

Karlsruher Schriften
zur Anthropomatik

Band 21



Todor Dimitrov

**Permanente Optimierung dynamischer
Probleme der Fertigungssteuerung unter
Einbeziehung von Benutzerinteraktionen**

Todor Dimitrov

**Permanente Optimierung dynamischer
Probleme der Fertigungssteuerung unter
Einbeziehung von Benutzerinteraktionen**

Karlsruher Schriften zur Anthropomatik

Band 21

Herausgeber: Prof. Dr.-Ing. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe
erschienenen Bände finden Sie am Ende des Buchs.

Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen

von
Todor Dimitrov

Dissertation, Karlsruher Institut für Technologie (KIT)
Fakultät für Informatik, 2015

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark of Karlsruhe
Institute of Technology. Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under the Creative Commons Attribution-Share Alike 3.0 DE License
(CC BY-SA 3.0 DE): <http://creativecommons.org/licenses/by-sa/3.0/de/>*



*The cover page is licensed under the Creative Commons
Attribution-No Derivatives 3.0 DE License (CC BY-ND 3.0 DE):
<http://creativecommons.org/licenses/by-nd/3.0/de/>*

Print on Demand 2015

ISSN 1863-6489

ISBN 978-3-7315-0426-9

DOI 10.5445/KSP/1000048541

Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Todor Dimitrov

aus Sofia

Tag der mündlichen Prüfung: 02.07.2015

Erster Gutachter: Prof. Dr.-Ing. habil. Jürgen Beyerer

Zweiter Gutachter: Prof. Dr. rer. pol. habil. Frank Schultmann

Kurzfassung

Feinplanung und Steuerung von Fertigungskapazitäten ist ein mannigfaltiges NP-hartes Problem, das die Forschungswelt seit 70 Jahren untersucht. Obwohl viele fortschrittliche Feinplanungsverfahren aus dem Bereich Operations Research, der Künstlichen Intelligenz oder anderen Domänen entwickelt und angepasst wurden, finden sehr wenige und vor allem rudimentäre Feinplanungsverfahren Einsatz in der Industrie. In nahezu allen heutigen Fertigungssteuerungssystemen sind sie nur ein Hilfsmittel zur Erstellung eines Basisfertigungsplans, den der Planer manuell nachbearbeitet. Mit steigendem Automatisierungsgrad und dem Einsatz neuer Konzepte wie Industrie 4.0, Connected Manufacturing und Industrial Internet wird eine selbstständige Beteiligung des Feinplanungsverfahrens am täglichen Fertigungsablauf zunehmend wichtiger. Das parallele Agieren (mixed initiative) eines permanent laufenden Feinplanungsverfahrens und eines oder mehrerer Planer erfordert jedoch eine intelligente Berücksichtigung der Benutzerinteraktionen seitens der Feinplanungsverfahren und Nachvollziehbarkeit der Algorithmen seitens der Benutzer.

In dieser Arbeit wurde ein Konzept für die Steuerung von permanent laufenden Feinplanungsverfahren entworfen, das diesen ein intelligentes Verhalten in der Fertigungssteuerung gegenüber einem Planer verschafft. Um ein Feinplanungsverfahren nachvollziehbarer für einen Planer und reaktionsfähiger im täglichen Fertigungsablauf auszuführen, wurde es auf Fragmente des Fertigungsplans mit Optimierungspotential und -bedarf gerichtet. Um solche Fragmente im Fertigungsplan zu erkennen, wurden klassische Bewertungskennzahlen in der Feinplanung verbessert, um die Relevanz zur aktuellen Fertigungssituation mit zu betrachten.

Für die vollwertige Kollaboration mit den Planern wird das Feinplanungsverfahren auf Einhaltung der Benutzerinteraktionen und Sicherung der Planstabilität ausgelegt, um die Planer nicht zu verunsichern bzw. den Überblick über den auszuführenden Fertigungsplan nicht zu gefährden. Umfangreiche Veränderungen im Fertigungsplan oder Verletzung von Benutzerinteraktionen werden nur bei entsprechend großen Verbesserungen zugelassen.

Für die Steigerung der Performanz und die Verwendung der verfügbaren Rechenleistung wurde ein Ansatz für die Ausführung und Koordination von parallelen Instanzen von Feinplanungsverfahren auf demselben Fertigungsplan erarbeitet. Diese Verfahrensinstanzen optimieren disjunkte Fragmente des Fertigungsplans unabhängig voneinander und ohne eine übergeordnete Kontrollinstanz. Da sie sich auf optimierungsbedürftige Fragmente konzentrieren, erhalten sie ein nachvollziehbares menschenähnliches Verhalten.

Für die Validierung des Konzepts wurden die entwickelten Ansätze in einer performanten Feinplanungsplattform zusammengefasst. Ein genetischer Algorithmus wurde so modifiziert, dass er Optimierungshinweise der Bewertungskomponente verarbeitet. Als Ergebnis wurde eine hochwertige Zusammenarbeit zwischen mehreren permanent laufenden Verfahrensinstanzen eines genetischen Algorithmus und menschlichen Benutzern erreicht. Gleichzeitig stiegen die Performanz und die Lösungsqualität des genetischen Algorithmus deutlich. Der Nachweis erfolgte in einer dynamischen Simulationsumgebung mit Daten von zwei produzierenden Betrieben.

Vorwort

Die vorliegende Arbeit entstand während meiner achtjährigen Tätigkeit als wissenschaftlicher Mitarbeiter im Themenfeld Fertigungssteuerung am Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB).

Mein Dank für die hilfreiche Unterstützung bei der Wegweisung und Erstellung meiner Doktorarbeit geht vor allem an meinen Doktorvater Prof. Dr.-Ing. Jürgen Beyerer. Herrn Prof. Dr. rer. pol. Frank Schultmann danke ich für die Übernahme des Korreferats und die Betreuung der Arbeit bis zum Schluss.

Ich danke allen Kollegen der Abteilung „Informationsmanagement und Leittechnik“ sowie Herrn Dr.-Ing. Olaf Sauer für die wertvolle Zusammenarbeit in zahlreichen Projekten.

Mein besonderer Dank geht an Herrn Dr.-Ing. Michael Baumann, der mir als kritischer Sparringspartner für die zahlreichen Diskussionen stets zur Verfügung stand und mit seiner jahrelangen Erfahrung in der Fertigungssteuerung half.

Frau Yvonne Omelianenko und Herrn Dr. Andreas Braun danke ich für die sorgfältige Korrektur meiner Dissertation herzlich.

Nicht zuletzt danke ich meiner Familie und meinen engen Freunden für deren ununterbrochene geistige Unterstützung.

Karlsruhe, im Juli 2015

Todor Dimitrov

Inhaltsverzeichnis

Kurzfassung.....	i
Vorwort.....	iii
Abbildungsverzeichnis.....	ix
Tabellenverzeichnis	xv
Abkürzungsverzeichnis	xvii
Nomenklatur.....	xxi
1 Einleitung.....	1
1.1 Ausgangsituation.....	1
1.2 Motivation	3
1.3 Zielsetzung und eigene Beiträge.....	4
1.4 Aufbau der Arbeit.....	6
2 Stand der Technik.....	9
2.1 Der Begriff „Fertigungssteuerung“	9
2.1.1 Evolution des Begriffs und Einordnung in der betrieblichen Produktion.....	10
2.1.2 Bedeutung für die Unternehmen.....	18
2.2 Das Shop Scheduling Problem.....	20
2.2.1 Mathematische Definition	20
2.2.2 Arten von Shop Scheduling Problemen.....	23
2.2.3 Optimierungsziele.....	27
2.2.4 Optimalität	33
2.3 Verfahren für Shop Scheduling Probleme.....	35
2.3.1 Überblick der Verfahren	36
2.3.2 Optimale Konstruktionsverfahren (Exakte Verfahren)	38
2.3.3 Heuristische Konstruktionsverfahren	39
2.3.4 Heuristische Verbesserungsverfahren	43

2.3.5	Metaheuristische Verbesserungsverfahren.....	44
2.3.6	Hybride Verfahren	50
2.3.7	Vergleichbarkeit von Verfahren.....	52
2.4	Das dynamische Shop Scheduling Problem	53
2.4.1	Rescheduling Strategien.....	56
2.4.2	Kontinuierliche Anpassung.....	57
2.4.3	Anwendung von populationsbasierten Verbesserungsverfahren.....	59
2.4.4	Quantifizierung der Planstabilität.....	60
2.5	Kollaboration zwischen Verfahren und Planern	62
2.5.1	Der Mensch im Planungsprozess.....	63
2.5.2	Entscheidungsunterstützung in der Planung.....	65
2.5.3	Wissensbasierte Planung.....	67
2.5.4	Mixed-Initiative Mensch-Maschine Feinplanung	68
2.6	Planung in der Unternehmens-IT.....	70
2.6.1	Fertigungsfineplanung in Softwaresystemen	71
2.6.2	Eingesetzte Feinplanungsverfahren in der Praxis.....	72
2.6.3	Hindernisse für den Einsatz von Feinplanungssystemen	74
3	Analyse und Anforderungsermittlung	77
3.1	Feinplanungsverfahren in der Fertigungssteuerung heute.....	77
3.2	Die gewünschte Rolle der Feinplanungsverfahren in der Fertigungssteuerung	83
3.3	Anforderungen an ein fortschrittliches Feinplanungssystem	88
3.4	Technische Anforderungen.....	90
4	Ableitung eines Konzepts für eine Feinplanungsplattform...93	
4.1	Gesamtbild der Feinplanungsplattform	93
4.2	Aspekte der Bewertungskomponenten.....	95
4.2.1	Integration des aktuellen Zeitpunkts	96
4.2.2	Quantifizierung der Planstabilität.....	101

4.2.3	Akkumulierte Planstabilität (Shop Floor Nervousness).....	103
4.2.4	Einhaltung von Benutzerinteraktionen.....	105
4.2.5	Bewertung von Fragmenten des Fertigungsplans....	110
4.2.6	Bewertungskriterien für den Vergleich von Fragmenten	113
4.3	Vergleich von Fertigungsplänen.....	116
4.4	Unterstützung der Feinplanungsverfahren bei der Lösungssuche	118
4.4.1	Hinweise auf Fragmente mit Optimierungspotential.....	119
4.4.2	Dynamische Dekomposition des Fertigungsplans ...	122
4.4.3	Parallele Ausführung von Verfahrensinstanzen.....	124
4.5	Koordinierung gleichzeitiger Aktionen von Benutzern und Algorithmen.....	126
4.5.1	Verarbeitung von externen Ereignissen	127
4.5.2	Koordinierung mehrerer Verfahrensinstanzen.....	129
5	Die Feinplanungsplattform im Hinblick auf Performanz und Skalierbarkeit.....	133
5.1	Die Feinplanungsplattform in einer dynamischen Umgebung.....	134
5.2	Inkrementell berechnete Bewertung des Fertigungsplans	136
5.2.1	Inkrementell berechnete Kennzahlen.....	137
5.2.2	Performante Fragmentierungen.....	139
5.2.3	Der aktuelle Zeitpunkt in der inkrementellen Bewertung.....	141
5.2.4	Ableitung der Relevanzkurven.....	142
5.3	Vergleichsfunktion für Fertigungspläne und Übernahme von neuen Lösungen	146
5.4	Optimierungshinweise für die Unterstützung der Feinplanungsverfahren	150
5.4.1	Extraktion von Optimierungshinweisen.....	150

5.4.2	Markierung von Fragmenten	152
5.5	Anbindung eines von Optimierungshinweisen unterstützten Feinplanungsverfahrens.....	155
5.5.1	Auswahl eines geeigneten Feinplanungsverfahrens	155
5.5.2	Anbindung des ausgewählten genetischen Algorithmus an die Feinplanungsplattform	158
5.5.3	Operatoren für „genetisches Engineering“	163
5.5.4	Anpassung des Genetischen Algorithmus für eine kontinuierliche Optimierung.....	166
6	Anwendung und Ergebnisse.....	169
6.1	Simulationsumgebung.....	169
6.2	Implementierung der Feinplanungsplattform	175
6.3	Beschreibung der untersuchten Anwendungsfälle ..	179
6.4	Vorbereitung der Simulationsläufe	182
6.4.1	Simulationsumgebung und Kalibrierung.....	182
6.4.2	Evaluierungsplan.....	186
6.4.3	Konstruktion der dynamischen Problemstellungen	189
6.5	Evaluierung des Anwendungsfalls Batteriehersteller	191
6.5.1	Ansätze zur Performanzsteigerung.....	192
6.5.2	Kollaborationsansätze.....	197
6.6	Evaluierung des Anwendungsfalls Wafer-Fertiger ..	204
6.6.1	Ansätze zur Performanzsteigerung.....	204
6.6.2	Kollaborationsansätze.....	214
6.7	Abschließende Bewertung	221
7	Zusammenfassung und Ausblick.....	225
7.1	Zusammenfassung der Beiträge und Ergebnisse	225
7.2	Ausblick	227
	Literaturverzeichnis	229
	Eigene Veröffentlichungen	261

Abbildungsverzeichnis

Abbildung 1.1: Aufbau der Arbeit	7
Abbildung 2.1: Terminregelung innerhalb der Fertigungssteuerung [Bran67]	10
Abbildung 2.2: Aachener PPS Modell	11
Abbildung 2.3: Das Y-Modell [Sche87]	12
Abbildung 2.4: Das Ebenenmodell von PRM zusammengefasst in [Ansi13a]	13
Abbildung 2.5: Die Automatisierungspyramide	14
Abbildung 2.6: Feinplanung und Feinsteuerung [Vdi07, S.17] ...	15
Abbildung 2.7: Modell des operativen Produktions- managements [Ansi13a]	16
Abbildung 2.8: Aktivitätsdiagramm des „Production Operations Management“ [Ansi13a]	17
Abbildung 2.9: Veränderung der Informationsarchitektur in Produktionsunternehmen [VKBW09]	18
Abbildung 2.10: Evolution relevanter Aspekte der Fertigungssteuerung	19
Abbildung 2.11: Beispiel eines Fertigungsplans	22
Abbildung 2.12: Arten von Fertigungsplänen nach [Pine14, S.25]	23
Abbildung 2.13: Der disjunktive Graph des Fertigungsproblems aus Abbildung 2.11	26
Abbildung 2.14: Beispiel einer Pareto-Front	32
Abbildung 2.15: Komplexitätsklassendiagramm	34
Abbildung 2.16: Verfahren für Shop Scheduling Probleme	37
Abbildung 2.17: Giffler-Thompson Algorithmus	41
Abbildung 2.18: Ablaufschema des genetischen Algorithmus	46
Abbildung 2.19: Beispiel einer Mutation innerhalb des GA und ihre Auswirkung auf den Fertigungsplan	48
Abbildung 2.20: Beispiel einer Kreuzung innerhalb des GA	49

