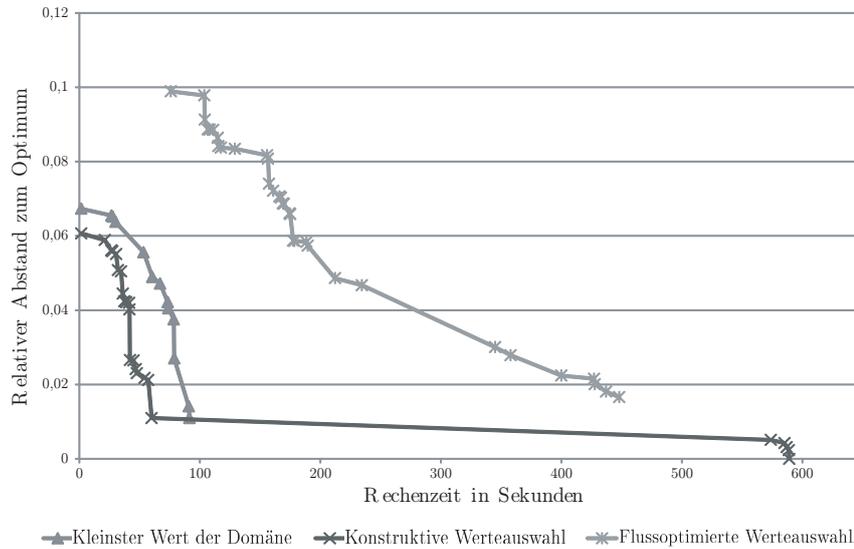


Abbildung 5.13.: Möglichkeiten der Werteauswahl für *sko100A* bei Variablenauswahl auf Basis von Objekt- und Domänengröße

Lücke groß genug ist, damit das nächste zu platzierende Objekt Platz findet. Nach Abschluss dieses Vorgehens haben die beiden Objekte ihren Platz getauscht und der Lösungszweig kann gegebenenfalls weiter untersucht werden. Natürlich ist ein solches Vorgehen, insbesondere bei größeren Objekten, sehr ineffizient. Diese Problematik bei der Konstruktion von Lösungen ist in Abbildung 5.15 zu sehen. In diesem Fall kann der erste Block von Objekt B zu Beginn auf allen Positionen von drei bis acht platziert werden. Nach einem Fehlschlag (beispielsweise da bereits eine bessere Lösung existiert) wandert der genannte Block jeweils eine Position weiter und es entsteht eine Lücke (in der Abbildung durch weiße Felder dargestellt). Nach dem dritten Fehlschlag wird Objekt B schließlich auf den Positionen sechs bis acht platziert und in der entstandenen Lücke findet Objekt C Platz.

Ein vergleichbares Problem kann in einem dreidimensionalen Layout und Objekten mit räumlicher Ausdehnung nicht auftreten. In diesen Fällen gibt es immer noch eine bessere Platzierungsmöglichkeit für ein Objekt, die keine Lücke im Layout entstehen lässt. Theoretisch ließe sich auch mit der Constraintprogrammierung ein Ansatz entwickeln, der dieser Problematik entgegenwirkt und mit dem sich Instanzen des SRFLP besser lösen ließen. Dies könnte beispielsweise dadurch geschehen, dass bei der Platzierung eines Objekts gleich die Maße des übernächsten Objekts (und wenn nötig auch der

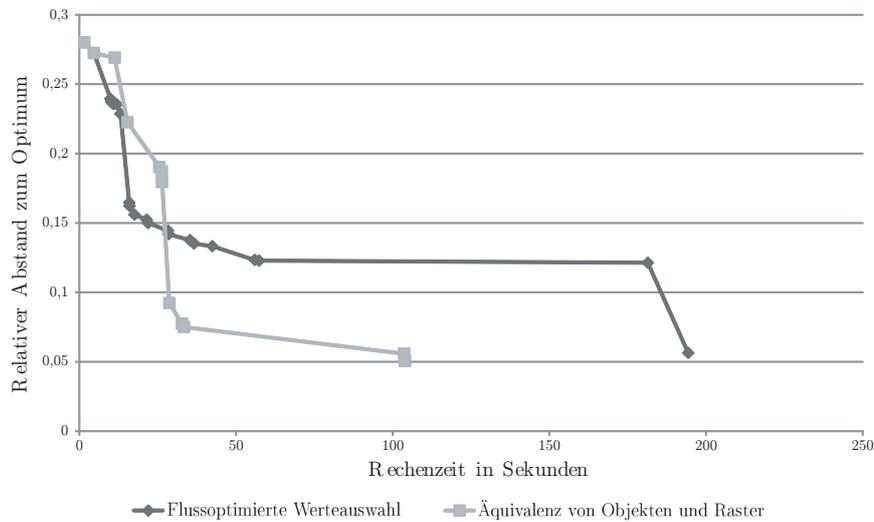


Abbildung 5.14.: Lösungssuche für die Instanz  $sim11_{SRFLP}$

darauf folgenden Objekte) betrachtet werden und die Werteauswahl direkt entsprechend große Lücken generiert. Da das SRFLP jedoch sehr weit vom ursprünglich angedachten Layoutproblem entfernt ist und eine ähnliche Problematik im mehrdimensionalen Raum nicht zu erwarten ist, soll ein solcher Ansatz in dieser Arbeit nicht weiter betrachtet werden.

Die beiden Instanzen  $baz12$  und  $baz14$  konnten jeweils innerhalb von ein bis zwei Sekunden gelöst werden. Der Zielfunktionswert lag dabei in einem Fall niedriger als das bei Bazaraa angegebene Optimum. Der Grund hierfür ist, dass bei Bazaraa eine mögliche Position des Objekts alle Einzelpositionen der Blöcke angibt, im vorliegenden Modell wird der Wertebereich eines einzelnen Blocks eingeschränkt. Dadurch kann es in Einzelfällen passieren, dass Blöcke eines Objekts Positionen belegen, die zwar die Abstandsbedingungen erfüllen, aber nach Bazaraas Modell zu verschiedenen Positionierungsmöglichkeiten gehören. Die Suchzeit lässt sich aufgrund des technischen Fortschritts, den unterschiedlichen Lösungsräumen und verschiedenen Modellierungszielen mit Bazaraa schlecht vergleichen, ist jedoch natürlich geringer.

### Instanzen entsprechend der Modellanforderungen

Für die erstellen Instanzen ließ sich in keinem Fall eine Lösung auf Basis einer zufälligen Objekt- oder Werteauswahl mehr finden. Zur Lösung wur-

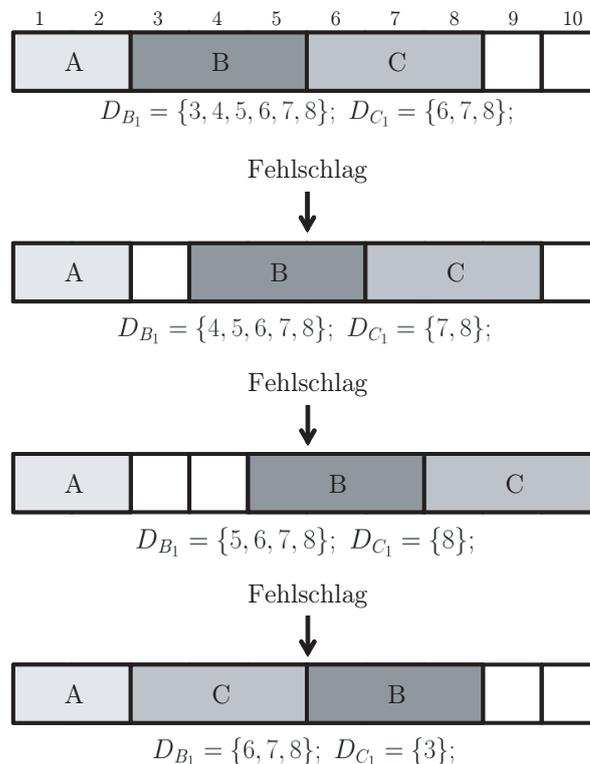


Abbildung 5.15.: Entstehen von Lücken bei der Lösungssuche für Instanzen des SRFLP

den vier Ansätze miteinander verglichen, bei denen für die Variablenauswahl jeweils eine gemischte Strategie gewählt wurde, bei der die Domänengröße einen Schwerpunkt bildete. Für die Positionierung wurde eine Mischung der konstruktiven und flussoptimierten Werteauswahl in Betracht gezogen.<sup>13</sup>

Hier sollen die Ergebnisse der vier Instanzen mit jeweils 20 Objekten betrachtet werden. Die mit  $a$  gekennzeichneten Instanzen besitzen einen Füllgrad des Rasters von 90%, die mit  $b$  bezeichneten Instanzen einen Füllgrad von 75%. Es zeigte sich, dass für die Instanzen mit niedrigerem Füllgrad deutlich leichter Lösungen gefunden wurden. Für die komplexer zu bestimmenden Instanzen

<sup>13</sup>Die jeweiligen Anteile sind in den Diagrammen angegeben; es wird zunächst die Variablenauswahl aufgeführt, anschließend die Werteauswahl. Dabei steht  $O$  für die Auswahl anhand der Objektgröße,  $D$  für Domänengröße,  $F$  für Flussgröße und  $N$  für Nebenbedingungen. Bei der Werteauswahl werden die Fälle konstruktive Werteauswahl ( $K$ ) und flussoptimierte Auswahl ( $F$ ) berücksichtigt.

## 5. Evaluierung der Ergebnisse

erwies sich ein höheres Gewicht für die Domänengröße als Vorteilhaft. In diesen Fällen wurden Lösungen deutlich schneller gefunden. Die Ergebnisse sind in den Abbildungen 5.16 bis 5.18 dargestellt.

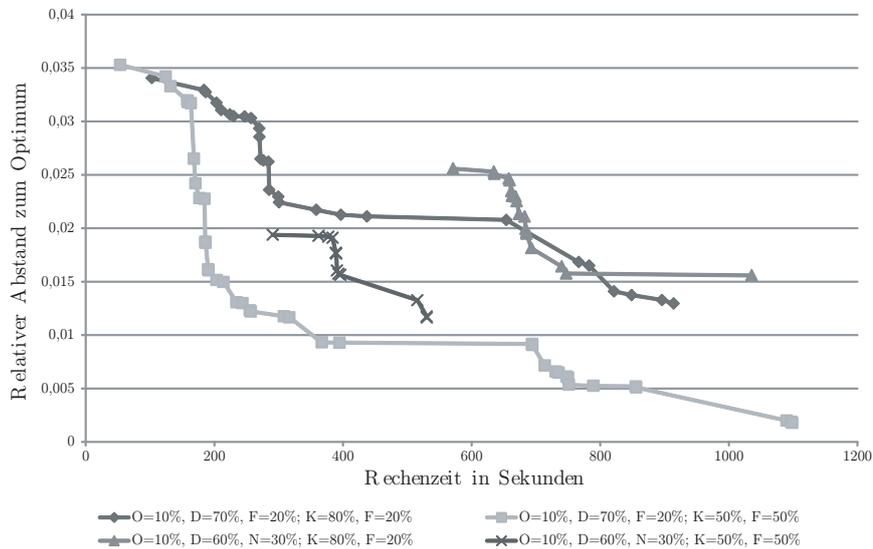


Abbildung 5.16.: Verlauf der Lösungssuche für unterschiedliche Strategien im Fall der Instanz *k20a2*

Anhand der Beobachtungen ist es jedoch schwer, pauschale Aussagen darüber zu treffen, in welchen Fällen eine bestimmte Strategie vorzuziehen ist. Beispielsweise ist es überraschend, dass für die Instanz *k20a2* unabhängig von der Variablenauswahl bei einer stärkeren Gewichtung des flussoptimierten Ansatzes früher zulässige Lösungen gefunden wurden. Ein Grund hierfür könnte sein, dass die Suche jeweils nach einer vorgegebenen Zahl an Fehlschlägen neu gestartet wird. Im angesprochenen Fall weisen die Strategien mit einer geringeren Berücksichtigung der konstruktiven Wertauswahl 10% beziehungsweise 40% mehr Fehlschläge innerhalb der Suchzeit auf. Entsprechend häufiger und früher wurde die Suche neu gestartet. Demnach ist es wahrscheinlich, dass aufgrund dieses häufigeren Neustarts die Suche erfolgreicher verlief. Es könnte folglich sinnvoll sein, ein Kriterium für das Neustarten der Suche zu definieren, das nicht nur von der Anzahl der Fehlschläge abhängt, sondern beispielsweise auch von der Suchtiefe, in der diese auftraten. Diese Möglichkeit besteht in der verwendeten Optimierungsumgebung nicht. Für die größeren Instanzen mit einem Füllgrad von 90% traf in der überwiegenden Mehrheit der Fälle die Annahme zu, dass sich über die konstruktive

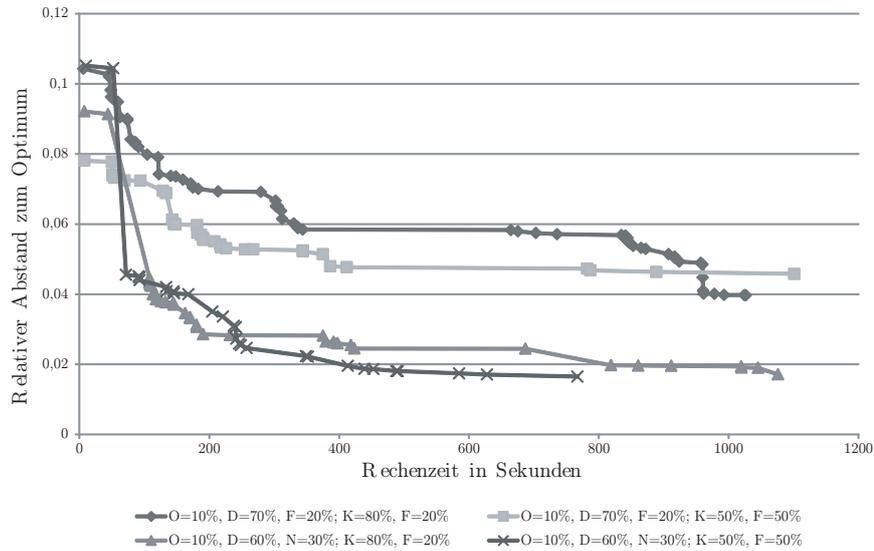


Abbildung 5.17.: Verlauf der Lösungssuche für unterschiedliche Strategien im Fall der Instanz *k20b1*

Werteauswahl wesentlich schneller Lösungen finden lassen. Bei einigen Berechnungen ließ sich mit dem geringeren Gewicht dieses Ansatzes keine Lösung innerhalb der gegebenen Suchzeit finden.

Ferner konnte beobachtet werden, dass die gemischten Ansätze, die mehrere Kriterien in der Suche berücksichtigen, erfolgreicher waren, als bei einer Begrenzung auf ein einzelnes Kriterium.

Da die Ergebnisse auf dieser Basis nur bedingt eine Aussage über die Auswirkungen der einzelnen Suchstrategien zulassen, wurde für die Instanz *k20a2* nochmals eine detailliertere Betrachtung mit einer größeren Zahl an unterschiedlichen Strategien vorgenommen. Jedoch ließ auch dieses Vorgehen keine weiteren grundsätzlichen Aussagen zu.

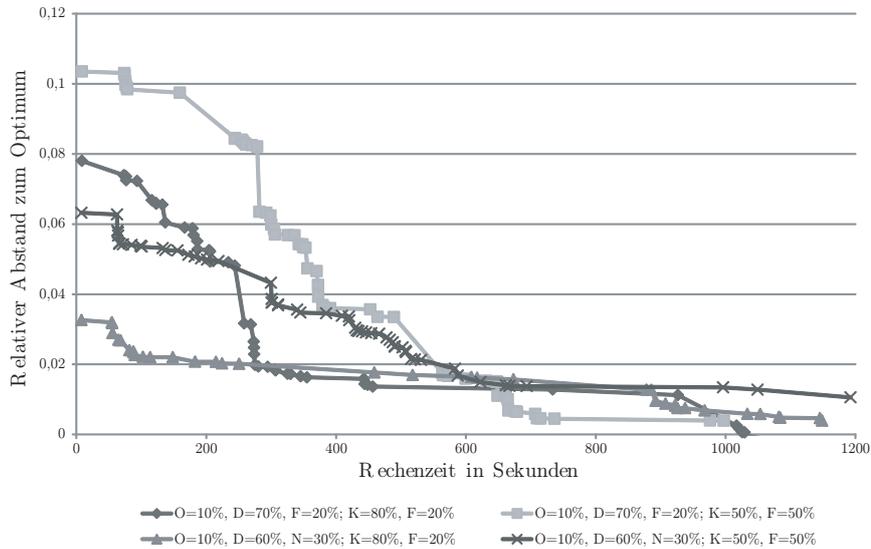


Abbildung 5.18.: Verlauf der Lösungssuche für unterschiedliche Strategien im Fall der Instanz *k20b2*

### 5.2.6. Zusammenfassung der Ergebnisse

Für die verschiedenen Arten von Instanzen ergeben sich sehr unterschiedliche Ergebnisse. Dabei erweisen sich die vorgestellten Methoden für die Instanzen, die anhand der Modellvorgaben erstellt wurden, am wirksamsten. Zwar lassen sich für alle Datensätze Lösungen unterschiedlicher Qualität finden, jedoch existieren offensichtlich für die Instanzen der unterschiedlichen Modelle bessere Lösungsalternativen. Es kann jedoch letztlich auch nicht der Anspruch sein, die Ergebnisse tatsächlich mit den jeweils besten Methoden zu vergleichen. Es sollte lediglich gezeigt werden, dass das vorgestellte Modell eine hohe Flexibilität besitzt, um auch mit unterschiedlichen Eingangsdaten umzugehen.

Für die Instanzen des QZP wurden innerhalb von einigen Minuten Lösungen gefunden, die je nach Instanzgröße bis zu vier Prozent vom Optimum abweichen. Ähnliches gilt für davon abgeleitete Testdaten, die je nach der Art der Teilung unterschiedlich komplex zu lösen waren, für die jedoch nach kurzer Zeit Lösungen nahe des Optimums oder das Optimum selbst gefunden werden konnte. Für die Ursprungsinstanzen erwies sich die Variablenauswahl auf Basis der Summe der ein- und ausgehenden Flüsse in Kombination mit einer flussoptimierten Wertauswahl als bester Ansatz. Für die daraus abgeleiteten

Datensätze wurden die Variablen anhand von Objektgröße und Domänengröße ausgewählt. Für die Platzierung zeigte das Bilden von Blöcken, wie im Falle der konstruktiven Wertauswahl anhand bereits positionierter Objekte die besten Ergebnisse.

Datensätze, denen alternative Formulierungen des Layoutproblems zugrunde liegen, führten zum Teil zu Abweichungen, die aus den unterschiedlichen Formulierungen resultieren und deren Annahmen sich nicht exakt nachbilden lassen. Instanzen, die dem Modell von Bazaraa folgen, lassen sich mit dem vorliegenden Ansatz sehr einfach lösen. Die Instanzgröße war jedoch in diesen Fällen nicht so groß, dass sich signifikante Unterschiede durch die Auswahl verschiedener Lösungsmethoden beobachten ließen. Instanzen des SRFLP ließen sich nur relativ schlecht lösen, der Grund hierfür ist die Art und Weise wie die Position zweier Objekte in einer Linie getauscht wird und die sehr ineffizient ist. Ähnliche Probleme lassen sich jedoch für mehrdimensionale Fälle nicht erwarten.