Abbildung 2.21: Beispielhafte Ereignisse in einer laufenden Fertigung.....	54
Abbildung 2.22: Gegenüberstellung der unterschiedlichen Ansätze für dynamische Shop Scheduling Probleme.....	55
Abbildung 2.23: Abstand zwischen Operationen auf μ_2 der Fertigungspläne aus Abbildung 2.19 mit Gewichtung abhängig von t_o	61
Abbildung 2.24: Visuelle Komponenten zur Darstellung des Fertigungsplans: Gantt Diagramm, stetige und diskrete Intervall-Gebirge.....	66
Abbildung 3.1: Grobe Abbildung und Einordnung der Fertigungssteuerung nach dem ISA-95 Standard [Ansi13a].....	78
Abbildung 3.2: Anwendungsfälle in bestehenden Fertigungssteuerungssystemen.....	79
Abbildung 3.3: Fertigungsfeinplanung mittels Planungsverfahren (Anwendungsfall H1).....	80
Abbildung 3.4: Anwendungsfall H2.....	82
Abbildung 3.5: Reaktion auf Störung / Eilauftrag (Anwendungsfall H3).....	83
Abbildung 3.6: Vernetzung in der Unternehmens-IT nach Industrie 4.0.....	84
Abbildung 3.7: Anwendungsfälle in zukünftigen Fertigungssteuerungssystemen.....	85
Abbildung 3.8: Anwendungsfall F1.....	86
Abbildung 3.9: Anwendungsfall F2.....	87
Abbildung 3.10: Anwendungsfall F3.....	87
Abbildung 4.1: Grobe Übersicht der Feinplanungsplattform.....	95
Abbildung 4.2: Der Fertigungsplan aus Sicht des aktuellen Zeitpunkts t_o nach [DiBa12].....	97
Abbildung 4.3: Gewichtung der Verspätung nach klassischer und zeitbehafteter Art [DiBa12].....	99

Abbildung 4.4:	Grafische Darstellung der Nervosität bei der Übernahme von Fertigungsplänen	103
Abbildung 4.5:	Schematische Gewichtung der entstandenen Nervosität für drei stattgefundene Übernahmen.....	105
Abbildung 4.6:	Planstabilität und Einhaltung der Benutzerinteraktion auf o_{12} in zwei Planvorschlägen	106
Abbildung 4.7:	Erneuerung der Auftragsmenge entlang der Zeit.....	108
Abbildung 4.8:	Schematische Gewichtung der Benutzerinteraktionen nach den Interaktionszeitpunkten	109
Abbildung 4.9:	Beispielhafte Nachschaltung von Fragmentierungsmerkmalen.....	111
Abbildung 4.10:	Fragmente samt Operationen aus der Bewertung nach $\sum T_j$	112
Abbildung 4.11:	Übernahme abhängig von θ und ϑ_b	117
Abbildung 4.12:	Allgemeiner Ablauf von Feinplanungsverfahren.....	120
Abbildung 4.13:	Ableitung von Optimierungshinweisen für die Unterstützung von Feinplanungsverfahren.....	121
Abbildung 4.14:	Untersuchung der Fragmenthierarchie	123
Abbildung 4.15:	Markierung von einem Teil des Fertigungsplans	124
Abbildung 4.16:	Beispielhafte Markierung bei mehreren Verfahrensinstanzen	125
Abbildung 4.17:	Warteschlange für Ereignisse in der Feinplanungsplattform.....	128
Abbildung 4.18:	Verarbeitung einer Benutzerinteraktion bei mehreren Verfahrensinstanzen	128
Abbildung 4.19:	Aktivitätsdiagramm einer Verfahrensinanz [DiBS11].....	131
Abbildung 5.1:	Einheitliches Konzept der Feinplanungsplattform und eigene Beiträge	135

Abbildung 5.2: Inkrementell berechnete $\sum T_j$ bei Änderung von Auftrag 2.....	137
Abbildung 5.3: Zwischenwerte für Auftrag 2 entlang der Fragmenthierarchie	141
Abbildung 5.4: Zwei typische Zeithorizonte im hierarchischen Planungsprozess.....	143
Abbildung 5.5: Arten von Relevanzfunktionen	145
Abbildung 5.6: Vergleich und Übernahme von Fertigungsplänen	149
Abbildung 5.7: Extraktion von Optimierungshinweisen	151
Abbildung 5.8: Möglichkeiten zur Markierungsausbreitung	154
Abbildung 5.9: Evaluierung eines Individuums im GA [DiBa11]	159
Abbildung 5.10: Ablauf des modifizierten GA	162
Abbildung 5.11: Gezielte Kreuzung im GA	164
Abbildung 5.12: Gezielte Mutationsarten abhängig von der Art des Optimierungshinweises im GA	165
Abbildung 5.13: Zusammenspiel zwischen Feinplanungsplattform und GA.....	168
Abbildung 6.1: Simulationsumgebung	171
Abbildung 6.2: Simulationsablauf im Fertigungssubstitut	174
Abbildung 6.3: Architektur der realisierten Feinplanungsplattform.....	176
Abbildung 6.4: Zehn Simulationsläufe von GAO für BAT	185
Abbildung 6.5: Zehn Simulationsläufe von GAO für WAF	185
Abbildung 6.6: Evaluierungsplan für die einzelnen Ansätze	187
Abbildung 6.7: Einstellung der Auftragsdaten für die dynamische Simulation	190
Abbildung 6.8: Ergebnis für BAT-1a- $\emptyset DLZ$	193
Abbildung 6.9: Ergebnis für BAT-1a-VI	194
Abbildung 6.10: Untersuchte Relevanzfunktionen in BAT-1b	195
Abbildung 6.11: Zwei repräsentative Simulationsläufe aus BAT-1b in dynamischen Simulationsläufen mit Störungen	197

Abbildung 6.12: Einhaltung der Planstabilität in statischen Simulationen (BAT-2a- θ)	198
Abbildung 6.13: Berücksichtigung der Benutzerinteraktionen in statischen Simulationen (BAT-2a- ϑ_b)	200
Abbildung 6.14: Untersuchte Relevanzfunktionen in BAT-2b....	200
Abbildung 6.15: Zwei repräsentative Simulationsläufe aus BAT-2b- θ in dynamischen Simulationsläufen mit Störungen	202
Abbildung 6.16: Zwei repräsentative Simulationsläufe aus BAT-2b- ϑ_b -5%BI in dynamischen Simulationsläufen mit Störungen	203
Abbildung 6.17: Ergebnis für WAF-1a- $\sum T_j$	205
Abbildung 6.18: Ergebnis für WAF-1a- TST	206
Abbildung 6.19: Untersuchte Fertigungspläne in Teilschritt WAF-1a	207
Abbildung 6.20: Untersuchte Fertigungspläne in Teilschritt WAF-1a- ZF	208
Abbildung 6.21: Ergebnis für WAF-1a-VI	209
Abbildung 6.22: Untersuchte Relevanzfunktionen in WAF-1b...	210
Abbildung 6.23: Zwei repräsentative Simulationsläufe aus WAF-1b- $\sum T_j$ in dynamischen Simulationsläufen mit Störungen	212
Abbildung 6.24: Zwei repräsentative Simulationsläufe aus WAF-1b- TST in dynamischen Simulationsläufen mit Störungen	212
Abbildung 6.25: Zwei repräsentative Simulationsläufe aus WAF-1b- ZF in dynamischen Simulationsläufen mit Störungen	214
Abbildung 6.26: Einhaltung der Planstabilität in statischen Simulationen (WAF-2a- θ)	216
Abbildung 6.27: Berücksichtigung der Benutzerinteraktionen in statischen Simulationen (WAF-2a- ϑ_b)	217
Abbildung 6.28: Untersuchte Relevanzfunktionen in WAF-2b...	218

Abbildung 6.29: Zwei repräsentative Simulationsläufe aus
WAF-2b- θ in dynamischen Simulationsläufen
mit Störungen 219

Abbildung 6.30: Zwei repräsentative Simulationsläufe aus
WAF-2b- ϑ_b -5%BI in dynamischen
Simulationsläufen mit Störungen 221

Tabellenverzeichnis

Tabelle 4.1:	Erweiterung klassischer Zielfunktionen aus 2.2.3 um die Relevanz zum aktuellen Zeitpunkt.....	100
Tabelle 4.2:	Erweiterung klassischer Zielfunktionen aus 2.2.3 und 4.2.1 um das Optimierungspotential (OP)	115
Tabelle 5.1:	Sinnvolle Fragmentierungen pro Kennzahlenart .	140
Tabelle 5.2:	Abdeckung gestellter Kriterien von oft eingesetzten Feinplanungsverfahren	157
Tabelle 5.3:	Unterschiedliche Repräsentationen eines GA	158
Tabelle 6.1:	Daten der simulierten Anwendungsfälle	180
Tabelle 6.2:	Rechenzeit für typische Vorgänge in der Feinplanungsplattform auf den Anwendungsfällen in Millisekunden	183
Tabelle 6.3:	Unterschiede zwischen GA0, GA1, GA2 und GA3 ..	184
Tabelle 6.4:	Ergebnis für BAT-1a- \emptyset DLZ in Tagen.....	192
Tabelle 6.5:	Simulationsergebnisse für BAT-1b in Tagen	196
Tabelle 6.6:	Ergebnis für BAT-2a- ϑ_b in Tagen.....	199
Tabelle 6.7:	Einhaltung der Planstabilität in dynamischen Simulationen (BAT-2b- θ) in Tagen	201
Tabelle 6.8:	Berücksichtigung der Benutzerinteraktionen in dynamischen Simulationen (BAT-2b- ϑ_b) in Tagen	202
Tabelle 6.9:	Ergebnis für WAF-1a in Tagen	205
Tabelle 6.10:	Ergebnis für WAF-1a-ZF in Tagen	207
Tabelle 6.11:	Simulationsergebnisse für WAF-1b- $\sum T_j$ in Tagen	210
Tabelle 6.12:	Simulationsergebnisse für WAF-1b-TST in Tagen	211
Tabelle 6.13:	Simulationsergebnisse für WAF-1b-ZF mit zwei Kombinationen aus Relevanzfunktionen in Tagen	213
Tabelle 6.14:	Ergebnis für WAF-2a- ϑ_b in Tagen	217

Tabelle 6.15: Einhaltung der Planstabilität in dynamischen Simulationen (WAF-2b- θ) in Tagen	218
Tabelle 6.16: Berücksichtigung der Benutzerinteraktionen in dynamischen Simulationen (WAF-2b- ϑ_b) in Tagen	220

Abkürzungsverzeichnis

AMICE	European CIM Architecture Consortium
ANSI	American National Standards Institute
AwF	Ausschuss für Wirtschaftliche Fertigung e.V.
CIM	Computer Integrated Manufacturing
CPS	Cyber-Physisches System / Cyber-Physical System
DIN	Deutsches Institut für Normung e. V.
DLZ	Durchlaufzeit
EN	Europäische Norm
ERP	Enterprise Resource Planning
FSP	Flow Shop Problem
GA	Genetischer Algorithmus
GK	Globales Kriterium
GSP	Group Shop Problem
GTA	Giffler und Thompson Algorithmus
GUI	Graphical User Interface / Grafische Benutzeroberfläche
GW	Grenzwert
IEC	International Electrotechnical Commission / Internationale Elektrotechnische Kommission

IOSB	Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung
ISA	Instrument Society of America
ISO	International Organization for Standardization
IT	Informationstechnik
JSP	Job Shop Problem
KPI	Key Performance Indicator (Leistungskennzahl)
MES	Manufacturing Execution System
MESA	Manufacturing Enterprise Solutions Association
MOM	Manufacturing Operations Management
MRP II	Manufacturing Resource Planning
NIST	National Institute of Standardization (USA)
OSP	Open Shop Problem
PERA	Purdue Enterprise Reference Architecture
PPS	Produktionsplanung und -steuerung
PRM	Purdue Reference Model
REFA	Verband für Arbeitsstudien und Betriebsorganisation e.V. (bis 1951 als Reichsausschuss für Arbeitszeitermittlung)
TSC	Total Setup Cost / Summe der Rüstkosten
TST	Total Setup Time / Summe der Rüstzeiten

VDI	VDI Verein Deutscher Ingenieure e.V.
VDMA	Verband Deutscher Maschinen- und Anlagenbau

Nomenklatur

J	Menge der Aufträge
O	Menge der Operationen
O_j	Menge der Operationen von Auftrag j
o_{ji}	Operation i des Auftrags j
M	Menge der Maschinen
μ_k	Maschine k
M_{ji}	zulässige Maschinen für o_{ji}
μ_{ji}	zu o_{ji} zugeordnete Maschine
n	Anzahl Aufträge, $ J $
m	Anzahl Maschinen, $ M $
\mathcal{R}	Menge der zusätzlichen Ressourcen
R_k	Ressource k
\mathcal{R}_{ji}	zulässige Ressourcenkombinationen für o_{ji}
R_{ji}	zu o_{ji} zugeordnete Ressourcenkombinationen
t_0	aktueller Zeitpunkt
t^p	Referenzzeitpunkt
r_j, r_{ji}	frühester Startzeitpunkt des Auftrags j bzw. o_{ji} ,

d_j, d_{ji}	spätester Fertigstellungszeitpunkt des Auftrags j bzw. o_{ji}
p_{ji}	Bearbeitungsdauer von o_{ji}
w_j	Gewicht des Auftrags j
S_j, S_{ji}	geplanter Startzeitpunkt des Auftrags j bzw. o_{ji}
C_j, C_{ji}	geplanter Endzeitpunkt des Auftrags j bzw. o_{ji}
st_{ijk}	1. anfallende reihenfolgenabhängige Rüstzeit bei einem Übergang zwischen Operationen der Aufträge j und k auf μ_i 2. wird auch als Indikator für reihenfolgenabhängige Rüstzeiten in der Problemstellung verwendet
YS_{ji}	geplanter Startzeitpunkt des Rüstvorgangs vor o_{ji}
YC_{ji}	geplanter Endzeitpunkt des Rüstvorgangs vor o_{ji}
Y_{ji}	Geplanter Rüstzustand der o_{ji}
YR_{ji}	Umrüstkosten von o_{ji}
L_j	Terminabweichung von Auftrag j
T_j	Verspätung von Auftrag j
U_j	Indikator für Termineinhaltung ($U_j = 0$) bzw. Verspätung ($U_j = 1$) von Auftrag j
E_j	Verfrühung eines Auftrags j
$C_j^\theta(t)$	C_j für $S_j = t$ von Auftrag j , falls $J = \{j\}$
P_i	Fertigungsplan i

$P_{\#}$	Referenzfertigungsplan
v_{ji}	notwendige Vorlaufzeit der Operation o_{ji}
$\vartheta^a(P_i, P_k)$	Abstand zwischen P_i und P_k
$\vartheta(P_i)$	Abstand zwischen P_i und $P_{\#}$
$\vartheta^b(P_i)$	Einhaltung der Benutzerinteraktionen in P_i
$\theta(P_k)$	Akkumulierte Planstabilität bei Übernahme von P_k
θx	Akkumulierte Planstabilität berechnet aus den letzten x Übernahmen
$g(t)$	Gewichtungsfunktion für Zeitpunkte
b_k	Benutzerinteraktion k
$g^b(b_k)$	Gewichtungsfunktion für das Alter von b_k
d_h	typischer Zeithorizont in der Problemstellung
$g_o(t)$	konstante Funktion definiert als $g_o(t) \equiv 1$
$g_o g_x$	zweistufiger Vergleichsoperator für zeitbehaftete Versionen von Bewertungskriterien; bei einer Differenz mit g_o kleiner als 3% wird mit g_x verglichen

1 Einleitung

1.1 Ausgangssituation

Existenz-, Innovations- und Kostendruck treiben produzierende Unternehmen zur ständigen Optimierung ihrer Fertigung. Die zunehmend kurzfristigen Änderungen in der Produktpalette und den Fertigungsanlagen und die stark schwankende Auftragslage stoßen neue Optimierungsanstrengungen an. Eine Fertigungsfabrik ist heute ständig im Wandel.