Für alle Instanzen mit schwankenden Objektgrößen sowie höherem Füllgrad des Rasters, erwies sich das Propagieren möglicher Positionen als sinnvoll und zum Teil als zwingend notwendig. Ebenso ist es nicht sinnvoll, mehrere Objekte zeitgleich zu positionieren. Die dargestellte untere Schranke erwies sich für das Modell als nicht wirksam und bedeutet letztlich nur zusätzlichen Rechenaufwand. Auch das Propagieren der Nachfrage nach einer Position zeigte in der dargestellten Form nur in wenigen Fällen Wirkung. Allerdings scheint der Grundgedanke durchaus sinnvoll und eine Ausweitung des Propagierens auf mehr als zwei konkurrierende Objekte sowie gleichzeitig ein gezielterer Einsatz zur Reduktion der Komplexität ist empfehlenswert.

Bei den für die Variablen- und Wertauswahl dargestellten Methoden konnte für die Instanzen, die anhand der Modellanforderungen erstellt wurden folgende Beobachtungen gemacht werden:

- Die gewählten Strategien sind grundsätzlich hilfreich bei der Suche nach zulässigen Lösungen; auf Basis einer zufälligen Auswahl der Objekte ließen sich keine Lösungen finden.
- Gemischte Strategien erwiesen sich gegenüber reinen Strategien als vorteilhaft. Die Domänengröße der Variablen sollte dabei ein signifikantes Gewicht aufweisen.
- Insbesondere bei komplexeren Instanzen stellt die konstruktive Wertauswahl in der Regel den am Besten geeigneten Ansatz zum Finden einer Lösung dar.
- Die konstruktive Wertauswahl führt in der Regel zu einer geringeren

Zahl an Fehlschlägen innerhalb der gleichen Suchzeit. Die Suche wird seltener neu gestartet.

Darüber hinaus ist es jedoch nicht einfach generelle Aussagen zur Auswahl einer geeigneten Suchstrategie zu treffen. Zwar bestehen Tendenzen, beispielsweise dass höhere Schwankungen der Materialflüsse Ansätze die auf denselben basieren vorteilhafter werden lassen, jedoch hängt dies von weiteren Faktoren ab, die nicht ohne weiteres zu bestimmen sind.

### 5.3. Auswahl der Suchstrategie

Die unterschiedlichen Alternativen zur Auswahl von Variablen und Werten können je nach Charakteristik der Ausgangsdaten alle einen Mehrwert bringen. Zu diesem Zweck ist es jedoch notwendig nochmals die angesprochenen Merkmale der Instanzen zu betrachten und aus diesen eine passende Strategie abzuleiten. Dabei wurden die wesentlichen Gesichtspunkte bereits genannt. Dazu zählen unter anderem die Größe der Objekte oder der Füllgrad des Rasters. Wichtig ist hierbei nun mehr, ein Maß für die einzelnen Merkmale zu finden, anhand derer sich Unterschiede quantifizieren lassen.

#### 5.3.1. Merkmale der Datensätze

Für die Instanzen wurden eine Reihe von Größen identifiziert, die einen Einfluss auf die Suchstrategie haben können. Zum Teil kann dabei auch erst das Zusammenwirken mehrerer Faktoren für eine bestimmte Strategie sprechen. Im Folgenden sollen die Merkmale angesprochen und ihre Einflussmöglichkeiten auf die Suchstrategie kurz skizziert werden.

##### **Anzahl der Objekte**

Dieses Merkmal beeinflusst wesentlich die Komplexität einer Instanz. Bleiben alle sonstigen Parameter gleich, lässt sich für eine Instanz mit vielen Objekten leichter eine zulässige Lösung finden; auf der anderen Seite ist der Lösungsraum jedoch größer.

##### **Durchschnittliche Objektgröße**

Dieses Merkmal kann bei der Entscheidung, ob die Propagierungsverfahren für einen bestimmten Datensatz angewendet werden sollten, hilfreich sein. Bei

steigender Objektgröße wird ohne dieses Vorgehen nur schwer eine zulässige Lösungen zu bestimmen sein. Ferner ist ein solches Merkmal auch zum Vergleich mit bereits berechneten Instanzen sinnvoll. In diesen Fällen beeinflussen Objektgröße und Objektanzahl die Komplexität einer Instanz und eine Berücksichtigung ist notwendig, um für das gegebene Zeitbudget eine sinnvolle Strategie wählen zu können.

### **Schwankung der Objektgröße**

Als Maß für dieses Merkmal dient die Variabilität der Anzahl der Blöcke pro Objekt. Es hilft bei der Entscheidung ob ein Propagieren der Objekte und deren möglicher Positionen notwendig ist. Ist die Variabilität gleich null, besitzen alle Objekte die gleiche Größe und sie können identische Positionen im Raster einnehmen. Das Propagieren wäre in diesem Fall überflüssig. Ebenso besitzt es Einfluss auf die Variablen- und Wertauswahl. Eine Variablenauswahl anhand der Objektgröße kann insbesondere bei hohen Schwankungen und hohem Füllgrad des Rasters sinnvoll sein. Auch die Frage, ob eine Positionierung anhand der Äquivalenz von Objekt- und Rasterbewertung sinnvoll sein kann und mit welchem Gewicht die Objektgröße dabei berücksichtigt werden sollte, hängt von diesem Merkmal ab.

### **Schwankung der Objektmaße**

Neben der Schwankung der Objektgröße kann auch die Vielfalt unterschiedlicher Objektmaße von Bedeutung sein. Liegen viele identische Maße vor, kann das Entstehen von Kanten im Layout besser vermieden werden und das Finden einer zulässigen Lösung wird vereinfacht. Zur Messung werden die Maße aller Objekte je nach ihrer Größe in unterschiedliche Klassen eingeteilt. Die maximale Anzahl an Klassen entspricht dabei dem Maximum aus Länge, Breite und Höhe des Rasters. Als Maß der Konzentration wird anschließend der Gini-Koeffizient  $\Gamma$  auf die relative Häufigkeit  $q_i$  der einzelnen Klassen angewendet. Für das Maß gilt somit:

$$\Gamma = \sum_i q_i^2 \tag{5.1}$$

Die Gesamtzahl an Beobachtungen, die für die Bestimmung der relativen Häufigkeiten  $q_i$  notwendig ist, entspricht der dreifachen Anzahl der Objekte, da jedes Objekt je ein Maß zur Angabe der Länge, Breite und Höhe besitzt.

### **Füllgrad des Rasters**

Dieses Merkmal gibt an, zu wie viel Prozent das verfügbare Raster mit den Objekten gefüllt wird. Ein hoher Füllgrad kann das Finden einer zulässigen Lösung erschweren. Dies gilt insbesondere bei einer stark schwankenden Objektgröße und Schwankungen der Objektmaße. In diesen Fällen sollten die möglichen Positionen der Objekte propagiert werden, um den Lösungsraum einzugrenzen. Zudem kann das Merkmal für die Anwendung einer konstruktiven Wertauswahl sprechen, um die Platzierung der weiteren Objekte zu erleichtern.

### **Schwankung der Materialflüsse**

Sofern das Finden einer zulässigen Lösung verhältnismäßig einfach, beziehungsweise wenn die gegebene Suchzeit ausreichend lange ist, ist neben dem Finden einer zulässigen Lösung, die Suche nach einer möglichst guten Lösung von Interesse. In diesen Fällen können die jeweiligen Flussgrößen bei der Auswahl von Objekten und ihrer anschließend Platzierung im Raster relevant sein. Diese Ansätze werden umso vorteilhafter, je stärker die auftretenden Materialflüsse schwanken. Zu diesem Zweck wird dieses Merkmal mit Hilfe der Variabilität gemessen.

### **Durchschnittliche Bewertung der Nebenbedingungen**

Die auftretenden Nebenbedingungen schränken die Wertebereiche der jeweiligen Variablen ein und begrenzen auf diese Weise den Lösungsraum. Je mehr Nebenbedingungen auftreten, desto schwieriger wird folglich die Suche nach einer zulässigen Lösung. In diesen Fällen ist es sinnvoll, stark restringierte Objekte frühzeitig zu platzieren, da diese im Laufe der Suche am ehesten zu einem Fehlschlag führen können. Für dieses Merkmal wird auf die Bewertung der Nebenbedingungen, wie sie in Abschnitt 4.1.1 beschrieben wurde, zurückgegriffen. In diesem Rahmen wurden alle Nebenbedingungen betrachtet, und die betroffenen Variablen entsprechend bewertet. Als Maß, wie stark das Layout durch Nebenbedingungen begrenzt wird, wird die durchschnittliche Bewertung der Variablen verwendet.

### 5.3.2. Einfluss der Kenngrößen auf die Suchstrategie

Eine eindeutige Aussage, die einen klaren Bezug zwischen den dargestellten Kenngrößen und einer daraus resultierenden Suchstrategie herstellt, ist aufgrund der vielen unterschiedlichen Möglichkeiten bei der Suche und der unterschiedlich starken Auswirkung auf das Ergebnis, kaum zu treffen. Beispielsweise ist die Berücksichtigung der Nebenbedingungen in der Suchstrategie zwar durchaus sinnvoll, als Hauptmerkmal sind sie jedoch ungeeignet. Als Beispiel sei hierfür ein Layout betrachtet, in dem eine Reihe kleinerer Objekte über Abstandsbedingungen in Beziehung stehen, sowie ein einzelnes größeres Objekt enthalten ist, das jedoch an keinerlei Bedingungen geknüpft ist. Würde die Bewertung der Nebenbedingungen als Hauptkriterium herangezogen, würden immer zunächst die kleineren Objekte positioniert, das einzelne große Objekt folgt am Schluss und ist entsprechend schwierig zu platzieren. Offenbar sollte also das Gewicht, mit dem die Nebenbedingungen in die Strategie eingehen, begrenzt werden. Da eine sinnvolle Aussage über eine solche Grenze jedoch kaum zu treffen ist, sollen im Folgenden Wechselwirkungen zwischen den Merkmalen und deren Einfluss auf die Suchstrategie zunächst auf Basis der Beobachtungen qualitativ betrachtet werden. Diese sind in Tabelle 5.5 dargestellt. Dabei bedeutet ein +, dass eine starke Ausprägung des Merkmals für eine stärkere Berücksichtigung eines bestimmten Ansatzes in der Suchstrategie spricht, ein – entsprechend einen gegenläufigen Effekt. Mit o werden Beziehungen gekennzeichnet, bei denen keine Auswirkungen beobachtet werden. Beachtet werden muss dabei, dass es sich bei den beiden Propagierungsvarianten um binäre Entscheidungen handelt, bei den Parametern der Variablen- und Wertauswahl um Gewichtungen. Diese Zusammenhänge wurden beim Zusammenstellen der getesteten Suchstrategien beobachtet und sollen einen Anhaltspunkt geben, in welchen Fällen welcher Aspekt beachtet werden sollte.