Je nach Branche haben die Unternehmen unterschiedliche Optimierungsschwerpunkte: Autobauer wollen ihre Taktzeiten reduzieren, Gießereien ihren Energiebedarf senken oder das Material reduzieren. Auf der Tagesordnung der Unternehmen jeglicher Branchen steht jedoch die Optimierung des täglichen Fertigungsablaufs. Die Fertigung soll die Aufträge ohne Verspätung abarbeiten und negative Folgen von Anlagenstörungen vermeiden. Parallel dazu soll sie für niedrige Rüstzeiten, Durchlaufzeiten und Materialbestände sowie kürzere Transportwege und eine hohe, gleichmäßige Auslastung sorgen.

Die Verfolgung dieser vielfältigen, teils widersprüchlichen Optimierungsziele hat einen komplexen inner- und außerbetrieblichen kontinuierlichen Geschäftsprozess zum Ergebnis. Er involviert Beteiligte aus fast allen betrieblichen Bereichen von Vertrieb und Einkauf über Lagerverwaltung und Werkzeugaufbereitung bis hin zu Fertigung und Transport. Softwaresysteme für Produktions- und Unternehmensmanagement entlasten die sonst aufwändige Kommunikation zwischen den Beteiligten und sollen mithilfe unterschiedlicher Optimierungsverfahren auch Aufgaben in der Fertigungsplanung erledigen.

Das Thema Fertigungsfeinplanung ist seit mehr als 70 Jahren, somit fast schon seit dem Beginn der Informatik, kontinuierlich Gegenstand der Forschung, da bereits kleine Verbesserungen der Fertigungsabläufe hohe finanzielle Auswirkungen haben können. Die Fertigungsfeinplanung ist als theoretische Problemstellung sehr anspruchsvoll und bis jetzt nicht optimal gelöst, da schon wenige Fertigungsaufträge und wenige Fertigungsanlagen die Anzahl möglicher Fertigungspläne drastisch erhöhen. Aus diesem Grund gibt es unterschiedliche Vorschläge für Feinplanungsverfahren aus der Informatik, der Mathematik und den Wirtschaftswissenschaften.

In den letzten Jahren hat sich ein Wandel vollzogen, der sich in den zunehmenden Standardisierungsbemühungen im Fertigungsbereich zeigt. ISA-95 Standard (auch als ISO/IEC 62264 bekannt) [Ansi10], VDI 5600 [Vdi07], VDMA 66412 [Vdma09] und ISO 22400 [Iso14] streben die Unifizierung der Abläufe, Schnittstellen und Kennzahlen in der Fertigung an. Einer der zentralen Aspekte in den Standards ist auch die Fertigungsteuerung und die darin enthaltene Fertigungsfeinplanung.

Die Unternehmen fokussieren sich zunehmend auf die Optimierung ihres Fertigungsablaufs. Nachdem auch sehr kleine Unternehmen Systeme für die Verwaltung und Steuerung ihrer Geschäftsprozesse im Betrieb nutzen, plant die Industrie Investitionen in Fertigungsmanagementsysteme. Der Markt an Fertigungsmanagementsystemen soll bis 2018 kontinuierlich mit zweistelliger Prozentzahl jährlich wachsen [Mark13, Tech13].

Industrieländer, internationale Organisationen und einflussreiche Konzerne haben diese Tendenz erkannt und verschiedene Geschäftsmodelle entworfen. Die deutsche Regierung startete die Initiative „Industrie 4.0“ mit dem Ziel, die Automatisierungs- und Vernetzungsbestrebungen in der Produktion zu kanalisieren [KaWH12], um zukünftige Produktangebote zu vereinheitlichen.

In Nordamerika haben die einflussreichen Firmen Cisco und General Electric mit Connected Manufacturing [HKSC06] bzw. Industrial Internet [EvAn12] ähnliche Konzepte ins Leben gerufen. Die zunehmende Automatisierung und Vernetzung der Fertigungsfabriken erfordert eine reaktionsfähigere, intelligenterere und kooperative Feinplanung für deren Steuerung.

1.2 Motivation

Obwohl viele fortschrittliche Feinplanungsverfahren mit Blick auf einen praktischen Einsatz in der Fertigung entwickelt wurden, fand nur eine sehr kleine Anzahl Einzug in kommerzielle Fertigungsmanagementsysteme. Der Hauptgrund ist die Vernachlässigung des menschlichen Faktors im Planungsprozess [McWi06, S.177–178, McWi99, Weze06]. Die Feinplanungsverfahren können nur unzureichend Umplanungen von den Planern berücksichtigen, nehmen oft umfangreiche, nicht nachvollziehbare Veränderungen im Fertigungsplan vor und gefährden dadurch die Planstabilität und -akzeptanz. Eine Zusammenarbeit zwischen Planern und Feinplanungsverfahren ist auf dieser Grundlage nicht möglich. Dies zwingt den Planer dazu, die Oberhand über den auszuführenden Fertigungsplan zu behalten, den Einfluss des Feinplanungsverfahrens einzuschränken und es nur bei akutem Bedarf auszuführen. Damit ist das Feinplanungsverfahren vom Fertigungsablauf entkoppelt und kann bei Störungen und Abweichungen nicht eingreifen [RHMD10a, S.315].

Trotz des hohen Bedarfs sind kooperative Feinplanungssysteme zurzeit nicht verfügbar. In der Fertigungssteuerung in Unternehmen werden heute deshalb immer noch rudimentäre Feinplanungsverfahren wie selbst erstellte Tabellenkalkulationen in Microsoft Excel oder klassische Karteisysteme eingesetzt.

1.3 Zielsetzung und eigene Beiträge

Ziel dieser Arbeit ist die Konzipierung eines Ansatzes für eine benutzerfreundliche, praxistaugliche Fertigungssteuerung unter Verwendung von etablierten Feinplanungsverfahren. Die Feinplanungsverfahren sollen in Kollaboration mit den Planern agieren und möglichst optimal die zur Verfügung stehende Rechenleistung nutzen, um nachvollziehbare Plananpassungen u. a. bei unplanmäßigen Fertigungsereignissen durchzuführen.

In dieser Arbeit werden die Gründe für die geringe Verbreitung von fortschrittlichen Verfahren für Fertigungssteuerung untersucht. Die Analyse berücksichtigt die Sichtweise und die Wünsche des „typischen“ Verantwortlichen für die Fertigungssteuerung in Unternehmen sowie die zukünftige umfangreiche Vernetzung in der Fertigung. Dabei werden drei Bedarfspunkte bei der Zusammenarbeit zwischen Mensch und Optimierungsverfahren identifiziert. Durch eine bessere Einhaltung der Benutzerinteraktionen, Sicherung der Planstabilität und die Konzentration der Optimierung auf Teile des Fertigungsplans (nachfolgend Fragmente genannt) mit Optimierungsbedarf und -potential soll das Verhalten eines permanent laufenden Feinplanungsverfahrens dem eines menschlichen Planers ähneln.

Klassische Bewertungskriterien für Fertigungspläne werden erweitert, um eine stärkere Gewichtung auf Fragmente des Fertigungsplans mit höherer Relevanz zur aktuellen Fertigungssituation sicherzustellen und deren Optimierungsbedarf einzuschätzen. Ferner wird der Spielraum für eine Verbesserung innerhalb der Fragmente gemessen, um das Optimierungspotential zu errechnen. Fragmente mit einer überdurchschnittlich hohen Kombination aus Optimierungsbedarf und -potential sollen vom Feinplanungsverfahren optimiert werden. Das soll unterschiedlichen, auch stochastischen Feinplanungsverfahren ein nachvollziehbares Verhalten geben.

Mithilfe der erweiterten Kennzahlen kann ein Feinplanungsverfahren lang-, mittel- und kurzfristig unterschiedliche Ziele verfolgen und gleichzeitig für die lang- und mittelfristige Planung als auch für die unmittelbare Fertigungssteuerung eingesetzt werden. Diese oft dreistufige Fertigungsplanung und -steuerung wird somit mittels eines Systems und Feinplanungsverfahrens realisierbar.

Zur Sicherung der Planstabilität werden entsprechende Kennzahlen aus der Forschung für den Einsatz in einer laufenden Fertigung erweitert. Sie sollen nicht nur die anstehenden Veränderungen im Fertigungsplan berücksichtigen, sondern auch die Änderungshistorie, um die Planstabilität nicht durch viele kleine Veränderungen zu gefährden.

In der Arbeit wird für die Einhaltung der Benutzerinteraktionen eine neue Kennzahl definiert, die den Unterschied zwischen den vom Feinplanungsverfahren geplanten und den vom Benutzer gewünschten Ausführungszeiten der Aufträge bewertet. In die Berechnung fließen die getätigten Ressourcenwechsel sowie die Zeitpunkte der Interaktionen ein.

Um die verfügbare Rechenzeit optimal zu nutzen, wird zusätzlich ein Ansatz aufgezeigt, um Fragmente des Fertigungsplans mit Optimierungspotential und -bedarf abzugrenzen und zur exklusiven Optimierung an Verfahrensinstanzen zu übergeben. Dieser Ansatz soll die parallele Ausführung mehrerer selbstständiger unabhängiger Verfahrensinstanzen ohne zusätzliche Steuerungsinstanz ermöglichen. Neben der optimalen Auslastung gegenwärtiger Rechnersysteme soll sich die Optimierungszeit reduzieren, da die einzelnen Fragmente als kleine Unterprobleme handhabbarer für ein Feinplanungsverfahren sind.

Die aufgezählten Beiträge und Ansätze, die in dieser Arbeit entwickelt werden, sind notwendig, um das Verhalten eines Feinpla-

nungsverfahrens der Art und Weise eines menschlichen Planers anzupassen. Um dies darzulegen, werden Simulationsläufe mit Industriedaten in einer hierfür erstellten dynamischen Simulationsumgebung durchgeführt.

1.4 Aufbau der Arbeit

Die Arbeit ist in sechs Kapitel gegliedert. Die Struktur und die Zusammenhänge zwischen den Kapiteln sind in Abbildung 1.1 dargestellt.

Kapitel 2 beschreibt die unterschiedlichen Fertigungssteuerungsprobleme aus theoretischer Sicht sowie in der Forschungswelt untersuchte Verfahren zu deren Lösung. Ferner beschreibt es Ansätze für den Einsatz von Verfahren in der Praxis.

Kapitel 3 analysiert die Ursachen für die fehlende Akzeptanz und den breiteren Einsatz bestehender kommerzieller Fertigungssteuerungssysteme und leitet Anforderungen an ein benutzerfreundliches Planungsverfahren ab.

Ausgehend von den Anforderungen wird ein Konzept für eine Feinplanungsplattform zur Unterstützung von Fertigungssteuerungsverfahren in Kapitel 4 entwickelt. Es erläutert verschiedene Aspekte der Feinplanungsplattform, die die gestellten Anforderungen erfüllen.

Die Implementierung des Gesamtkonzepts ist in Kapitel 5 ausführlich dargestellt. Besonderes Gewicht wird dabei auf die Implementierungsentscheidungen gelegt, mittels derer die geforderte Performanz und die Vereinfachung der Plattformkonfiguration realisiert werden.

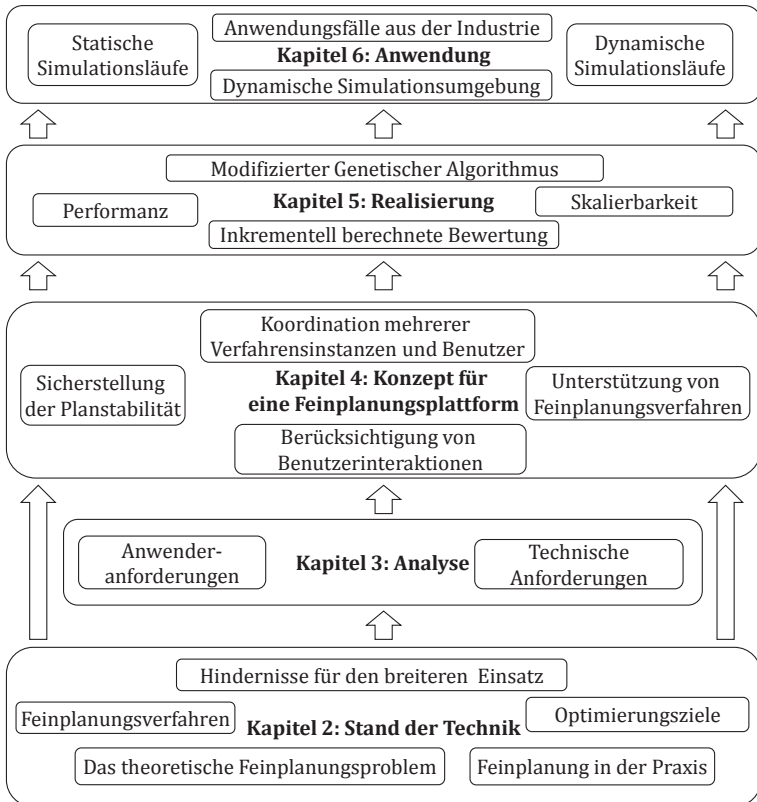


Abbildung 1.1: Aufbau der Arbeit

Die Validierung des Konzepts erfolgt in Kapitel 6. Anschließend werden die Ergebnisse der Simulationsläufe mit der prototypisch implementierten Feinplanungsplattform dem konzipierten Verhalten gegenüber gestellt.

Die Zusammenfassung der Ausarbeitung und einen Ausblick gibt Kapitel 7 wieder.

2 Stand der Technik

Dieses Kapitel erläutert den Begriff Fertigungssteuerung im Abschnitt 2.1. Anschließend stellt Unterkapitel 2.2 die Sicht auf die Fertigungssteuerung als mathematische Aufgabe und listet häufig verwendete Optimierungsziele und -modelle auf. Bewährte Feinplanungsverfahren für die statische und die dynamische Variante dieser mathematischen Aufgabe sind in 2.3 bzw. 2.3.6 zu finden. Die Kollaboration zwischen menschlichem Benutzer (Planer) und algorithmischer Optimierung in der Fertigungssteuerung wird in 2.5 erörtert. Unterkapitel 2.6 gibt die praktische Sicht auf die Fertigungssteuerung wieder. Dort sind relevante Standardisierungen, kommerzielle Fertigungssteuerungssysteme und in der Praxis eingesetzte Verfahren erläutert.

2.1 Der Begriff „Fertigungssteuerung“

Im täglichen Sprachgebrauch wird der Begriff Fertigung oft als Synonym für Produktion verwendet [KSWA03]. In der Fachsprache ist die Fertigung als ein Teilbereich der Produktion zu verstehen [MeTo95, S.178, 382]: „Die Fertigung umfasst alle organisatorischen und technischen Maßnahmen zur Herstellung von Material oder Erzeugnissen“ während „die Produktion alle Vorgänge und Tätigkeiten, die unmittelbar oder mittelbar zur Herstellung von Erzeugnissen dienen“, umfasst. Die Produktion umfasst neben der Fertigung auch weitere Teilbereiche wie Konstruktion, Beschaffung oder Qualitätssicherung. Die Aufgabe der Fertigungssteuerung ist das „Veranlassen, Überwachen und Sichern der Durchführung von Fertigungsaufgaben hinsichtlich Bedarf (Menge, Termin), Qualität, Kosten und Arbeitsbedingungen“ [Vdiv83, S.74] und wird als Funktion der Produktionssteuerung gesehen [MeTo95, S.197].

2.1.1 Evolution des Begriffs und Einordnung in der betrieblichen Produktion

Die heutige Bedeutung der Fertigungssteuerung unter den betrieblichen Funktionen ist Ergebnis des ununterbrochen steigenden Automatisierungsgrads im Produktionsumfeld.

Seit der Erfindung des Fließbands im 19ten Jahrhundert reduziert sich die Aufenthaltszeit eines Werkstücks in der Fertigung kontinuierlich. Die Erfindung des Rechners und dessen Einsatz in der Maschinensteuerung in den 60er Jahren prägten den Wunsch nach einer rechnergestützter Steuerung des gesamten Maschinenparks. In [Bran67] wurde einen Regelkreis vorgeschlagen, der die Terminierung der Fertigungsaufträge innerhalb der Fertigungssteuerung mittels rechnergestützter und manueller Eingriffe unter möglichen Abweichungen in der Auftragsausführung durchführen soll (s. Abbildung 2.1). Dieser Regelkreis ist in leicht abgewandelter Form in heutigen zentralisierten Steuerungskonzepten zu finden.

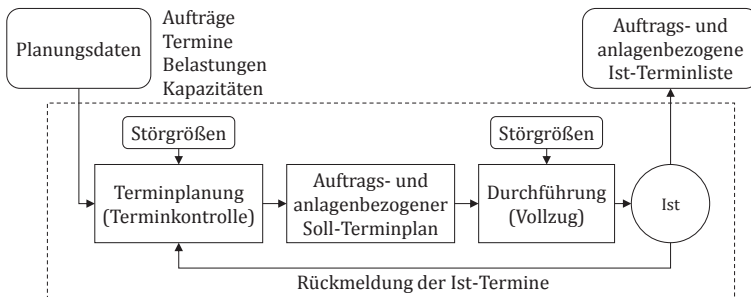


Abbildung 2.1: Terminregelung innerhalb der Fertigungssteuerung [Bran67]

In den 1970ern wurde das Manufacturing Resource Planning (MRP II) Konzept entwickelt, mit dem Ziel alle Unternehmensressourcen effizient zu planen [Wigh84]. Innerhalb dieses Konzepts konzentriert sich die Fertigungssteuerung auf die Terminierung von Fertigungsaufträgen für einen kurzen Zeithorizont.

In den 1980ern wurden viele Modelle für eine ganzheitliche Planung aller Produktionsprozesse unter dem Stichwort Produktionsplanung und -steuerung (PPS) präsentiert. Statt einer einheitlichen Definition schlagen Forschungs- und Industrieorganisationen wie REFA, VDI, AwF, AMICE, das Stuttgarter Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA) oder die Universität Bordeaux unterschiedliche Deutungen des PPS Begriffs vor [UhKü88, Hack89, Refa90, Amic91, DVZC92]. Die Unterschiede liegen in der Abgrenzung der PPS von den anderen betrieblichen Prozessen und in der Aufteilung der Aufgaben.

Einen gewissen Bekanntheitsgrad erreicht das Aachener PPS Modell [Horn96], das noch heute verwendet und erweitert wird. Es unterscheidet zwischen Kern- und Querschnittsaufgaben sowie zwischen Fremdbezug und Eigenfertigung (s. Abbildung 2.2). Die Fertigungssteuerung ist hier nur Teil der Eigenfertigungsplanung und -steuerung. Das Aachener PPS Modell ist auch heute noch aktuell und wird weiterentwickelt bzw. verfeinert [ScGi06].

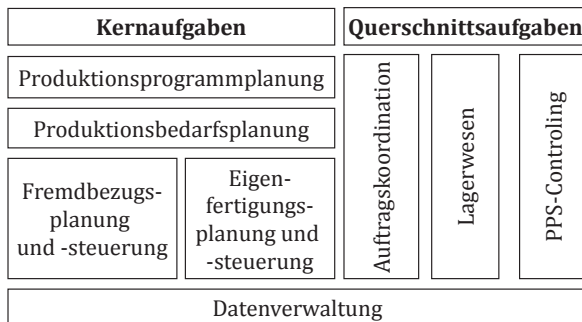


Abbildung 2.2: Aachener PPS Modell

In den 1980ern wurde mit der Einführung des Computer Integrated Manufacturing (CIM) Begriffs auch PPS dort integriert [Awf85]. Das Y-Modell für CIM [Sche87] ist das bekannteste Modell in Europa, auch aufgrund seiner umfassenden und klaren Betrachtung vieler Produktionsaspekte. In ihm sind die einzelnen Aufgaben bzw. Systeme nach technischer und betriebswirtschaftlicher sowie nach Planungs- und Realisierungsphase aufgeteilt. Hier ist die Bedeutung der Fertigungssteuerung höher und eigenständig gekennzeichnet (s. Abbildung 2.3). Nach [Sche87, S.24] führt die Fertigungssteuerung die Optimierung der Fertigungsaufträge nach neuen Optimierungskriterien wie Verschnitt- und Rüstoptimierung durch, die in der Planungsphase nicht betrachtet werden.

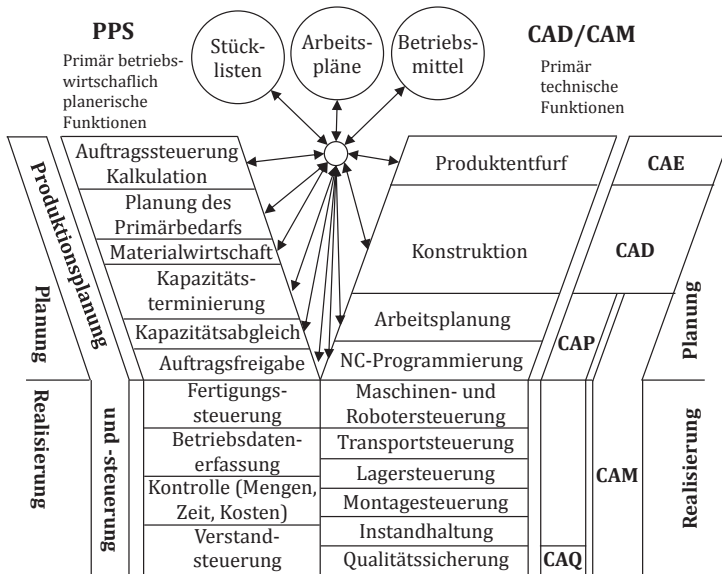


Abbildung 2.3: Das Y-Modell [Sche87]

Ein weiteres Modell ist das von der Universität Purdue (Indiana, USA) vorgeschlagene Purdue Reference Modell (PRM) for CIM [Will89]. PRM ordnet Planungs- und Steuerungsfunktionen ver-

schiedenen Ebenen zu (s. Abbildung 2.4). Gleichzeitig definiert PRM die Beziehungen und die Abläufe zwischen den einzelnen Funktionen. Bei den Abläufen spielt die Fertigungssteuerung eine zentrale Rolle.

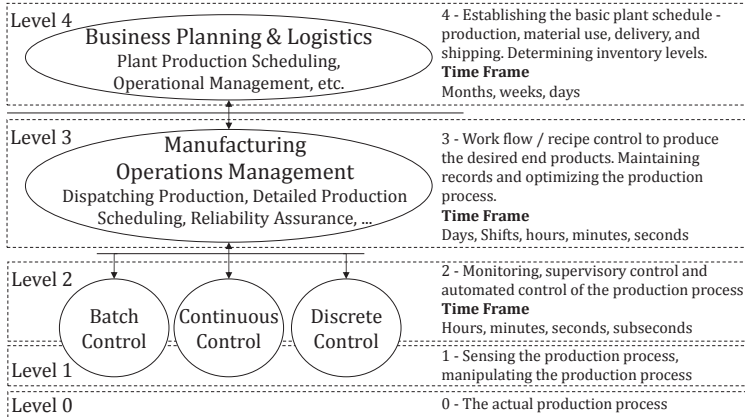


Abbildung 2.4: Das Ebenenmodell von PRM zusammengefasst in [Ansi13a]

PRM erreichte im englischsprachigen Raum eine gewisse Verbreitung, weshalb die Universität Purdue auf diesem Gebiet aktiv blieb und auch die Purdue Enterprise Reference Architecture (PERA) vorstellte [Will98]. Die in Deutschland entwickelten PPS bzw. CIM Modelle wurden hauptsächlich im deutschsprachigen Raum bekannt. Das Akronym PPS schaffte jedoch einen Transfer als „Production Planning and Scheduling“ oder als „Production Planning and Control“ (PPC) in den englischen Begriffsraum und das deutsche Wort „Leitstand“ wurde als Bezeichnung für grafisches Werkzeug zur Visualisierung des Fertigungsfortschritts und der Arbeitsver-/zuteilung [MeTo95, S.304] direkt übernommen [AdKa91].

Die CIM Modelle sind großteils vom damaligen Stand der EDV-Technik beeinflusst. Die einzelnen Funktionen werden von separaten Systemen übernommen. Der Datenaustausch zwischen den

Systemen wurde minimiert, da die Schnittstellen und die Netzwerkverbindungen nicht dem heutigen Stand entsprachen. Die Systeme waren selbst für die Ausführung auf einem Rechner (Server oder Mainframe) vorgesehen.

Die Verbreitung der Systeme für die unternehmerische Ressourcenplanung (Enterprise Resource Planning Systeme, ERP) wie SAP R/3 in den 90er Jahren zwang zum Überdenken der vorigen Konzepte für PPS. Da ein ERP System viele Planungsfunktionen des PPS übernimmt, ist die Nachfrage nach kompletten integrierten PPS Systemen stark gesunken. Der Bedarf am Steuerungsteil hat sich nicht verändert oder sogar erhöht. Um die entstandene Lücke zwischen ERP und den einzelnen Maschinensteuerungen der sog. Fabrikebene zu füllen, haben sich über die Zeit Hilfssysteme, zusammengefasst unter dem Begriff Manufacturing Execution System (MES), etabliert [YPHY10]. Die heute entstandene, allgemein akzeptierte Automatisierungspyramide in einfacher Ausprägung ist in Abbildung 2.5 dargestellt.

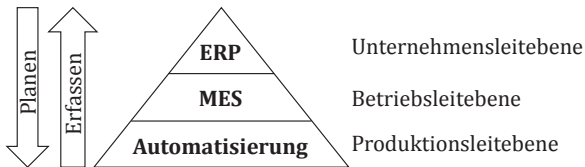


Abbildung 2.5: Die Automatisierungspyramide

Der Begriff MES wird heute als Fertigungsmanagementsystem übersetzt [Vdi07], das je nach Definition verschiedene Hauptfunktionen umfasst. Nach [Vdi07] sind es acht Hauptfunktionen, wobei der auf Abbildung 2.1 angezeigte Regelkreis in abgewandelter Form in der Hauptfunktion „Feinplanung und Feinsteuerung“ zu finden ist (s. Abbildung 2.6).

MESA dagegen beschreibt elf Hauptfunktionen [Mesa97, S.12]. Die einzelnen Aufgaben der Fertigungssteuerung aus der Definition des VDI [Vdiv83, S.74] sind wie folgt verteilt: Hauptsächlich

geschieht das Veranlassen von Fertigungsaufgaben durch die Funktion „Dispatching Production Units“; die Überwachung wird großteils von der Funktion „Resource Allocation and Status“ übernommen und das Sichern der Durchführung wird durch das „Operations/Detail Scheduling“ realisiert. Die restlichen acht Funktionen unterstützen die Fertigungssteuerung teilweise.

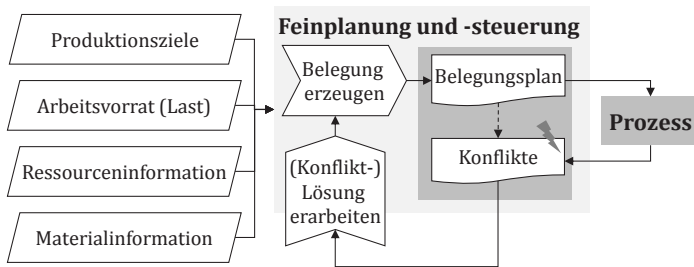


Abbildung 2.6: Feinplanung und Feinsteuerung [Vdi07, S.17]

Die Aufteilung der Ebenen der Automatisierungspyramide in Abbildung 2.5 ähnelt dem PRM, da ERP der Ebene 4 und MES der Ebene 3 im PRM zugeordnet werden kann. Aus diesem Grund nahm die International Society of Automation (ISA) PRM als Basis für den ISA-95 Standard [Ansi10]. Dieser Standard wurde später von IEC in Zusammenarbeit mit ISO mit leichten Modifikationen unter der Nummer ISO/IEC 62264 veröffentlicht. Die erste im Jahr 2000 erschienene Version wurde 2008 in Europa und Deutschland unter DIN/EN 62264 übersetzt [Dine08]. Auf Ebene 3 wird das operative Produktionsmanagement (Manufacturing Operations Management/MOM) definiert (s. Abbildung 2.7). Die Fertigungssteuerung findet hier in der Kategorie „Production Operations“ statt. In der deutschen Übersetzung [Dine08] dieser englischen (amerikanischen) Norm wird „manufacturing“ nicht als „Fertigung“ sondern als „Produktion“ übersetzt und nicht auf den Unterschied zwischen Produktion und Fertigung eingegangen. Auch „production scheduling“ wird als Disposition und nicht entsprechend VDI-5600 als Feinplanung übersetzt.