Alternativ wurde eine Möglichkeit implementiert, die unter Bezug auf die bereits berechneten Instanzen eine geeignete Strategie ableiten soll. Zu diesem Zweck wurde ein Ähnlichkeitsmaß definiert, das die zu berechnende Instanz mit den bisherigen Daten auf Basis der genannten Merkmale vergleicht und die Instanz mit der größten Ähnlichkeit auswählt. Unter Berücksichtigung eines vom Anwender gesetzten Zeitbudgets wird anschließend aus den auf diese Instanz angewendeten Strategien diejenige ausgewählt, die für diesen Fall die besten Ergebnisse erbracht hat.

Zu diesem Zweck werden zunächst die einzelnen Ausprägungen  $u_{i,j}$  jeder Instanz  $i$  und jedes Merkmals  $j$  anhand von Mittelwert und Standardabweichung

Tabelle 5.5.: Einfluss der Merkmale auf Aspekte der Suchstrategie

	Anzahl Objekte	Durchschnittliche Objektgröße	Schwankungen der Objektgröße	Schwankungen der Objektmaße	Füllgrad des Rasters	Schwankungen der Materialflüsse	Bewertung der Nebenbedingungen
Propagieren möglicher Positionen	-	+	++	++	++	o	+
Propagieren der Objekte	-	++	++	++	++	o	+
Parameter Variablenauswahl							
Wertebereich	-	+	++	++	++	o	++
Objektgröße	+	o	+	+	+	o	o
Bewertung Nebenbedingungen	o	o	o	o	o	o	++
Bewertung Flüsse	o	o	o	o	o	++	o
Parameter Wertauswahl							
Äquivalenz Objekt- und Rasterbewertung	+	-	+	+	-	+	o
Konstruktive Wertauswahl	-	o	++	-	++	o	o
Flussoptimierte Wertauswahl	+	-	-	-	-	++	o

normalisiert. Für die sich daraus ergebenden Werte  $\tilde{u}_{i,j}$  wird die Ähnlichkeit der betrachteten Instanz  $k$  zu jeder bereits bestimmten Instanz  $i$  ermittelt und das Maximum ausgewählt. Als Ähnlichkeitsmaß wird die euklidische Distanz gewählt:

$$\max_{i \in I} \sqrt{\sum_{j \in J} (\tilde{u}_{k,j} - \tilde{u}_{i,j})^2} \quad (5.2)$$

Aus den zu der gewählten Instanz  $k$  gehörenden Lösungsstrategien wird anschließend, je nachdem, ob der Fokus bei der Suche auf einer zulässigen Lösung, oder einer möglichst guten Lösung liegen soll, die für das gegebene Zeitlimit beste Strategie ausgewählt.

# 6. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Modell zur dreidimensionalen Layoutplanung großtechnischer Anlagen entwickelt. Besondere Rücksicht sollte dabei auf deren speziellen Eigenschaften und Nebenbedingungen genommen werden. Den Modellanforderungen entsprechende Testinstanzen existieren in der Literatur bisher nicht. Aus diesem Grund wurde der Lösungsansatz auf mehrere Instanzen des quadratischen Zuordnungsproblems sowie einige davon abgeleitete Testdaten angewendet. Daneben wurde ebenso auf Daten einiger ähnlicher Layoutplanungsprobleme wie dem sogenannten Single Row Facility Location Problem zurückgegriffen. Zuletzt wurden eigene Instanzen entwickelt und die unterschiedlichen Ansätze darauf angewendet.

Im Folgenden sollen zunächst nochmals die Ergebnisse im Hinblick auf die in der Einleitung vorgestellten Fragenkomplexe zusammengefasst werden. Da in dem betrachteten Forschungsfeld auf verhältnismäßig wenig bestehende Literatur zurückgegriffen werden kann, sollen darüber hinaus noch einige Ansatzpunkte für Weiterentwicklungen angegeben werden. Dabei soll der Fokus wiederum auf dem praktischen Bezug liegen.

## 6.1. Zusammenfassung

### **Fragenbereich 1: Eigenschaften existierender Modelle**

Kapitel 2 grenzte zunächst unterschiedliche Arten der Layoutplanung anhand einiger Kriterien voneinander ab. Dabei wurde unter anderem auf die Größe und Form der Einheiten, die zeitliche Komponente bei der Planung oder auch die Anzahl der Dimensionen eingegangen. Ebenso wurde kurz aufgezeigt, welche Arten von Nebenbedingungen in der Praxis eine Rolle spielen können. Im Anschluss wurden die verbreitetsten Modelle kurz erläutert und ihre konkrete Formulierung skizziert. Auf Basis der identifizierten Charakteristika kann die Eignung der Modelle für den beabsichtigten Anwendungsfall bewertet werden.

### **Fragenbereich 2: Anforderungen an ein Modell**

Die zum Teil stark variierenden Raumgrößen sowie die Vielfalt der auftretenden Nebenbedingungen wurden in Kapitel 3 als ein wesentliches Kriterium bei der Planung großtechnischer Anlagen identifiziert. Neben diesen beiden Eigenschaften sollen Materialflüsse zudem an unterschiedlichen Punkten eines Objekts beginnen oder enden können. Die Annahme, dass ein zentraler Ort innerhalb eines Objekts eine hinreichend genaue Näherung darstellt, gilt somit nicht. Diese Eigenschaften weichen vom Großteil bestehender Modelle entscheidend ab. Aus diesem Grund wurde ein auf dem quadratischen Zuordnungsproblem basierendes Modell entwickelt, das mit Hilfe der Constraint Programmierung (CP) gelöst wurde. Auf diesem Wege besteht auch die notwendige Flexibilität zur Abbildung unterschiedlicher Nebenbedingungen.

Zur Abbildung dieser technischen und wirtschaftlichen Restriktionen wurden vielfältige unäre und binäre Bedingungen dem Modell hinzugefügt. Diese bestimmen beispielsweise die Ausrichtung von Objekten zueinander oder deren Abstand. Ebenso ist es möglich, Objekte einem vordefinierten Bereich im Raster zuzuordnen.

### **Fragenbereich 3: Aufbau eines Suchalgorithmus**

Die Suchzeit eines CP-Problems wird im Wesentlichen durch die verwendeten Strategien zur Variablen- und Werteauswahl bestimmt. Für beide Aspekte wurden in Kapitel 4 mehrere Verfahren vorgestellt. Dazu wurden zunächst einige Merkmale der zu platzierenden Objekte genannt, auf denen Entscheidungen im Laufe des Suchprozesses basieren können. Dabei wurde unter anderem die Möglichkeit zur Bewertung von Constraints oder der ein- und ausgehenden Materialflüssen genannt. Im Suchprozess sollen Objekte, die für die Lösung kritisch sind, möglichst früh berücksichtigt werden. Dies kann sowohl für die Zulässigkeit einer Lösung im Rahmen einer Nebenbedingung gelten als auch für die Güte der Lösung bei Objekten mit einer hohen Summe an ein- und ausgehenden Materialflüssen. In beiden Fällen wirkt sich eine späte Berücksichtigung mit hoher Wahrscheinlichkeit negativ aus.

Bei der Werteauswahl wurden für unterschiedliche Charakteristiken der Eingangsdaten eigene Lösungsansätze entwickelt. Für Problemstellungen, die über ihre Nebenbedingungen nur schwach restringiert werden, wurde eine flussoptimierende Positionierung der Objekte beschrieben. Dem gegenüber wird bei heterogenen Raumgrößen und einem hohen Füllgrad des Rasters eher eine Blockbildung bevorzugt, die die Platzierung weiterer Objekte er-

leichtert. Ein weiterer Ansatz berücksichtigt mit der Objektgröße und den Flussgrößen sowohl Aspekte der Zulässigkeit als auch der Optimalität. Im Vordergrund steht dabei eine Äquivalenz von Objekt- und Rasterbewertung. Die Ansätze erwiesen sich dabei als grundsätzlich geeignet, jedoch ist es schwierig eine präzise Aussage zu treffen, aus welchen Kriterien sich der ideale Suchalgorithmus für eine Instanz zusammensetzen sollte.

#### **Fragenbereich 4: Reduktion des Lösungsraums**

Neben einer Reihe von unären und binären Nebenbedingungen wurden auch einige Constraints vorgestellt, die intern für eine Reduktion des Lösungsraums sorgen sollen. Dies umfasst beispielsweise eine Nebenbedingung, die im Laufe der Suche die vorhandenen Freiräume untersucht und mit der Größe der noch zu platzierenden Objekten abgleicht. Ebenso wurde ein im Rahmen des quadratischen Zuordnungsproblems verwendeter Ansatz einer unteren Schranke auf die Problemstellung angepasst.

Dabei erwies sich das Propagieren der Objekte sowie das Propagieren der möglichen Positionen eines Objekts bis auf wenige Ausnahmen als sinnvoll. Nicht eingesetzt werden sollten diese Methoden bei Instanzen, in denen alle Objekte die gleiche Größe aufweisen. Das Propagieren der Nachfrage unterschiedlicher Objekte nach einer bestimmten Position erwies sich nicht als vorteilhaft. In diesen Fällen übersteigt die notwendige Rechenzeit regelmäßig den Nutzen, da eine solche Situation auch ohne Propagieren kurze Zeit später festgestellt würde. Ferner war in Anbetracht der betrachteten Instanzgrößen auch die dargestellte untere Schranke ohne nennenswerte Wirkung. Diese ist in fast allen Fällen zu weit von der optimalen Lösung entfernt, um Suchzweige ausschließen zu können und sich auf diese Weise positiv auf die Rechenzeit auszuwirken. Auch scheinen alternative untere Schranken des QZP nur als wenig aussichtsreich.

#### **Fragenbereich 5: Einfluss unterschiedlicher Parameter**

Es wurde eine Reihe von Merkmalen identifiziert, die geeignet sind einen gegebenen Datensatz zu charakterisieren. Dies umfasst beispielsweise den Füllgrad des Rasters, aber auch Schwankungen der Materialflüsse oder Objektmaße. Es wurde für diese Merkmale ferner qualitativ angegeben, welche Strategien durch sie begünstigt werden und auf welche sie sich tendenziell negativ auswirken.

Zur Auswahl einer Strategie wurde darüber hinaus ein Ansatz implementiert, der die Ähnlichkeit einer Instanz mit bestehenden Datensätzen vergleicht und auf dieser Basis eine Empfehlung trifft.

### 6.2. Ausblick

Auch unter Berücksichtigung der in Abschnitt 2.4 vorgestellten dreidimensionalen Layoutplanungsmodelle existiert nur verhältnismäßig wenig Literatur, die sich mit dem behandelten Forschungsgebiet beschäftigt. Dies liegt neben der hohen Komplexität, die die Verwendung analytischer Verfahren einschränkt, sicherlich auch an den begrenzten Möglichkeiten zur Anwendung. Jedoch lassen sich nicht alle Layoutplanungsaufgaben auf zweidimensionale Modelle zurückführen, sodass für diese Anwendungsbereiche zwingend eigene Verfahren benötigt werden.

In dieser Arbeit wurde ein flexibler Ansatz präsentiert, der neben verschiedenen anderen Arten von Layoutmodellen speziell zur Lösung von Problemen, die durch eine heterogene Objektgröße und eine große Vielfalt zusätzlicher Nebenbedingungen gekennzeichnet ist. Auch die vorgestellten Lösungsverfahren wurden unter diesem Gesichtspunkt entworfen.

Da derartige Probleme bisher nur wenig Aufmerksamkeit erhielten, kann diese Arbeit nur ein erster Ansatz zu deren Lösung sein. Dabei erwies sich das vorgestellte Modell unter Verwendung der Constraint Programmierung als Lösungsverfahren als grundsätzlich geeignet. Darauf aufbauend lassen sich jedoch einige Möglichkeiten aufzeigen, die zu einer effizienteren Lösung beitragen können oder auch den Ansatz dem praktischen Einsatzgebiet noch weiter annähern würden.

Vielversprechend könnte dabei unter anderem eine weitere Betrachtung von Constraints sein, die möglichst frühzeitig Lösungen ausschließen beziehungsweise die Wertebereiche der Variablen noch stärker eingrenzen. Die zu diesem Zweck entwickelten Bedingungen haben positive Ergebnisse gezeigt. Generell ist jegliche Einschränkung der Wertebereiche wünschenswert, jedoch kann der dazu notwendige Rechenaufwand die resultierende Ersparnis wieder egalisieren. Dies konnte zum Beispiel an einer Nebenbedingung beobachtet werden, die Positionen identifizieren sollte, die bei der Suche zum Engpass werden könnten. Diese Bedingung war so aufwendig zu berechnen, dass sie sich nicht positiv auf die Rechenzeit auswirken konnte. Eine Einschränkung des Constraints zur Reduktion des Aufwands führte dazu, dass diese Bedingung nur selten den Suchraum tatsächlich einschränkte. Eine weitere Möglichkeit könn-

te sein, aufwendige Constraints nur gezielt einzusetzen; dazu müsste ein entsprechendes Kriterium gefunden werden.

Je nach Charakteristik der Eingangsdaten kann neben der Zulässigkeit der Lösung auch die Optimalität der Lösung im Vordergrund stehen. Zu diesem Zweck ist eine verbesserte untere Schranke notwendig. Viele Verfahren, die im Rahmen des quadratischen Zuordnungsproblems entwickelt wurden, sind für dieses Ziel ungeeignet, da sie die Einschränkungen des Lösungsraums über Nebenbedingungen unberücksichtigt lassen und dadurch die Lücke zum optimalen Ergebnis potentiell vergrößern. Eine weitere Schwierigkeit liegt dabei in der Größe der Instanzen, da bekannte Verfahren mit wachsender Problemgröße an Qualität einbüßen. Eine Möglichkeit könnte dabei eventuell der Rückgriff auf untere Schranken stetiger Layoutplanungsmodelle sein.

Auch ein Ansatz, der die zur Verfügung stehenden innerbetrieblichen Transportmittel stärker berücksichtigt, ließe sich auf Basis des vorgestellten Modells entwickeln. Diese finden bisher nur über die Nebenbedingungen Eingang. So könnte ein Modell erstellt werden, das, aufbauend auf einer möglichen Lösung des Layoutproblems, das kostengünstigste Transportnetz bestimmt, mit dem die jeweiligen Angebots- und Bedarfsorte verbunden werden können. In einem zweiten Schritt ließe sich dieses Netz unter Berücksichtigung der Anforderungen des Layoutmodells und der Randbedingungen der Fördermittel verbessern. Eine Verbesserung könnte in diesem Fall beispielsweise darin liegen, Überschneidungen von Transportwegen zu vermeiden oder die Anzahl der Richtungsänderungen eines Weges zu reduzieren. Die daraus resultierenden Arbeitsbereiche der Fördermittel können anschließend wiederum als eigener Bereich im Layout definiert werden und somit wieder Eingang in die Layoutplanung finden. Auf diese Weise würde ein iteratives System entstehen, bei dem aus jeder Berechnungsstufe mehrere Lösungen generiert werden können. Aus diesen kann für den nächsten Schritt eine sinnvolle Auswahl getroffen (siehe Abbildung 6.1).

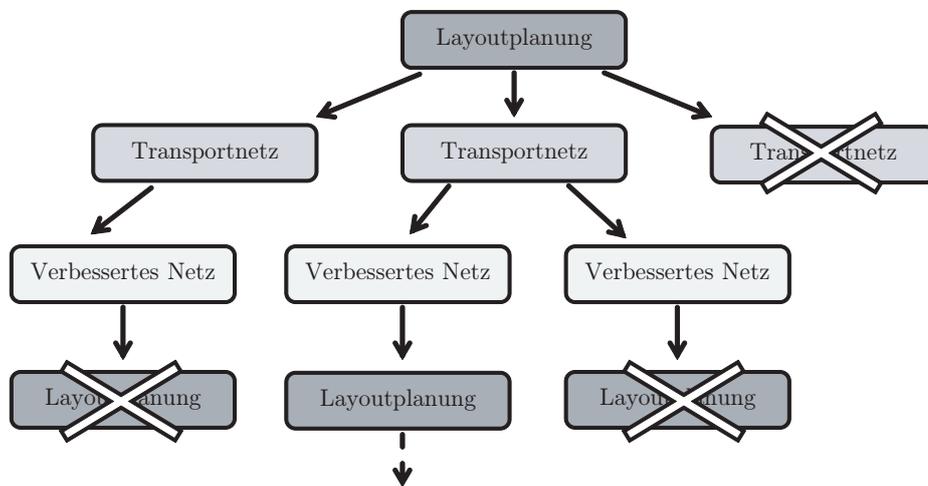


Abbildung 6.1.: Iterative Planung von Layout und Fördersystemen

# Literatur

- Abdou, G. und S. P. Dutta (1990). An integrated approach to facilities layout using expert systems. *International Journal of Production Research* 28(4), S. 685–708.
- Adams, W. P., M. Guignard, P. M. Hahn und W. L. Hightower (2007). A level-2 reformulation linearization technique bound for the quadratic assignment problem. *European Journal of Operational Research* 180(3), S. 983 – 996.
- Anjos, M. F. und A. Vannelli (2006). A new mathematical-programming framework for facility-layout design. *INFORMS Journal on Computing* 18(1), S. 111–118.
- Anstreicher, K. M. und N. W. Brixius (2001). A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming* 89(3), S. 341–357.
- Arbelaez, A., Y. Hamadi und M. Sebag (2009). Online Heuristic Selection in Constraint Programming. International Symposium on Combinatorial Search - 2009.
- Ariafar, S., N. Ismail, S. H. Tang, M. K. A. M. Ariffin und Z. Firoozi (2011). A stochastic facility layout model in cellular manufacturing systems. *International Journal of the Physical Sciences* 6(15), S. 3754–3758.
- Asef-Vaziri, A. und G. Laporte (2005). Loop based facility planning and material handling. *European Journal of Operational Research* 164(1), S. 1 – 11.
- Badiru, A. B. und A. Arif (1996). FLEXPART: facility layout expert system using fuzzy linguistic relationship codes. *IIE Transactions* 28(4), S. 295–308.
- Balakrishnan, J., C. H. Cheng, D. G. Conway und C. M. Lau (2003). A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics* 86(2), S. 107–120.
- Bazaraa, M. S. (1975). Computerized layout design: a branch and bound approach. *AIIE Transactions* 7(4), S. 432–438.
- Bénabès, J., F. Bennis, E. Poirson, Y. Ravaut et al. (2010). Accessibility in layout optimization. In: *Proceedings of the EngOpt2010 Conference*.