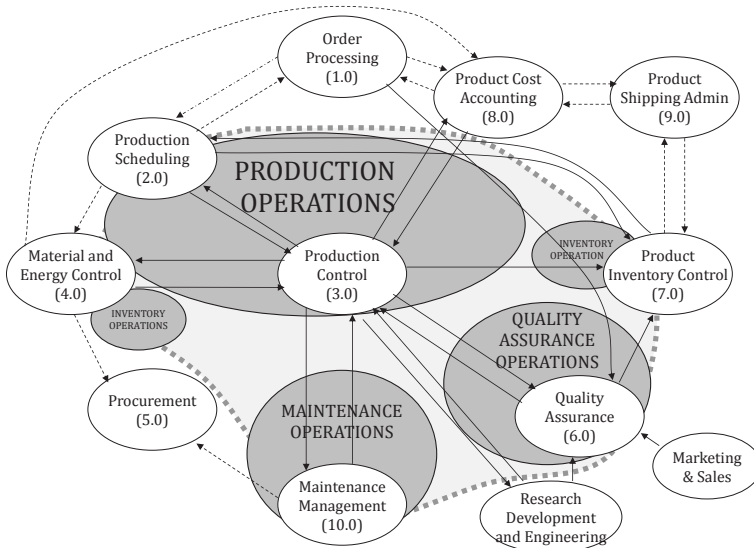


Abbildung 2.7: Modell des operativen Produktionsmanagements [Ansi13a]

Der ISA-95 Standard erweitert den Regelkreis der Fertigungssteuerung (s. Abbildung 2.8). Die erfassten Daten (Production Data Collection) aus der Fertigung werden weiter analysiert (Production Performance Analysis) und die Ergebnisse fließen in die Überwachung des Produktionsfortschritts (Production Tracking) und von dort aus in die Feinplanung (Detailed Production Scheduling) ein. Die Linien „A“ bis „E“ zeigen mögliche hypothetische Grenzen zwischen Ebene 4 (z. B. ERP-Systeme) und MOM. Je nach Fertigungsart und Organisation kann die Fertigungssteuerung auf unterschiedliche Systeme verteilt sein.

Die Initiative „Industrie 4.0“ der Bundesregierung gibt Umsetzungsempfehlungen für die zukünftige Produktions-IT [KaWH12]. Aktuelle IT-Konzepte wie Cloud Computing [BYVB09] und cyberphysische Systeme (CPS) [Lee08, SGLW08] sollen die Unternehmensanwendungen (ERP, MES usw.) und Unternehmensressourcen aufwerten.

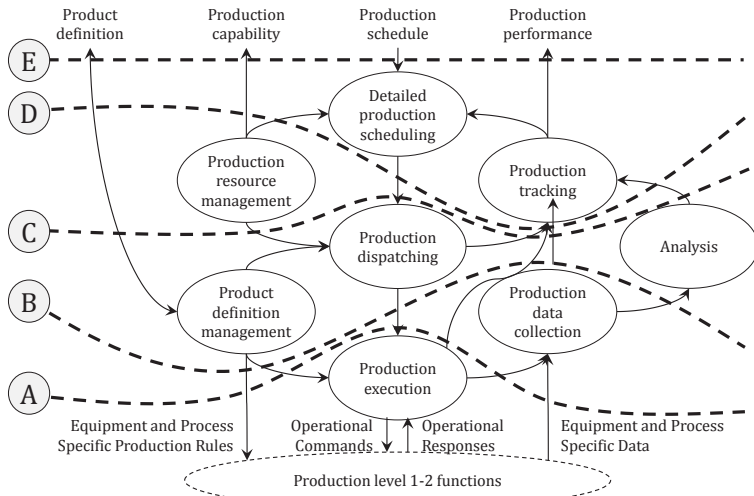


Abbildung 2.8: Aktivitätsdiagramm des „Production Operations Management“ [Ansi13a]

Aus IT-Sicht ist ein CPS ein System bestehend aus Sensoren, Mechanik und Software, die auch miteinander kommunizieren können, z. B. über Internet. Unternehmensressourcen wie Maschinen, Werkzeuge, Transporteinheiten oder Werkstücke können als CPS implementiert werden und Teile der zentralen Fertigungssteuerung übernehmen.

Cloud Computing ist ein Ansatz zur dynamischen Zuteilung, Verwendung und Freigabe von IT-Ressourcen (Datenspeicher, Rechenleistung, Netzwerkkapazitäten) bei Bedarf und bei minimalem Verwaltungsaufwand [MeGr09]. Für die Feinplanung der Fertigungssteuerung bietet das Cloud Computing eine hohe verfügbare Rechenleistung, die bei Bedarf auch kurzfristig fortschrittlichen Feinplanungsverfahren zugeteilt werden kann.

Aus Sicht der Industrie 4.0 soll es in Zukunft keinen starren Regelkreis der Fertigungssteuerung mehr geben, sondern die einzelnen Ressourcen sich selbst organisieren können und Aufgaben

wie Datenerfassung und -analyse übernehmen, während z. B. zentrale oder übergeordnete Dienste mittels Cloud Computing realisiert werden können. Die Planungs- und Steuerungshierarchie in der Automatisierungspyramide soll aufgelöst werden [VKBW09]. Alle Beteiligten in der Fertigung sollen mittels eines einheitlichen Informationsmodells und Internetverbindung in der Lage sein, die notwendigen Informationen auch ohne Vermittler auszutauschen (s. Abbildung 2.9 mit einer Variante der Automatisierungspyramide).

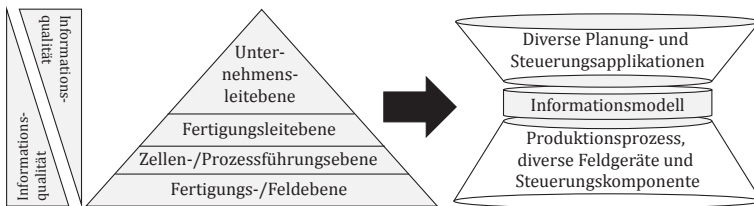


Abbildung 2.9: Veränderung der Informationsarchitektur in Produktionsunternehmen [VKBW09]

Die Industrie 4.0 schlägt einen direkten Informationsweg zwischen Steuerungs- und Ausführungsinstanzen vor, der in der umfassend vernetzten Fertigung immer herstellbar sein wird. Die Steuerungslogik wird auf die einzelnen Instanzen verteilt, was zu einer Steigerung der Anforderungen für Kollaboration und Reaktionsfähigkeit in der gesamten Fertigung führt.

2.1.2 Bedeutung für die Unternehmen

Die Evolution des Begriffs Fertigungssteuerung und seiner Bedeutung zeigt die ständig wachsenden Erwartungen der Unternehmen (s. Abbildung 2.10). Zum ursprünglichen Ziel am Anfang des 20sten Jahrhunderts, die Fabriken voll auszulasten, kamen weitere Zielsetzungen wie die Einhaltung zugesicherter Liefertermine. Später [Sche87, S.24–25] kam noch die Rüst- und Verschnittoptimierung hinzu. Heute soll eine Reihe von Produktions-

kennzahlen, sog. Key Performance Indicators (KPIs), neben den reihenfolgenabhängigen Rüstzeiten oder der Ausfallsicherheit in der laufenden Fabrik mittels einer geeigneten Fertigungssteuerung verbessert werden.

	Verfolgte Ziele	Rechnerunterstützung	Ansätze für Fertigungssteuerung	Verbreitete Feinplanungsverfahren
Vor 1960	Fabrikauslastung	keine	Räumliche Aufteilung	keine
1960-1980	+Terminreue +Durchlaufzeiten	Centralized Computing (Mainframe)	Funktionale Aufteilung	Einfache heuristische Konstruktionsverfahren
1980-2000	+Verschnitt- +Rüsto Optimierung	Grid Computing	Zentralisierte Fertigungssteuerung	Einfache heuristische Konstruktionsverfahren
2000-2010	+OEE	Utility Computing	Hierarchische Aufteilung	Einfache heuristische Konstruktionsverfahren
Ab 2010	+KPIs nach ISO 22400 und VDMA 66412	Cloud Computing Cyber-Physical Systems	Industrie 4.0	Einfache heuristische Konstruktionsverfahren

Abbildung 2.10: Evolution relevanter Aspekte der Fertigungssteuerung

Um das zu erreichen, erforschten die Unternehmen verschiedene Ansätze zur Fertigungssteuerung. Um die Komplexität der Fertigungssteuerung in den Griff zu bekommen, wurden in der Vergangenheit verschiedene Aufteilungen untersucht. Zu Zeiten, in denen noch keine Rechnerunterstützung bestand, überließ man die Fertigungssteuerung den Hallen- und Schichtmeistern. Später realisierte man im Rahmen von CIM die einzelnen Aufgaben der Fertigungssteuerung als separate Systeme. Die Konzentration der Fertigungssteuerung auf einem Rechner brachte nicht den versprochenen Erfolg, weshalb sie auf mehrere Ebenen verteilt wurde. Heute wird innerhalb des Konzepts „Industrie 4.0“ versucht, mit verteilter Steuerungslogik die Fertigungssteuerung in den Griff zu bekommen.

Das bis dato nicht optimal oder zumindest nicht zufriedenstellend gelöste Problem der Fertigungssteuerung besteht in der Verteilung der Fertigungsaufgaben auf die Fertigungsressourcen. Dies geschieht in kommerziellen Systemen immer noch mit sehr

einfachen Regeln, obwohl in der Forschung sehr komplexe Verfahren untersucht wurden. Dies veranschaulichen die nächsten Unterkapitel.

2.2 Das Shop Scheduling Problem

Die Problemstellung der Fertigungssteuerung ist sehr vielfältig und die verschiedenen Arten sind in der Literatur unter Shop Scheduling Problem untergliedert. Der Begriff „shop“ oder „shop floor“ bezieht sich auf das Stockwerk (floor) in einer Fabrik, einer Werkstatt, einem Lagerhaus usw. (shop), auf dem die Maschinen und/oder das ausführende Personal angesiedelt sind.

Der Begriff „scheduling“ hat kein deutsches Äquivalent und wird oft mit Feinplanung, Ablaufplanung oder Steuerung übersetzt [EvKe00]. Nach [Pine95] bedeutet Scheduling „die Zuordnung von Aufgaben (tasks) auf beschränkte Ressourcen (limited resources), um bestimmte Zielvorgaben zu optimieren“.

Nachfolgend werden eine formale Definition des Shop Scheduling Problems gegeben sowie die Zuordnung der Komplexitätsklasse und häufig eingesetzte Optimierungsziele beschrieben.

2.2.1 Mathematische Definition

Das Shop Scheduling Problem wird durch eine Menge von Aufträgen definiert, die jeweils eine Menge von Operationen enthalten, die einer Menge von Maschinen zeitlich zugeordnet werden sollen. Die Bezeichnungen für diese Objekte stehen stellvertretend für die mittlerweile im englischsprachigen Raum etablierten Begriffe „job“, „operation“, „machine“ [Leun04a, Pine14]. Nachfolgend wird eine erweiterte praxisnahe mathematische Formulierung des Problems in Anlehnung an [BEPT04, S.38–39, Bruc07,

S.1–2] unter Berücksichtigung etablierter Namenskonventionen gegeben:

- $J := (j_1, \dots, j_n)$ Menge von n Aufträgen (jobs),
- $O := \bigcup_{j \in J} O_j$ die Menge aller Operationen (operations, tasks, auch Arbeitsgänge, Arbeitsschritte, Arbeitsvorgänge), wobei $O_j := (o_{j1}, \dots, o_{jn_j})$ die Menge aller Operationen des Auftrags j , wobei n_j die Anzahl der Operationen innerhalb des Auftrags j repräsentiert,
- $M := (\mu_1, \dots, \mu_m)$ Menge von m Maschinen (machines, processors, auch Arbeitsplätze), M_{ji} Menge der zulässigen Maschinen für o_{ji} (in der Literatur wird teilweise k anstatt m verwendet),
- $\mathcal{R} := (R_1, \dots, R_s)$ Menge von s zusätzlichen Ressourcen (additional resources), \mathcal{R}_{ji} Menge der zulässigen Ressourcenkombinationen für o_{ji} ,
- r_j, r_{ji} frühester Startzeitpunkt (release time, arrival time, ready time, auch Freigabezeitpunkt) des Auftrags j bzw. o_{ji} ,
- d_j, d_{ji} spätester Fertigstellungszeitpunkt (due date, auch Fälligkeitstermin, Liefertermin, Sollende) von Auftrag j bzw. o_{ji} ,
- p_{ji} Bearbeitungsdauer (processing time, auch Bearbeitungszeit, Sollzeit) der Operation o_{ji} ,
- w_j Gewicht (weight, auch Priorität) des Auftrags j .

Zeitpunkte und Dauer sind ganzzahlig. In der Regel hat ein Auftrag maximal eine Operation pro Maschine. Der Planvorschlag zu einem Shop Scheduling Problem ist „schedule“, bzw. Terminplan [EvKe00, S.221]. In dieser Arbeit wird er Fertigungsplan genannt, weil er zusätzlich die ausgewählten Ressourcen für die Operationen enthält. Den Operationen werden die folgenden Zeitpunkte, Maschinen und Ressourcen zugewiesen:

- S_j, S_{ji} geplanter Startzeitpunkt (starting time) des Auftrags j bzw. der Operation o_{ji} ,
- C_j, C_{ji} geplanter Endzeitpunkt (completion time, finish time, end time) des Auftrags j bzw. der Operation o_{ji} ,
- μ_{ji} geplante Maschine,
- R_{ji} geplante Ressourcenkombinationen.

Abbildung 2.11 zeigt ein Beispiel eines Fertigungsplans im typischen Gantt-Diagramm. Für die Operation o_{32} des Auftrags 3 (j_3) sind die oberen zwei Termine zusammen mit p_{ji} zusätzlich auf der Zeitachse eingezeichnet. t_o bezeichnet den aktuellen Zeitpunkt. In theoretischen Problemen wird oft $t_o = 0$ angesetzt.

Ein Fertigungsplan nennt sich zulässig (feasible), wenn die Reihenfolgenbeziehungen zwischen den Operationen eines Auftrags eingehalten und zusätzlich die Kapazitätvorgaben der Ressourcen nicht überschritten sind. Ein optimaler Fertigungsplan (optimal schedule) ist ein zulässiger Fertigungsplan, der die Zielvorgabe optimiert, d. h. die Zielfunktion minimiert bzw. maximiert [Fröh06, S.193].

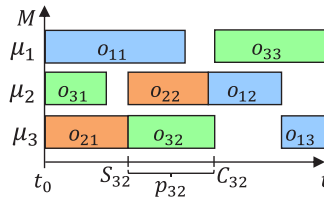


Abbildung 2.11: Beispiel eines Fertigungsplans

Ein zulässiger Fertigungsplan ist semiaktiv, wenn keine Operation nach links entlang der Zeitachse verschoben werden kann (lokale Linksverschiebung, local left-shift), ohne dabei die Reihenfolge der Operationen auf den Maschinen zu ändern oder die Zulässigkeit des Fertigungsplans zu verletzen [KäTe01].