- Bhowmik, R. (2008). An Approach to the Facility Layout Design Optimization. *International Journal of Computer Science and Network Security* 8, S. 212–220.
- Bland, J. A. (1999). Space-planning by Ant Colony Optimization. *International Journal of Computer Applications in Technology* 12, S. 320–328.
- Blte, A. und U. W. Thonemann (1996). Optimizing simulated annealing schedules with genetic programming. *European Journal of Operational Research* 92(2), S. 402 – 416.
- Bock, S. und K. Hoberg (2007). Detailed layout planning for irregularly-shaped machines with transportation path design. *European Journal of Operational Research* 177(2), S. 693–718.
- Bonäs, F., J. Westerlund und T. Westerlund (2007). An interactive optimization tool for allocation problems. *Chemical Engineering Transactions* 11, S. 107–112.
- Bozer, Y. A., R. D. Meller und S. J. Erlebacher (1994). An Improvement-Type Layout Algorithm for Single and Multiple-Floor Facilities. *Management Science* 40(7), S. 918–932.
- Braglia, M. (1996). Optimisation of a Simulated Annealing-based Heuristic for Single Row Machine Layout Problem by Genetic Algorithm. *International Transactions in Operational Research* 3(1), S. 37–49.
- Burkard, R. E., E. Çela und P. M. Pardalos (1998). The Quadratic Assignment Problem. In: D. Du und P. M. Pardalos (Hrsg.), *Handbook of combinatorial optimization. 3*, Volume 4. Springer.
- Cagan, J., D. Degentesh und S. Yin (1998). A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout. *Computer-Aided Design* 30(10), S. 781 – 790.
- Cagan, J., K. Shimada und S. Yin (2002). A survey of computational approaches to three-dimensional layout problems. *Computer-Aided Design* 34, S. 597–611.
- Chau, K. und M. Anson (2002). A Knowledge-Based System for Construction Site Level Facilities Layout. In: T. Hendtlass und M. Ali (Hrsg.), *Developments in Applied Artificial Intelligence*, Volume 2358 of *Lecture Notes in Computer Science*, S. 393–402. Springer Berlin Heidelberg.
- Chen, C., S. Lee und Q. Shen (1995). An analytical model for the container loading problem. *European Journal of Operational Research* 80(1), S. 68 – 76.
- Commander, C. W. (2003). *A survey of the quadratic assignment problem, with applications*. Dissertation, university of Florida.
- Drezner, Z. (1987). A Heuristic Procedure for the Layout of a Large Number of Facilities. *Management Science* 33(7), S. pp. 907–915.

- Drira, A., H. Pierreval und S. Hajri-Gabouj (2007). Facility layout problems: A survey. *Annual Reviews in Control* 31, S. 255–267.
- Finke, G., R. E. Burkard und F. Rendl (1987). Quadratic Assignment Problems. *Annals of Discrete Mathematics* 31, S. 61–82.
- Foulds, L. R. (1983). Techniques for Facilities Layout: Deciding which Pairs of Activities Should be Adjacent. *Management Science* 29(12), S. 1414–1426.
- Foulds, L. R. und D. F. Robinson (1976). A Strategy for Solving the Plant Layout Problem. *Operational Research Quarterly (1970-1977)* 27(4), S. pp. 845–855.
- Furber, S. (2002). *ARM-Rechnerarchitekturen für System-on-Chip-Design* (1 Aufl.). Bonn: mitp.
- Furuholmen, M., K. Glette, M. Hovin und J. Torresen (2010). A Coevolutionary, Hyper Heuristic Approach to the Optimization of Three-dimensional Process Plant Layouts; a Comparative Study. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*, S. 1–8.
- Ganesharajah, T., N. Hall und C. Sriskandarajah (1998). Design and operational issues in AGV-served manufacturing systems. *Annals of Operations Research* 76, S. 109–154.
- Gilmore, P. C. (1962). Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the Society for Industrial Applied Mathematics* 10(2), S. 305–313.
- Golmohammadi, V. R., A. A. Jaafari, M. B. Baygi, E. Esfanji und M. Rahimi (2010). An Approach for Facility Layout Problem. *Proceedings of the International MultiConference of Engineers and Computer Scientists* 3, S. 2154–2156.
- Hahn, P. und T. Grant (1998). Lower bounds for the quadratic assignment problem based upon a dual formulation. *Operations Research* 46(6), S. 912–922.
- Hahn, P. M., B.-J. Kim, T. Stützle, S. Kanthak, W. L. Hightower, H. Samra, Z. Ding und M. Guignard (2008). The quadratic three-dimensional assignment problem: Exact and approximate solution methods. *European Journal of Operational Research* 184(2), S. 416–428.
- Hahn, P. M. und J. Krarup (2001). A hospital facility layout problem finally solved. *Journal of Intelligent Manufacturing* 12(5-6), S. 487–496.
- Hahn, P. M., Y.-R. Zhu, M. Guignard, W. L. Hightower und M. J. Saltzman (2012). A level-3 reformulation-linearization technique-based bound for the quadratic assignment problem. *INFORMS Journal on Computing* 24(2), S. 202–209.
- Hassan, M. und G. L. Hogg (1987). A review of graph theory application

- to the facilities layout problem. *Omega* 15(4), S. 291–300.
- Heragu, S. S. (2008). *Facilities design* (3 Aufl.). Boca Raton, Fla. [u.a.]: CRC Press. Includes bibliographical references and index. - Previous ed.: 2006. - Formerly CIP.
- Heragu, S. S. und A. Kusiak (1991). Efficient models for the facility layout problem. *European Journal of Operational Research* 53(1), S. 1–13.
- Hofstedt, P. und A. Wolf (Hrsg.) (2007). *Einführung in die Constraint-Programmierung : Grundlagen, Methoden, Sprachen, Anwendungen*. eXamen.pressSpringerLink : Bücher. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Huang, C., C. Wong und C. Tam (2011). Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming. *Automation in Construction* 20(5), S. 571 – 580.
- IBM (2012). *CP Optimizer User's Manual (Version: 12.4)*. Armonk, New York, U.S.: IBM Corporation.
- Jang, S.-H. (2008). A study on three dimensional layout design by the simulated annealing method. *Journal of Mechanical Science and Technology* 22(11), S. 2016–2023.
- Jiang, L., L. Huang und Q. Xu (2009). Test architecture design and optimization for three-dimensional SoCs. In: *Design, Automation Test in Europe Conference Exhibition, 2009. DATE '09.*, S. 220–225.
- Johnson, R. V. (1982). Spacecraft for Multi-Floor Layout Planning. *Management Science* 28(4), S. 407–417.
- Koopmans, T. C. und M. Beckmann (1957). Assignment Problems and the Location of Economic Activities. *Econometrica* 25(1), S. pp. 53–76.
- Köpcke, H. und A. Schröder (2004). Constraint Programming versus Logik: Vergleich zweier Ansätze zur Aufstellungsplanung von Chemieanlagen. Forschungsbericht, Universität Dortmund, Secretary of the SFB 531, University of Dortmund, Dept. of Computer Science/XI, 44221 Dortmund, Germany.
- Kouvelis, P., W.-C. Chiang und G. Yu (1995). Optimal algorithms for row layout problems in automated manufacturing systems. *IIE Transactions* 27(1), S. 99–104.
- Krasnow, P. C. und B. I. Parker (Hrsg.) (1998). *Correctional facility design and detailing* (6. print. Aufl.). New York [u.a.]: McGraw-Hill.
- Kulturel-Konak, S. und A. Konak (2011). Unequal area flexible bay facility layout using ant colony optimisation. *International Journal of Production Research* 49(7), S. 1877–1902.
- Kumara, S. R. T., R. L. Kashyap und C. L. Moodie (1988). Application of expert systems and pattern recognition methodologies to facilities

- layout planning. *International Journal of Production Research* 26(5), S. 905–930.
- Lawler, E. L. (1963). The quadratic assignment problem. *Management science* 9(4), S. 586–599.
- Lee, K.-Y., M.-I. Roh und J. Hyuk-Su (2005). An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *Computers & Operations Research* 32, S. 879–899.
- Lee, T., G. Moslemipour, T. Ting und D. Rilling (2012). A Novel Hybrid ACO/SA Approach to Solve Stochastic Dynamic Facility Layout Problem (SDFLP). In: D.-S. Huang, P. Gupta, X. Zhang, und P. Premaratne (Hrsg.), *Emerging Intelligent Computing Technology and Applications*, Volume 304 of *Communications in Computer and Information Science*, S. 100–108. Springer Berlin Heidelberg.
- Lengauer, T. und M. Schäfer (2001). Automated Layout Generation and Wiring Area Estimation for 3D Electronic Modules. *ASME Journal of Mechanical Design* 123(3), S. 330–336.
- Li, Y., P. Pardalos, K. Ramakrishnan und M. Resende (1994). Lower bounds for the quadratic assignment problem. *Annals of Operations Research* 50(1), S. 387–410.
- Liu, F., M. Dong, F. Hou und F. Chen (2006). Facility Layout Optimization with Stochastic Logistic Flows. In: *Service Operations and Logistics and Informatics, 2006. SOLI '06. IEEE International Conference on*, S. 534–539.
- Loiola, E. M., N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn und T. Querido (2007). A survey for the quadratic assignment problem. *European Journal of Operational Research* 176, S. 657–690.
- Malapert, A., H. Cambazard, C. Guéret, N. Jussien, A. Langevin und L.-M. Rousseau (2012). An Optimal Constraint Programming Approach to the Open-Shop Problem. *INFORMS Journal on Computing* 24(2), S. 228–244.
- Mayer, S. H. (2009). *Development of a completely decentralized control system for modular continuous conveyors*. Dissertation.
- Meller, R. D. und Y. A. Bozer (1996). A new simulated annealing algorithm for the facility layout problem. *International Journal of Production Research* 34(6), S. 1675–1692.
- Meng, G., S. Heragu und H. Zijm (2004). Reconfigurable layout problem. *International Journal of Production Research* 42(22), S. 4709–4729.
- Nugent, C. E., T. E. Vollmann und J. Ruml (1968). An experimental comparison of techniques for the assignment of facilities to locations. *Operations research* 16(1), S. 150–173.