Aktiv (active) dagegen ist ein zulässiger Fertigungsplan, wenn keine Operation mittels Linksverschiebungen auch mit Veränderung der Operationsreihenfolge auf den Maschinen (globale Linksverschiebung) früher starten kann, ohne dass weitere Operationen später enden [Fröh06, S.194]. Somit ist die Menge der aktiven Fertigungspläne eine Untermenge der semiaktiven.

Bei verzögerungsfreien (non-delay) Fertigungsplänen darf keine Maschine zu keinem Zeitpunkt ungeplant stehen, wenn zu diesem Zeitpunkt eine Operation startbar ist. Nach dieser Definition folgt, dass die verzögerungsfreien Fertigungspläne eine Untermenge der aktiven sind [Fren82, S.157].

Abbildung 2.12 zeigt die Arten von Fertigungsplänen und ihre Relation zueinander. Obwohl die verzögerungsfreien Fertigungspläne die kleinste Menge bilden, existiert nicht immer ein optimaler verzögerungsfreier Fertigungsplan, wie ein Beispiel aus [Fren82, S.157] zeigt. Auch aktive oder i. A. zulässige Fertigungspläne können optimal bzgl. einer Zielvorgabe sein.

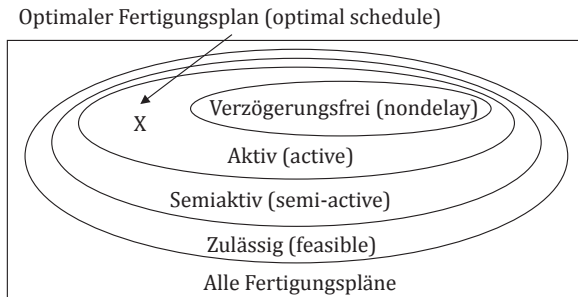


Abbildung 2.12: Arten von Fertigungsplänen nach [Pine14, S.25]

2.2.2 Arten von Shop Scheduling Problemen

Das Shop Scheduling Problem wird meistens nach der Art der Reihenfolgenbeziehungen zwischen Operationen eines Auftrags

und der Art der Berechnung der Bearbeitungsdauer von Operationen aufgeteilt.

In der Fertigungsfeinplanung ist die Bearbeitungsdauer (die Sollzeit) der Operationen von vornherein bekannt (deterministisch), da sie im allgemeinen Fall zuvor schon systematisch abgeleitet wurde [Refa76, S.62]. Aus diesem Grund wird die Feinplanung der Fertigung am häufigsten mittels deterministischer Modelle [BEPT04] abgebildet.

In der Realität sind Schwankungen im tatsächlichen Fertigungsablauf auch bei vollautomatisierter Fertigung nicht ausgeschlossen, weshalb stochastische Modelle für die Forschung zunehmend interessanter werden [CuFe09, S.60, Pine14]. Die stochastischen Modelle benötigen zusätzlich die Art und die Parameter der Verteilung für die Bearbeitungsdauer der Operationen. Produzierende Betriebe ermitteln diese Information im allgemeinen Fall nicht.

Des Weiteren unterscheiden sich die Shop Scheduling Probleme auch darin, in welcher Reihenfolge die Operationen eines Auftrags abgearbeitet werden dürfen. Das Open-Shop Problem (OSP) als allgemeinste Klasse definiert keine Restriktionen in der Bearbeitungsreihenfolge [Gonz04]. Mittels OSP werden beispielsweise Reparaturzentren abgebildet, bei denen die Reihenfolge der durchzuführenden Reparaturen an einem Gegenstand keine Rolle spielt.

Erheblich mehr Fertigungsorganisationen entsprechen der Klasse der Flow Shop Probleme (FSP). Bei FSP ist die Reihenfolge der Maschinen festgelegt und die Reihenfolge der Aufträge auf jeder Maschine identisch [GoSa78]. Ein typisches Beispiel hierfür ist die Linienfertigung, bei der alle Werkstücke alle Stationen (Maschinen) einer Fertigungslinie durchlaufen.

Im Gegensatz zur Linienfertigung sind die Maschinen in einer klassischen Werkstattfertigung voneinander entkoppelt. Die Klasse Job Shop Problem (JSP) beschreibt diese am häufigsten vorkommende Fertigungsorganisation, in der jeder Auftrag einen eigenen Weg durch die Maschinen nehmen darf. Nur die Reihenfolge der Operationen innerhalb des Auftrags muss eingehalten werden [Fren82, S.5ff].

JSP ist nicht nur sehr praxisnah, sondern auch gleichzeitig die anspruchsvollste aus allen drei Klassen, weshalb JSP auch am häufigsten Gegenstand der Forschung ist. FSP wird dagegen als ein Spezialfall von JSP gesehen, da FSP zusätzliche Einschränkungen gegenüber JSP hat. FSP und JSP wurden schon in den 1960er Jahren definiert, während die ersten Versuche einer rechnergestützten Fertigungssteuerung liefen.

Bei den Versuchen das Problem der Fertigungssteuerung realitätsnäher abzubilden, wurden später weitere Klassen neben diesen drei Hauptklassen definiert. Für die Abbildung von parallelen Fertigungslinien in der Linienfertigung und sich ersetzenden Maschinen in der Werkstattfertigung wurden in den 1970ern die generalisierten Klassen Flexible Flow Shop (auch Hybrid Flow Shop) bzw. Flexible Job Shop Problem definiert [FaM]07, RuVá10].

Die unterschiedlichen Fertigungsarten besitzen neben den Reihenfolgenabhängigkeiten zwischen Operationen und Maschinen noch weitere spezifische Eigenschaften. Aufgrund der Notwendigkeit für eine realitätsnahe Planung innerhalb des MRP II Konzepts entstand eine Reihe von mathematischen Problemen wie JSP mit unterbrechbaren Operationen (Preemptive JSP) oder FSP ohne erlaubte Wartezeiten zwischen aufeinanderfolgenden Operationen eines Auftrags (No-Wait FSP) [PaBa98, SaCh79]. Um die Klassifizierung der unterschiedlichen Scheduling Probleme zu unifizieren, hat [GLLK79] die bis dato verwendete $\alpha|\beta|\gamma$ Notation vorgeschlagen. Dabei steht α für die Art des Scheduling Problems

(F für FSP, J für JSP etc.) sowie die Anzahl der Maschinen (optional), β (optional) für die Auftrags- und Operationsspezifika (Unterbrechung, verzögerungsfrei, reihenfolgenabhängige Rüstzeiten etc.) und γ für die Zielfunktion.

Zu den schon etablierten Hauptklassen wurden weitere Klassen wie das Cycle Shop Problem (CSP) und das Mixed Shop Problem (MSP) definiert, die wiederholbare Operationen bei FSP erlauben bzw. gleichzeitige Operationen im Auftrag mit fester und beliebiger Reihenfolge einführen [MaIn85, Timk04]. Ein Versuch wurde unternommen, OSP, JSP und MSP durch die Definition des Group Shop Problems zu generalisieren [SBMR02].

Die beschriebene Vielfalt an Problemklassen zeigt, wie schwierig es ist, die Feinplanung innerhalb der Fertigungssteuerung einheitlich zu definieren. Für einen Großteil der Shop Scheduling Probleme ist die Darstellung als disjunktiver Graph möglich.

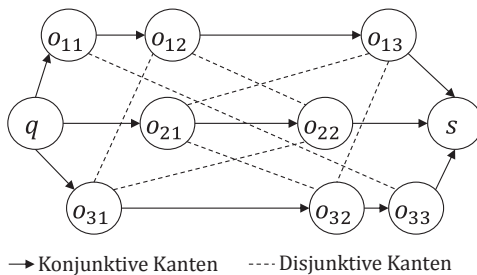


Abbildung 2.13: Der disjunktive Graph des Fertigungsproblems aus Abbildung 2.11

Der disjunktive Graph $G = (V, E) = (O \cup \{q, s\}, \mathbb{C} \cup \mathbb{D})$ hat alle Operationen als Knoten, sowie die zwei speziellen Knoten Quelle q und Senke s [Bala69, CPPR04]. Einen disjunktiven Graph mit dem Beispiel aus Unterkapitel 2.2.1 zeigt Abbildung 2.13. Konjunktive und disjunktive Kanten (\mathbb{C} und \mathbb{D}) verbinden die Knoten miteinander. Die konjunktiven Kanten \mathbb{C} sind gerichtet und stellen die Reihenfolgenbeziehungen zwischen je zwei Operationen (z. B.

innerhalb des Auftrags oder auftragsübergreifend) dar. Die disjunktiven Kanten \mathbb{D} sind ungerichtet und zeigen, dass die verbundenen Operationen nicht gleichzeitig durchlaufen werden dürfen. Disjunktive Kanten werden zwischen Operationen gezogen, wenn sie demselben Auftrag angehören, aber deren Reihenfolge nicht durch Reihenfolgenbeziehungen eindeutig festgelegt ist oder wenn sie dieselbe Maschine oder dieselbe zusätzliche Ressource exklusiv für die Abarbeitung benötigen. Konjunktive Kanten verbinden zusätzlich q mit allen Operationen ohne Vorgänger sowie alle Operationen ohne Nachfolger mit s .

Der disjunktive Graph ist eine beliebte Basisstruktur für viele Feinplanungsverfahren, da er alle Beziehungen zwischen Operationen enthält. Nicht nur jedes Feinplanungsproblem sondern auch jeder aktive Fertigungsplan lässt sich abbilden, indem die disjunktiven Kanten im Graph azyklisch gerichtet werden. Die Richtung zeigt die Abarbeitungsreihenfolge.

Obwohl die Modellierung mittels disjunktiven Graphen in ihrer ursprünglichen Form für die allgemeinen JSP, OSP und MSP geeignet ist, existieren zahlreiche Performanzverbesserungen und Erweiterungen auch für spezifische Problemstellungen [WhRo90, BIPS00]. Für viele Shop Scheduling Probleme ist der disjunktive Graph jedoch nicht ausreichend und weitere Datenstrukturen wie Intervallbäume (interval tree) oder Intervall-Gebirge sind für deren Beschreibung notwendig [BKOS00, S.212–217, HBBW01, S.130–170].

2.2.3 Optimierungsziele

Nicht nur wird jede Fertigung unterschiedlich modelliert, sondern i. d. R. werden auch unterschiedliche, spezifische Zielvorgaben verfolgt. In der Forschung wurde meist die Minimierung folgender Zielfunktionen betrachtet [Fren82, S.12–13, Bruc07, S.7, Pine14, S.18–19]:

- Maximaler Fertigstellungszeitpunkt C_{max} (makespan),
 $C_{max} := \max\{C_1, \dots, C_n\}$,
- Maximale Terminabweichung L_{max} (maximum lateness),
 $L_{max} := \max\{L_j | L_j := d_j - C_j, j \in J\}$, L_j ist die Terminabweichung von Auftrag j ,
- Summe der Fertigstellungszeitpunkte $\sum C_j$ (total completion time) und deren gewichtete Variante $\sum w_j C_j$ (total weighted completion time)
- Gewichtete Summe der Aufenthaltsdauer $\sum w_j F_j$ (total flow time), wobei $F_j := C_j - r_j$ die Aufenthaltsdauer des Auftrags j ist, (Aufgrund der konstanten $w_j r_j$ ist $\sum w_j F_j$ äquivalent zu $\sum w_j C_j$.)
- Summe der Verspätungen $\sum T_j$ (total tardiness) und mittlere Verspätung $1/n \sum T_j$ (mean tardiness) als menschenfreundlichere Variante, $T_j := \max\{L_j, 0\}$,
- Gewichtete Summe der Verspätungen $\sum w_j T_j$ (total weighted tardiness),
- Gewichtete Anzahl von Verspätungen $\sum w_j U_j$ (weighted number of late jobs), $U_j \in \{0,1\}$, wobei $U_j = 1$ falls $T_j > 0$ und $U_j = 0$ sonst.

Aus Forschungssicht sind die oben erwähnten Zielfunktionen interessant und auch heute noch Untersuchungsgegenstand. In der Praxis passt C_{max} als Zielfunktion beispielsweise zum Schiffsbau, um die Fertigstellung des gesamten Schiffs zu optimieren. Die Zielfunktionen sind aber im Allgemeinfall für die Quantifizierung der Terminabweichung des Fertigungsplans in der diskreten Fertigung unzureichend. Denn diese Zielfunktionen können die einzelnen C_j der verspäteten Aufträge nicht steuern. Unternehmensziel sind wenige verspätete Aufträge und falls es solche gibt, sollen die Verspätungen nicht sehr stark variieren. Praxisnahe Hilfe hierfür sollen die folgenden zusätzlichen Zielfunktionen liefern [Haupt89]:

- Varianz der Verspätung σ_T^2 (variance of tardiness),
- Standardabweichung σ_T der Verspätung (standard deviation of tardiness).

Variationen dieser Zielfunktionen mit einer überproportionalen Gewichtungsfunktion $f(T_j)$ anstelle von T_j wurden ebenfalls untersucht. Diese unvollständige Einreihung von Zielfunktionen zeigt die Schwierigkeit, eine universelle Zielvorgabe über die Termineinhaltung mathematisch zu formulieren.

Mit der Einführung von CIM (vgl. 2.1) gewannen neben der Verbesserung der Termintreue auch die Optimierung des Fertigungsflusses und der reihenfolgenabhängigen Rüstzeiten an Bedeutung. Für deren Berechnung spielen auch die folgenden Parameter eine Rolle:

- $E_j := \max\{0, C_j - d_j\}$ Verfrühung eines Auftrags j (earliness),
- st_{ijk} die anfallende reihenfolgenabhängige Rüstzeit (in dieser Arbeit einfach als Rüstzeit bezeichnet) bei einem Übergang zwischen Operationen der Aufträge j und k auf μ_i (sequence dependent setup time). Asymmetrische Rüstzeiten ($st_{ijk} \neq st_{ikj}$) sind nicht unüblich und treten beispielsweise in Lackieranlagen bei Übergängen zu einer helleren Lackfarbe oder in der Werkstattfertigung bei Verwendung eines zusätzlichen Werkzeugs auf,
- sc_{ijk} die anfallenden reihenfolgenabhängigen Rüstkosten (sequence dependent setup cost) anstelle von st_{ijk} , falls die Rüstkosten nicht linear zu st_{kji} steigen und manche Rüstvorgänge kostenintensiver als andere sind.

Die Rüstzeiten (st_{ijk}) können auch bei Optimierung der Terminabweichung teilweise implizit mitreduziert werden, falls dies zur

Minimierung der Zielfunktion beiträgt. Jedoch können sie selber ein Optimierungsziel sein [ANCK08, Pine14, S.70, 464]:

- Summe der Rüstzeiten TST (total setup time), $TST := \sum_{j,i}(YC_{ji} - YS_{ji})$ mit YS_{ji} und YC_{ji} als geplanter Startzeitpunkt bzw. Endzeitpunkt des Rüstvorgangs von o_{ji} ,
- Summe der Rüstkosten TSC (total setup cost), $TSC := \sum_{j,i} YR_{ji}$, YR_{ji} die geplanten Rüstkosten für o_{ji} ,
- Summe der Verfrühungen $\sum E_j$ (total earliness).