- Pardalos, P. M., F. Rendl und H. Wolkowicz (1994). The Quadratic Assignment Problem: A Survey and Recent Developments. In: *Proceedings of the DIMACS Workshop on Quadratic Assignment Problems, volume 16 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, S. 1–42. American Mathematical Society.
- Pesch, E. (1999). Efficient facility layout planning in a maximally planar graph model. *International Journal of Production Research* 37(2), S. 263–283.
- Povh, J. und F. Rendl (2009). Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization* 6(3), S. 231–241.
- Refalo, P. (2004). Impact-based search strategies for constraint programming. In: *Principles and Practice of Constraint Programming–CP 2004*, S. 557–571. Springer.
- Rosenblatt, M. J. (1986). The Dynamics of Plant Layout. *Management Science* 32(1), S. 76–86.
- Sagan, H. (1994). *Space filling curves*. Universitext. New York: Springer. Literaturverz. S. 177 - 185.
- Scholz, D. (2010). *Innerbetriebliche Standortplanung: Das Konzept der Slicing Trees bei der Optimierung von Layoutstrukturen*. SpringerLink : Bücher. Wiesbaden: Gabler.
- Scholz, D., A. Petrick und W. Domschke (2009). STaTS: A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research* 197(1), S. 166 – 178.
- Seppänen, J. und J. M. Moore (1970). Facilities Planning with Graph Theory. *Management Science* 17(4), S. B242–B253.
- Shayan, E. und A. Chittilappilly (2004). Genetic algorithm for facilities layout problems based on slicing tree structure. *International Journal of Production Research* 42(19), S. 4055–4067.
- Simmons, D. M. (1969). One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research* 17(5), S. 812–826.
- Skorin-Kapov, J. (1990). Tabu search applied to the quadratic assignment problem. *ORSA Journal on computing* 2(1), S. 33–45.
- Solimanpur, M. und A. Jafari (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers & Industrial Engineering* 55(3), S. 606–619.
- Stockmeyer, L. (1983). Optimal orientations of cells in slicing floorplan designs. *Information and Control* 57(2-3), S. 91 – 101.
- Tam, C., T. Tong, A. Leung und G. Chiu (2002). Site Layout Planning using Nonstructural Fuzzy Decision Support System. *Journal of Construction*

- 
- Engineering and Management* 128(3), S. 220–231.
- Tam, K. Y. und S. G. Li (1991). A hierarchical approach to the facility layout problem. *International Journal of Production Research* 29(1), S. 165–184.
- Tate, D. M. und A. E. Smith (1995). Unequal-area facility layout by genetic search. *IIE Transactions* 27(4), S. 465–472.
- Tavakkoli-Moghaddam, R., N. Javadian, B. Javadi und N. Safaei (2007). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. *Applied Mathematics and Computation* 184(2), S. 721 – 728.
- Teng, H.-F. T., S.-L. Sun, D.-Q. Liu und Y.-Z. Li (2001). Layout optimization for the objects located within a rotating vessel - a three-dimensional packing problem with behavioral constraints. *Computers & Operations Research* 28(6), S. 521 – 535.
- Wilhelm, M. R. und T. L. Ward (1987). Solving quadratic assignment problems by simulated annealing. *IIE transactions* 19(1), S. 107–119.
- Wong, K. Y. und Komarudin (2010). Solving facility layout problems using Flexible Bay Structure representation and Ant System algorithm. *Expert Systems with Applications* 37(7), S. 5523 – 5527.
- Wu, X., Y. Chen, K. Chakrabarty und Y. Xie (2010). Test-access mechanism optimization for core-based three-dimensional SOCs. *Microelectronics Journal* 41(10), S. 601 – 615.
- Xia, Y. (2008). Gilmore-Lawler bound of quadratic assignment problem. *Frontiers of Mathematics in China* 3(1), S. 109–118.
- Xie, W. und N. V. Sahinidis (2008). A branch-and-bound algorithm for the continuous facility layout problem. *Computers and Chemical Engineering* 32, S. 1016–1028.
- Zhang, B., H.-F. Teng und Y.-J. Shi (2008). Layout optimization of satellite module using soft computing techniques. *Applied Soft Computing* 8(1), S. 507 – 521.
- Zhu, J. H., P. Beckers und W. H. Zhang (2010). On the multi-component layout design with inertial force. *Journal of Computational and Applied Mathematics* 234(7), S. 2222–2230.



# Symbolverzeichnis

## Abkürzungen

CP	Constraint Optimierungsproblem; Constraint-logische Programmierung
CSP	Constraint Satisfaction Problem
EVB	Eigenwert-Schranke
GLB	Gilmore-Lawler-Schranke
HFTM	High Flux Test Module
IFMIF	International Fusion Materials Irradiation Facility
LFTM	Low Flux Test Module
MFTM	Middle Flux Test Module
OPL	Optimization Programming Language, Modellierungssprache von IBM ILOG
QZP	Quadratisches Zuordnungsproblem
SRFLP	Single Row Facility Layout Problem
IBM	Software zur Abbildung von Constraint Optimierungsproblem, auch als
ILOG CP	CP Optimizer bezeichnet
Soc	System-on-a-chip

## Formelzeichen

$\alpha$	Wahrscheinlichkeit eines Leerflusses beim Generieren der Flussmatrix
$B$	Breite des verfügbaren Rasters
$b_o$	Breite des Objekts $o$
$C$	Menge der Constraints
$D(\pi_i)$	Wertebereich der Variablen $\pi_i$
$d_{j,k}$	Distanz zwischen zwei Position $j$ und $k$
$E_i$	Constraint zur Beschränkung des Wertebereichs der Variablen $\pi_i$
$\vec{e}_{v_a}$	Einheitsvektor des Vektors $\vec{v}_a$
$F$	Flussmatrix des quadratischen Zuordnungsproblems
$\hat{f}_i$	$i$ -te Zeile der Matrix $F$ ohne das Element $f_{ii}$
$G$	Kostenmatrix des quadratischen Zuordnungsproblems
$\hat{g}_i$	$i$ -te Zeile der Matrix $G$ ohne das Element $g_{ii}$
$\Gamma$	Gini-Koeffizient
$H$	Höhe des verfügbaren Rasters

$h_o$	Höhe des Objekts $o$
$I$	Menge aller bereits berechneten Instanzen
$J$	Menge der Merkmale einer Instanz
$K$	Kostenmatrix zur Bestimmung der Gilmore-Lawler-Schranke
$L$	Länge des verfügbaren Rasters
$l_o$	Länge des Objekts $o$
$\lambda$	Aufsteigend sortierte Eigenwerte der Matrix $A$
$\mu$	Aufsteigend sortierte Eigenwerte der Matrix $B$
$N$	Anzahl der Positionen im verfügbaren Raster
$n_o$	Anzahl der Blöcke eines Objekts
$p_j$	Tupel aus $(x_j, y_j, z_j)$ zur Angabe der Koordinaten einer Position $j$
$p'_{i,o}$	Tupel aus $(x'_{i,o}, y'_{i,o}, z'_{i,o})$ zur Angabe der der relativen Position eines Blocks $i$ innerhalb des Objekts $o$
$p'_{min,i,o}$	Tupel der Mindestmaße $(x'_{min,i,o}, y'_{min,i,o}, z'_{min,i,o})$ eines Blocks $i$ innerhalb des Objekts $o$
$\tilde{p}'_{min,i,o}$	Aufsteigend sortiertes Tupel der Mindestmaße $(x'_{min,i,o}, y'_{min,i,o}, z'_{min,i,o})$ eines Blocks $i$ innerhalb des Objekts $o$
$\pi$	Menge aller Variablen $\pi_i$
$\pi_i$	Variable zur Bezeichnung der Position eines Blocks $i$
$q_i$	Relative Häufigkeit einer Beobachtung zur Berechnung des Gini-Koeffizient
$\rho$	Menge aller Variablen $\pi_i$
$\rho_j$	Variable zur Bezeichnung des Blocks einer Position $j$
$s_{\pi_h, \pi_i}$	Anzahl identischer Koordinaten der durch die Variablen $\pi_h$ und $\pi_i$ gekennzeichneten Positionen
$t_{h,i}$	Transportmenge zwischen zwei Blöcken $h$ und $i$
$u_{i,j}$	Ausprägung des Merkmals $j$ für Instanz $i$
$\tilde{u}_{i,j}$	Normalisierter Wert des Merkmals $j$ für Instanz $i$
$\vec{v}_a$	Bewertungsvektor der Variablen bezüglich eines Merkmals $a$
$x_j$	x-Koordinate der Position $j$
$x'_{i,o}$	Relative x-Koordinate des Blocks $i$ innerhalb des Objekts $o$
$x'_{min,i,o}$	Mindestmaße von Block $i$ innerhalb des Objekts $o$ in x-Richtung
$y_j$	y-Koordinate der Position $j$
$y'_{i,o}$	Relative y-Koordinate des Blocks $i$ innerhalb des Objekts $o$
$y'_{min,i,o}$	Mindestmaße von Block $i$ innerhalb des Objekts $o$ in y-Richtung
$z_j$	z-Koordinate der Position $j$
$z'_{i,o}$	Relative z-Koordinate des Blocks $i$ innerhalb des Objekts $o$
$z'_{min,i,o}$	Mindestmaße von Block $i$ innerhalb des Objekts $o$ in z-Richtung

# A. Daten der Testinstanzen

## A.1. Vom quadratischen Zuordnungsproblem abgeleitete Instanzen

8	9	35	56	14	29	18	60	36
7	50	58	45	52	20	48	46	61
6	37	23	1	22	24	5	40	27
5	33	7	44	16	55	39	30	12
4	42	25	21	64	4	51	43	3
3	59	26	17	63	62	6	28	34
2	2	13	8	49	19	47	11	41
1	31	57	53	10	38	54	32	15
	1	2	3	4	5	6	7	8

sko64

8	9	35	56	14	29	18	60	36
7	50	7	58	45	52	8	20	48
6	37	23	1	22	24	5	40	9
5	33	7	44	16	55	39	30	12
4	42	25	21	64	4	51	43	6
3	59	26	17	63	62	6	28	34
2	2	13	8	49	19	2	47	11
1	31	57	53	10	38	54	32	15
	1	2	3	4	5	6	7	8

sko64A

8	9	15	35	16	14	7	29	18	19	60	36
7	50	13	45	52	20	48	18	46	61	20	
6	37	23	1	22	24	5	14	27	11		
5	33	7	44	16	55	39	30	12	10		
4	42	8	25	21	9	4	51	43	3	10	
3	59	26	17	63	5	6	28	34	7		
2	2	13	8	49	19	47	3	41	4		
1	31	57	53	10	38	54	32	15	5		
	1	2	3	4	5	6	7	8			

sko64B

8	9	35	56	14	29	18	60	15	36	
7	50	11	58	45	52	20	48	46	14	61
6	37	23	1	22	24	5	40	27	13	
5	33	7	44	16	55	39	30	12	10	
4	42	25	21	64	4	51	43	9	3	
3	59	26	17	63	62	6	28	8	34	
2	2	13	8	49	19	47	11	41	5	
1	31	57	53	10	38	54	32	15	5	
	1	2	3	4	5	6	7	8		

sko64C

A. Daten der Testinstanzen

9	47	40	54	10	53	28	71	72	8
8	80	77	79	24	75	59	43	29	22
7	16	64	2	35	73	36	5	1	55
6	76	63	12	58	13	46	52	62	45
5	60	37	51	18	32	7	42	68	27
4	34	20	26	44	39	11	70	57	78
3	23	69	14	19	15	21	31	17	74
2	67	65	4	49	81	6	56	48	9
1	30	25	33	38	61	41	3	50	66
	1	2	3	4	5	6	7	8	9

sko81

9	47	40	54	10	53	28	71	72	8
8	80	77	79	24	75	59	43	29	22
7	16	64	2	35	73	36	5	1	55
6	76	63	12	58	13	46	52	62	45
5	60	37	51	18	32	7	42	68	27
4	34	20	26	44	39	11	70	57	78
3	23	69	14	19	15	21	31	17	74
2	67	65	4	49	81	6	56	48	9
1	30	25	33	38	61	41	3	50	66
	1	2	3	4	5	6	7	8	9

sko81A

9	47	40	54	10	53	28	71	72	8
8	80	77	79	24	75	59	43	29	22
7	16	64	2	35	73	36	5	1	55
6	76	63	12	58	13	46	52	62	45
5	60	37	51	18	32	7	42	68	27
4	34	20	26	44	39	11	70	57	78
3	23	69	14	19	15	21	31	17	74
2	67	65	4	49	81	6	56	48	9
1	30	25	33	38	61	41	3	50	66
	1	2	3	4	5	6	7	8	9

sko81B

9	47	40	54	10	53	28	71	72	8
8	80	77	79	24	75	59	43	29	22
7	16	64	2	35	73	36	5	1	55
6	76	63	12	58	13	46	52	62	45
5	60	37	51	18	32	7	42	68	27
4	34	20	26	44	39	11	70	57	78
3	23	69	14	19	15	21	31	17	74
2	67	65	4	49	81	6	56	48	9
1	30	25	33	38	61	41	3	50	66
	1	2	3	4	5	6	7	8	9

sko81C

10	3	35	21	29	23	48	44	7	76	17
9	45	50	12	72	61	38	92	36	53	41
8	58	70	66	100	26	47	99	46	64	90
7	10	24	40	67	85	30	73	51	69	20
6	42	59	68	8	93	56	81	82	84	87
5	60	9	86	98	96	89	19	49	2	28
4	88	62	16	52	4	22	32	34	57	15
3	27	80	54	94	75	31	43	14	77	6
2	39	5	13	55	74	95	25	63	79	91
1	65	83	11	33	71	97	1	78	37	18
	1	2	3	4	5	6	7	8	9	10

sko100

10	3	35	21	29	23	48	44	7	76	17
9	45	50	12	72	61	38	92	36	53	41
8	58	70	66	100	26	47	99	46	64	90
7	10	24	40	67	85	30	73	51	69	20
6	42	59	68	8	93	56	81	82	84	87
5	60	9	86	98	96	89	19	49	2	28
4	88	62	16	52	4	22	32	34	57	15
3	27	80	54	94	75	31	43	14	77	6
2	39	5	13	55	74	95	25	63	79	91
1	65	83	11	33	71	97	1	78	37	18
	1	2	3	4	5	6	7	8	9	10

sko100A

A.1. Vom quadratischen Zuordnungsproblem abgeleitete Instanzen

10	3	35	21	14	29	23	48	44	7	76	20	
9	45	12	10	12	72	15	38	16	92	36	53	19
8	58	70	66	100	26	47	99	46	17	90		
7	10	24	10	67	85	30	11	51	69	20		
6	42	59	68	8	93	56	81	82	84	9	87	
5	60	9	86	98	96	89	19	49	2	8	28	
4	88	62	16	52	4	22	32	34	57	15		
3	27	80	54	94	75	31	43	14	77	4	6	
2	39	5	13	55	74	95	25	63	79	91		
1	65	83	11	33	71	97	1	78	37	18		
	1	2	3	4	5	6	7	8	9	10		

sko100B

10	3	35	21	29	23	48	44	7	76	17	
9	45	50	12	72	61	38	92	36	53	41	20
8	58	70	66	100	26	47	99	46	17	90	19
7	10	24	10	67	85	30	11	51	69	18	20
6	42	59	68	8	93	56	81	82	84	9	87
5	60	9	86	98	96	89	19	49	2	8	28
4	88	62	16	52	4	22	32	34	57	15	
3	27	80	54	94	75	31	43	14	77	4	6
2	39	5	13	55	74	95	25	63	79	91	
1	65	83	11	33	71	97	1	78	37	18	
	1	2	3	4	5	6	7	8	9	10	

sko100C

5	46	28	42	50	17	4	34	36	18	7	
4	22	10	21	6	11	37	39	8	13	26	
3	35	29	30	16	41	45	49	43	2	20	
2	14	5	32	44	3	24	38	23	12	25	
1	1	48	40	31	15	33	47	19	9	27	
	1	2	3	4	5	6	7	8	9	10	

wil50

5	46	28	42	50	17	4	34	36	18	7	
4	22	10	21	6	11	37	39	8	13	26	10
3	35	29	30	16	41	45	49	43	2	20	
2	14	5	32	44	3	24	38	23	12	25	
1	1	48	40	31	15	33	47	19	9	27	
	1	2	3	4	5	6	7	8	9	10	

wil50A

5	46	28	42	9	50	17	4	12	24	13	18	7	
4	22	10	8	6	11	37	39	8	13	15	26		
3	35	29	30	16	41	45	49	43	2	14	20		
2	14	5	32	3	44	3	4	24	38	5	23	12	25
1	1	48	40	31	15	33	47	19	9	6	27		
	1	2	3	4	5	6	7	8	9	10			

wil50B

5	46	6	28	42	50	17	4	34	36	18	15	7	
4	22	10	8	6	11	37	39	8	13	14	26		
3	35	29	30	16	41	45	49	43	2	13	20		
2	14	5	32	3	44	3	4	24	38	10	23	12	25
1	1	48	40	31	15	33	47	19	9	4	27		
	1	2	3	4	5	6	7	8	9	10			

wil50C

A. Daten der Testinstanzen

---

10	57	62	35	91	66	60	82	21	85	94
9	84	24	86	11	8	40	59	17	70	92
8	74	22	58	93	42	7	38	12	89	20
7	63	73	3	96	5	75	19	67	97	51
6	81	31	27	54	4	47	44	2	34	36
5	14	37	13	45	80	41	61	16	18	6
4	46	71	39	76	26	69	52	99	53	25
3	1	29	98	79	90	83	23	72	49	33
2	9	10	43	56	78	77	65	68	55	48
1	15	28	100	64	95	88	32	87	30	50
	1	2	3	4	5	6	7	8	9	10

wil100

10	57	62	35	91	66	60	82	21	85	94
9	84	24	86	11	8	40	59	17	70	92
8	74	22	58	93	42	7	38	12	89	20
7	63	73	3	96	5	75	19	67	97	51
6	81	31	27	54	4	47	44	2	34	36
5	14	37	13	45	80	41	61	16	18	6
4	46	71	39	76	26	69	52	99	53	25
3	1	29	98	79	90	83	23	72	49	33
2	9	10	43	56	78	77	65	68	55	48
1	15	28	100	64	95	88	32	87	30	50
	1	2	3	4	5	6	7	8	9	10

wil100A

10	57	62	35	91	66	60	82	21	85	94
9	84	24	86	11	8	40	59	17	70	92
8	74	22	58	93	42	7	38	12	89	20
7	63	73	3	96	5	75	19	67	97	51
6	81	31	27	54	4	47	44	2	34	36
5	14	37	13	45	80	41	61	16	18	6
4	46	71	39	76	26	69	52	99	53	25
3	1	29	98	79	90	83	23	72	49	33
2	9	10	43	56	78	77	65	68	55	48
1	15	28	100	64	95	88	32	87	30	50
	1	2	3	4	5	6	7	8	9	10

wil100B

10	57	62	35	91	66	60	82	21	85	94
9	84	24	86	11	8	40	59	17	70	92
8	74	22	58	93	42	7	38	12	89	20
7	63	73	3	96	5	75	19	67	97	51
6	81	31	27	54	4	47	44	2	34	36
5	14	37	13	45	80	41	61	16	18	6
4	46	71	39	76	26	69	52	99	53	25
3	1	29	98	79	90	83	23	72	49	33
2	9	10	43	56	78	77	65	68	55	48
1	15	28	100	64	95	88	32	87	30	50
	1	2	3	4	5	6	7	8	9	10

wil100C