In den oberen Zielfunktionen spielt die Verspätung entweder keine Rolle oder ist eine harte Restriktion ($\sum T_j = 0$). Es existiert eine vergleichsweise kleine Anzahl an Zielfunktionen für die Rüstzeiten und die Verfrühungen, weil sie sich objektiv mittels fälliger Lagerungs- und Rüstaufwände quantifizieren lassen. Die Verspätung dagegen ist eher subjektiv und dadurch nur indirekt aus verschiedenen Blickwinkeln quantifizierbar.

Die Forschung untersucht normalerweise Zielfunktionen mit nur einer Parameterart. Ausnahme ist die Summe der Verfrühungen und Verspätungen $\sum E_j + \sum T_j$ (total earliness plus total tardiness) und die gewichtete Variante $\sum w'_j E_j + \sum w''_j T_j$ mit den unterschiedlichen Gewichtungen w'_j und w''_j für Verfrühungen bzw. Verspätungen. Diese Zielfunktionen sind ein Versuch zwei Arten von Optimierungskriterien zu kombinieren.

In der Praxis sind multikriterielle Zielsetzungen in der Planung (multi objective scheduling) zunehmend erwünscht und gefordert. Die Feinplanung kann abhängig von den Zielkriterien auch als lineare oder nichtlineare multikriterielle Optimierungsaufgabe gesehen werden. Das Problem der multikriteriellen Optimierung ist seit langem bekannt und sehr breit untersucht [Miet99], weil es bei vielen verschiedenartigen Optimierungsaufgaben auftritt.

Im Allgemeinfall soll die Menge der Zielfunktionen $\{f_1(\mathcal{X}), f_2(\mathcal{X}), \dots, f_k(\mathcal{X}) | \mathcal{X} \in S\}$ für $k \geq 2$ minimiert werden, wobei S die Menge aller zulässigen Fertigungspläne ist. Für den Fertigungsplan \mathcal{X} wird der Entscheidungsvektor (decision vector) $z := f(\mathcal{X}) := (f_1(\mathcal{X}), f_2(\mathcal{X}), \dots, f_k(\mathcal{X}))^T = (z_1, z_2, \dots, z_k)^T$ Zielvektor (objective/criteria vector) und die einzelnen $z_i := f_i(\mathcal{X})$, $i = 1, \dots, k$, Kriteriumswerte (objective/criterion value) genannt [Miet99, S.5–6]. Die gleichzeitige Minimierung aller $f_i(\mathcal{X})$ ist selten möglich, weshalb oft Kompromisslösungen unvermeidbar sind. Für die Auswahl eines Fertigungsplans anhand des ermittelten Zielvektors werden im Folgenden die meistverwendeten Methoden in der Feinplanung erläutert.

Die Gewichtung der einzelnen z_i des Zielvektors $\sum w_i z_i$ (weighted aggregation) für die Erstellung eines einzelnen skalaren Planwerts (total schedule cost) ist sicherlich die am stärksten verbreitete und verwendete Methode in der Feinplanung [NaHH95], da sie sehr einfach zu realisieren ist und die Entscheidung wieder auf eine einzelne Dimension reduziert wird. Aus diesem Grund ist diese Methode breit untersucht und ihre Nachteile bekannt. Die größte Schwierigkeit ist die Bestimmung der einzelnen Gewichtungen w_i und die Lösungsauswahl beschränkt sich auf wenige Bereiche des Lösungsraums $Z := f(S)$ [DaDe97].

Der zweite sehr häufig verwendete Ansatz besteht in der Generierung mehrerer pareto-optimaler Fertigungspläne [KaHB02]. Ein zulässiger Fertigungsplan ist pareto-optimal, wenn kein anderer zulässiger Fertigungsplan existiert, der ein Zielkriterium verbessert ohne ein anderes zu verschlechtern [Miet99, S.10ff, DaDe97].

Die pareto-optimalen Kriterienvektoren bilden die sog. Pareto-Front. Abbildung 2.14 zeigt ein Beispiel einer zweidimensionalen Pareto-Front und einiger Lösungen für die Zielfunktionen $\sum T_j$ und TST . Zwei- und dreidimensionale Pareto-Fronten lassen sich

grafisch darstellen, während für mehrdimensionale nur die Tabellenform geeignet ist. Die Erstellung von pareto-optimalen Fertigungsplänen ist dem Feinplanungsverfahren überlassen. Aus ihren Kriterienvektoren wird die Pareto-Front approximiert und für die Auswahl des besten Fertigungsplans meist einem Entscheider, oft dem Planer, überlassen.

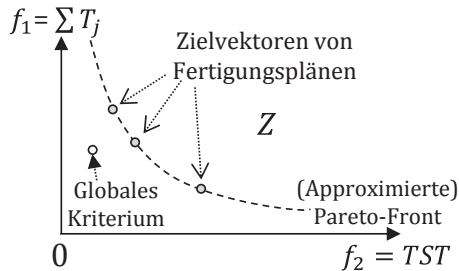


Abbildung 2.14: Beispiel einer Pareto-Front

Eine weitere, weniger verbreitete Methode ist das Definieren eines Globalen Kriteriums (global criterion). Mithilfe einer Distanzfunktion (Metrik) wird der Abstand des Zielvektors eines Fertigungsplans zum Globalen Kriterium gemessen. Der Fertigungsplan mit dem niedrigsten resultierenden skalaren Wert wird ausgewählt. Hier wie auch bei der gewichteten Summe der Kriteriumswerte ist es schwierig, das Globale Kriterium zu bestimmen. Als Metriken kommen u. a. der Euklidische Abstand und die Tchebyscheff Metrik zur Anwendung [Miet99, S.67ff, GuSi02, LoWH05].

Multikriterielle Scheduling Probleme sind in der Forschung aufgrund ihrer Komplexität wenig untersucht und die Untersuchungen konzentrieren sich oft auf Ein- und Zwei-Maschinen-Probleme [NaHH95, Hoog05, WZGL08].

2.2.4 Optimalität

Shop Scheduling ist aus theoretischer Sicht auch bei sehr kleinen Problemgrößen und einfachen Zielsetzungen ein sehr schwieriges Optimierungsproblem. Der Grund dafür ist die extreme kombinatorische Explosion.

Die maximale Anzahl möglicher semi-aktiver Fertigungspläne für ein Job/Open Shop Scheduling Problem ist $(n!)^m$ [Rinn76, S.16–17]. Sie ergibt sich aus der Anzahl möglicher Reihenfolgen der Operationen $n!$ auf einer einzelnen Maschine und dem Exponenten m für die möglichen Kombinationen über alle m Maschinen. Ein Rechenbeispiel zeigt für das (aus praktischer Sicht) sehr kleine Scheduling Problem $n = m = 5$ eine maximal mögliche Anzahl semiaktiver Fertigungspläne von 24.883.200.000. Ein naives Verfahren müsste alle Kombinationen durchprobieren, um einen optimalen Fertigungsplan zu finden, was zu einer hyperexponentiellen Steigerung der Rechenschritte in Beziehung mit der Problemgröße führt [Rinn76, S.31–32].

Entscheidend für die theoretische Existenz eines effizienten Verfahrens ist, welcher Komplexitätsklasse das Optimierungsproblem (oder genauer gesagt das daraus durch Modifikation der Aufgabenstellung abgeleitete Entscheidungsproblem) angehört. Auf Abbildung 2.15 sind die unterschiedliche Mengen P , NP , NP -vollständig (NP -complete) sowie NP -schwer (NP -hard) samt deren Beziehungen nach heutigem Kenntnisstand dargestellt, vorausgesetzt $P \neq NP$ [Gura89, Kap.5.3, Leun04b, Kap.2.4]. Zu jeder Menge sind einige Varianten des Scheduling-Problems in der $\alpha|\beta|\gamma$ Notation (s. 2.2.1) nach bisherigem Forschungsstand angegeben, um die Grenze zwischen den Klassen P und NP innerhalb des Shop Scheduling Problems zu verdeutlichen [LeKB77, GaJS76, GoSa78, ANCK08, BrKn09, Pine14, S.603ff]. Diese Grenze ist noch heute nicht eindeutig festgelegt und bestimmte Probleme können immer noch P oder NP zugeordnet werden.

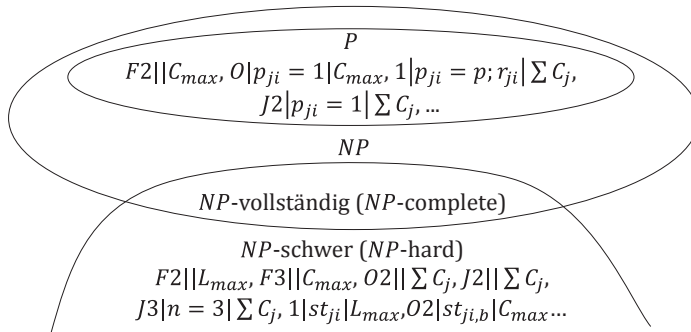


Abbildung 2.15: Komplexitätsklassendiagramm

In der Komplexitätsklasse *P* sind Zwei-Maschinen- und Ein-Maschinen-Probleme oder solche mit einheitlicher Bearbeitungsdauer (gekennzeichnet durch $p_{ji} = 1$ in β) enthalten. Auch weitere, jedoch weniger für die Fertigung relevante Varianten des Scheduling-Problems liegen in *P*. Für diese Probleme ist nachweislich ein polynomiales Verfahren konstruierbar.

Das allgemeine JSP dagegen ist schon bei $n = m = 3$ mit $\gamma = C_{max}$ oder $\gamma = \sum C_j$ *NP-schwer* [SoSh95]. Bekannterweise ist auch $J2||\sum C_j$ *NP-schwer*. Nebenbedingungen wie Rüstzeiten (st_{ji}) erschweren das Problem zusätzlich. $1|s_{ji}|C_{max}$ ist z. B. äquivalent mit dem Problem des Handlungsreisenden (Travelling Salesman Problem, TSP). Viele der *NP-schweren* Scheduling Probleme sind *NP-vollständig* [Ullm75]. Heute ist bekannt, dass Shop Scheduling Probleme je nach Art ab einer bestimmten Problemgröße *NP-schwer* und teilweise sogar stark *NP-schwer* (*NP-hard* in the strong sense) sind und dadurch keinen Vorteil aus der Ganzzahligkeit der Eingangsdaten ziehen.

Ein effizientes optimales Verfahren für Shop Scheduling Probleme ist aus den oben genannten Gründen nicht realisierbar. Es ist jedoch von Bedeutung, bestimmte Garantien bezüglich der Lösungsgüte eines suboptimalen Verfahrens zu bekommen. Wenn

das erreichbare Minimum der Zielfunktion f für ein bestimmtes Shop Scheduling Problem (das Optimum, OPT_f) bekannt wäre, wäre auch die Abweichung eines Fertigungsplans davon berechenbar. Das Optimum ist jedoch meist nicht bekannt.

Abhilfe schafft die Ermittlung einer unteren Schranke (lower bound, LB_f) der Zielfunktion für ein konkretes Problem. Es gilt $LB_f \leq OPT_f$ und LB_f ist in der Regel leichter als OPT_f zu berechnen und dient der Schätzung des Optimierungspotentials. Funktionen, die größere LB_f liefern, werden bevorzugt, da dadurch die Entfernung eines Fertigungsplans vom Optimum kleiner ist.

Je nach Zielfunktionen gibt es unterschiedliche Ansätze für die Ermittlung der unteren Schranke. Falls beim Weglassen von Restriktionen und Nebenbedingungen das Optimum oder die untere Schranke des neuen Problems leichter zu bestimmen sind, werden diese für das ursprüngliche Problem genommen. Das ursprüngliche Problem kann auch auf mehrere Teilprobleme, z. B. auf Ein-Maschinen-Probleme, aufgeteilt werden. Aus den optimalen Lösungen der Teilprobleme wird dann die untere Schranke ermittelt.

Verfahren für die Ermittlung der unteren Schranke sind für die gängigen Zielfunktionen wie C_{max} , L_{max} , $\sum T_j$ entwickelt [Rinn76, S.109–119, BDMN99, BrKn03, ANCK08, DrEK09]. Eine genauere untere Schranke ist für viele Feinplanungsverfahren von besonderer Bedeutung, wie im nächsten Unterkapitel gezeigt wird.

2.3 Verfahren für Shop Scheduling Probleme

Shop Scheduling Probleme beschäftigen mehrere Disziplinen seit ca. 70 Jahren. Da die Grenze zwischen den Komplexitätsklassen P und NP oft im Bereich der einstelligen Problemgröße liegt, ent-

standen viele meist heuristische Verfahren. Mit den vorgeschlagenen Ansätzen verfolgen die involvierten Disziplinen oft unterschiedliche Ziele. Aus Sicht eines Mathematikers sind die theoretische Analyse der Problemvarianten, ihre Komplexitätsklasse und Optimalität von Interesse. Ein Fertigungssteuerer braucht einfache, nachvollziehbare, einsetzbare praktische Verfahren zur Steuerung des täglichen Fertigungsablaufs. Ein Wirtschaftswissenschaftler strebt Kostenminimierung und Terminverlässlichkeit an. Ein Informatiker versucht mit den zurzeit verfügbaren Rechenkapazitäten möglichst effiziente und einsetzbare Verfahren zu entwickeln und verwendet dabei Fortschritte aus der künstlichen Intelligenz, Operations Research, Mathematik sowie anderen Disziplinen.

Nachfolgend wird ein Überblick über die unterschiedlichen Grundtypen von Verfahren gegeben, organisiert nach ihrer Art der Lösungsfindung. Anschließend folgt ein Überblick über unterschiedliche hybride Verfahren, die Zusammensetzungen aus mehreren Grundtypen sind. Im letzten Unterkapitel finden sich unifizierte Testdaten für die Vergleichbarkeit der Verfahren.

2.3.1 Überblick der Verfahren

Feinplanungsverfahren unterscheiden sich grundsätzlich in ihrer Herangehensweise an das Feinplanungsproblem. Konstruktionsverfahren versuchen ausgehend von einer Problemstellung einen Fertigungsplan zu erstellen, während Verbesserungsverfahren iterativ einen bestehenden Fertigungsplan optimieren. Des Weiteren unterscheiden sich die Verfahren in der Wahl des Lösungswegs und der Anzahl der untersuchten Fertigungspläne. Eine schematische Einordnung der Verfahren [Bruc07, Fren82, Lödd04, BuKe13, Pine14] ist in Abbildung 2.16 zu finden.

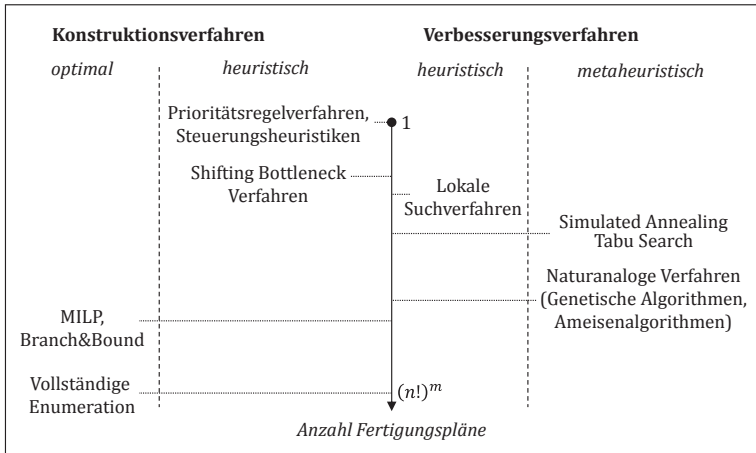


Abbildung 2.16: Verfahren für Shop Scheduling Probleme

Exakte Konstruktionsverfahren untersuchen den Lösungsraum und finden die optimale Lösung für eine bestimmte Zielfunktion, jedoch auf Kosten der Laufzeit. Durch Heuristiken lässt sich die Lösungssuche bei diesen Verfahren auf Kosten der Lösungsqualität einschränken, um die Performanz zu steigern.

Die Verbesserungsverfahren führen heuristische iterative Veränderungen in einen bestehenden Fertigungsplan ein und hinterlassen in der Regel nach jeder Iteration einen zulässigen Fertigungsplan. Aus diesem Grund sind sie immer unterbrechbar und eignen sich für die permanente Ausführung. Ein Großteil der Verbesserungsverfahren nutzt bekannte allgemeine Schemata zur Lösung von unterschiedlichen kombinatorischen Problemen (Metaheuristiken) und wird oft als metaheuristisches Feinplanungsverfahren bezeichnet [BIRo03].

In den nachfolgenden Unterkapiteln werden verbreitete und untersuchte Verfahren ausführlich beschrieben. Andere bekannte Verfahren werden lediglich erwähnt.

2.3.2 Optimale Konstruktionsverfahren (Exakte Verfahren)

Die einfachste Form der Findung einer optimalen Lösung ist die Untersuchung aller $(n!)^m$ Fertigungspläne zu einem konkreten Problem mithilfe einfacher Algorithmen [PaWi79, S.131ff]. Auch wenn die Iteration über alle aktiven Fertigungspläne eingeschränkt wird [Rinn76, S.32–34], ist sie selbst bei sehr kleinen Problemen praktisch nicht durchführbar (vgl. 2.2.4). Deshalb werden oft Methoden der gemischtganzzahligen (nicht)linearen Programmierung (mixed integer (non)linear programming, MILP) angewendet [PoWo06].

Die Erstellung und das Durchprobieren aller Kombinationen ist mittels des Branch&Bound Verfahrens (Verzweigung und Schranke) deutlich einfacher und performanter [CaPi89, DeHe92, Bruc07, S.202–221, Pine14, S.182–193]. An dieser Stelle wird sehr häufig die in Unterkapitel 2.2.2 beschriebene Formulierung des Scheduling Problems als disjunktiver Graph angewendet. Das Branch&Bound Verfahren führt eine Tiefensuche und/oder Breitensuche auf den disjunktiven Graphen durch und erstellt so (Teil-) Fertigungspläne. Welcher Zweig in der Suche zuerst untersucht oder überhaupt bis zur Senke verfolgt wird, hängt von den ermittelten oberen und unteren Schranken für die möglichen Verzweigungen innerhalb des Zweigs ab.

Die obere Schranke ist der Zielfunktionswert des aktuell besten Fertigungsplans. Die untere Schranke bezieht sich auf einen Zweig und zeigt das geschätzte Potential der weiteren Verzweigungen. Die weiteren Verzweigungen können als ein Unterproblem des gesamten Scheduling Problems gesehen werden, zu dem die obere Schranke mittels der in Unterkapitel 2.2.4 beschriebenen Verfahren ermittelt werden kann. Das Branch&Bound Verfahren verfolgt Zweige mit niedriger unterer Schranke mit Vorrang. Falls die

untere Schranke eines Zweigs über der oberen Schranke liegt, wird der Zweig nicht untersucht.

Branch&Bound Verfahren sind sehr verbreitet, da sie den Lösungsraum effizient erforschen. Durch geschickte Implementierung einer Kombination aus Tiefen- und Breitensuche ist nach kurzer Zeit ein erster Fertigungsplan erstellt. Somit ist ein Abbruch des Verfahrens mit abschließender Rückgabe des aktuell besten Fertigungsplans kurze Zeit nach dem Verfahrensaufwurf möglich. Ein weiterer Vorteil ist die leichte Verteilung auf mehrere Prozessoren und/oder Rechner.

Auch rein mathematische Optimierungsverfahren können auf das Feinplanungsproblem angewendet werden. Das Feinplanungsproblem ist als eine Menge von Ungleichungen repräsentierbar [Fren82, S.130–136, Rinn76, S.37ff] und bei Bedarf auch in ein Programm von Gleichungen umwandelbar. Mathematische Verfahren wie Simplex-Variationen können anschließend dieses Programm von Ungleichungen lösen [SrGh86, GoBN05, DeKü13]. Diese Verfahren können nicht nur selbstständig sondern auch innerhalb des Branch&Bound Verfahrens zur Berechnung der unteren Schranke angewendet werden. Je mehr Restriktionen (vgl. 2.3.2) ein konkretes Feinplanungsproblem enthält, auf desto mehr (Un-)Gleichungen wird das Problem abgebildet. Der Lösungsraum ist oft nicht konvex und dies erschwert die Anwendung der mathematischen Verfahren.

2.3.3 Heuristische Konstruktionsverfahren

Ab einer gewissen Problemgröße ist die Untersuchung des Lösungsraums nicht in vertretbarer Zeit möglich. Die Erstellung eines ausführbaren (zulässigen) Fertigungsplans ist jedoch gefordert. Dafür werden sogenannte Eröffnungsverfahren verwendet. Die Eröffnungsverfahren sind einfache heuristische Verfahren, welche in polynomialer Zeit einen Fertigungsplan mithilfe einer

Art Fertigungssimulation konstruieren [Rinn76, S.51–52, Fren82, S.155ff, Pine14, S.376–382].

Typische Eröffnungsverfahren pflegen für jede Maschine eine Warteschlange von ausführbaren Operationen. Bei Initialisierung der Verfahren werden alle Operationen ohne Vorgänger in die Warteschlangen eingefügt. Mithilfe einer Auswahlfunktion wird eine Operation aus einer zuvor ausgewählten Warteschlange ausgewählt und eingeplant. Die Nachfolger der Operation werden anschließend in die entsprechenden Warteschlangen eingefügt. Das Einplanen von Operationen wird solange ausgeführt bis alle Warteschlangen leer sind.

Im oben beschriebenen Ablauf sind die Auswahlfunktionen für die Warteschlangen und die Operationen zu definieren. Für die Auswahl der Warteschlange in einer Iteration wird der Giffler-Thompson Algorithmus (GTA) sehr häufig angewendet, weil er nachweislich aktive Fertigungspläne erzeugt [GiTh60] und in $O(n^2 * m)$ abläuft.

In Abbildung 2.17 ist der beispielhafte Ablauf des GTA bei der Erstellung des Fertigungsplans aus Abbildung 2.11 dargestellt. Der Algorithmus berechnet aus dem teilweise konstruierten Fertigungsplan die möglichen Endzeitpunkte c_{ji} aller Operationen aus allen Warteschlagen. Die Warteschlangen enthalten alle Operationen mit bereits eingeplanten Vorgängern oder ohne Vorgänger. Schließlich wählt der Algorithmus die Warteschlange μ_2 mit der Operation mit frühestem Endzeitpunkt (1). Dieser Zeitpunkt ist die Betrachtungsgrenze für die Auswahl der nächsten Operation aus der Warteschlange. Nur aus Operationen in der Warteschlange von μ_2 mit einem möglichen Startzeitpunkt vor der Betrachtungsgrenze wird eine zur Einplanung ausgewählt (2). Nach deren Einplanung werden alle ihre Nachfolger in die entsprechenden Warteschlangen eingefügt. Diese Iteration wird solange wiederholt, bis alle Warteschlangen leer sind.

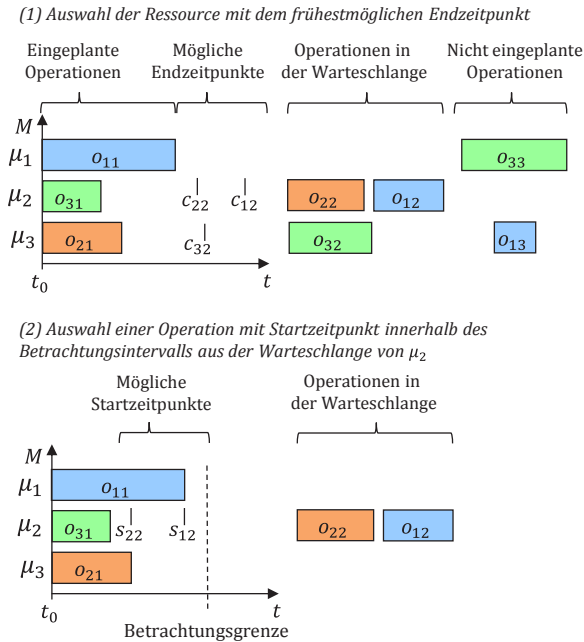


Abbildung 2.17: Giffler-Thompson Algorithmus

Für die Auswahl der nächsten Operation in der Warteschlange existieren viele Auswahlverfahren. Die sog. Greedy oder myopischen Heuristiken versuchen die Priorität einer Operation nur unter Betrachtung der lokalen Situation zu ermitteln.

Beispiele für solche Prioritätsregeln sind die kürzeste Bearbeitungsdauer der Operation (shorter process time, SPT), die längste Bearbeitungsdauer (longest process time, LPT), der früheste Fertigstellungszeitpunkt des Auftrags der Operation (earliest due date, EDD) oder Zufall (RANDOM). Etwas aufwändigere Prioritätsregeln berücksichtigen auch die verbleibende Bearbeitungsdauer des Auftrags der Operation (remaining time, RT), die Anzahl der Operationen in Folgewarteschlangen (work in next queue, WINQ) etc. Fortschrittliche Prioritätsregeln knüpfen die Auswahl der

Operation an Bewertungskriterien des Fertigungsplans. Die Forschung hat in den vergangenen 70 Jahren weit über hundert Prioritätsregeln [BIPH82, Haup89, Mose93] untersucht. In vielen Simulationen hat sich gezeigt, dass SPT sowie oft auch RANDOM trotz ihrer Einfachheit sehr gute Fertigungspläne erzeugen.

Mithilfe des GT Algorithmus lassen sich auch viele Steuerungsstrategien wie Kanban, Constant Work-In-Process (ConWIP), Belastungsorientierte Auftragsfreigabe (BOA) und viele weitere als Planungsverfahren implementieren [Lödd04].

Die Entscheidung über die nächste Operation in jeder Warteschlange (Maschine) findet lokal statt, weshalb diese Verfahren für dezentrale Steuerungen wie die Agentensteuerung eingesetzt werden können. Agenten repräsentieren die einzelnen Aufträge, die sich in die Warteschlangen der Agenten der Maschinen einreihen und mithilfe eines Verhandlungsverfahrens ihren Platz in der Ausführungsreihenfolge „erkaufen“ [Bake98, ShWH06]. Die dezentrale Steuerung kann in bestimmten Fällen zu einem sehr guten Fertigungsablauf führen, jedoch fehlt ein Fertigungsplan, der den übergeordneten Instanzen bestimmten Garantien über die Einhaltung der spätesten Fertigstellungstermine zusichert.

Die einfachen Prioritätsregeln berücksichtigen nur die anstehende Auslastung in der Warteschlange. Aus diesem Grund führt das Shifting-Bottleneck Verfahren (Engpassverfahren) für jede Maschine eine isolierte Planung nur mit den betroffenen Operationen durch. Dadurch wird die Engpassmaschine erkannt und zuerst geplant. Bei der Planung nachfolgender Maschinen ist die Umplanung bereits eingeplanter Maschinen erlaubt und manchmal notwendig. Die Umplanung kann mithilfe des Branch&Bound Verfahrens oder anderer Heuristiken ermittelt werden. Diese Heuristik ist komplexer als die einfachen Prioritätsregeln, weil sie mehrere Planungen durchführt und sich auf weitere Verfahren verlässt. Gleichzeitig ist ihre Laufzeit vorhersagbar und liefert

sehr gute Ergebnisse, weshalb viele Modifikationen vorgeschlagen wurden [AdBZ88, IvLa96, PiSi99].

Neben den einfachen Eröffnungsverfahren existieren weitere heuristische Konstruktionsverfahren, die mehrere Fertigungspläne mithilfe der Lagrange Relaxation [Thom96] oder mittels Kombinationen von unterschiedlichen Prioritätsregeln [KoPa01] erstellen. Je nach Problemgröße, Komplexität und vorhandener Rechenleistung wird dadurch heute eine entsprechend große Anzahl an Fertigungsplänen erzeugt und der Beste ausgewählt.

Die einfache Implementierung der heuristischen Konstruktionsverfahren im Allgemeinen und die zufriedenstellenden Ergebnisse spiegeln sich in deren hohem Verbreitungsgrad wider. Der erstellte Fertigungsplan kann als Eingangsparameter für Verbesserungsverfahren oder die manuelle Verfeinerung dienen.

2.3.4 Heuristische Verbesserungsverfahren

Verbesserungsverfahren sind auf schon konstruierte Fertigungspläne als Eingangsparameter angewiesen und führen iterative Verbesserungen durch. Mittels heuristischer Analysen werden bessere „benachbarte“ Fertigungspläne ermittelt.

Lokale Suchverfahren untersuchen benachbarte Fertigungspläne zum aktuellen Fertigungsplan [VaAl96, LoMS01, AdZA10]. Das Verfahren gewinnt die benachbarten Fertigungspläne über einfache Modifikationsregeln. Da deren Anzahl (je nach Modifikationsregel) sehr hoch ist, wird nur eine (vielversprechende) Untergruppe davon untersucht, die eine bessere Bewertung als der aktuelle Fertigungsplan hat. Daraus wird ein Fertigungsplan als aktueller Fertigungsplan übernommen. Da lokale Suchverfahren nur einen Teil des Fertigungsplans optimieren, lassen sie sich leicht parallelisieren [YaAZ10].

Fortschrittliche Suchverfahren betrachten den Restriktionsgraph (constraint graph) des aktuellen Fertigungsplans und versuchen durch Analyse Modifikationsschritte für die Einhaltung der Restriktionen abzuleiten (constraint satisfaction). Im Gegensatz zur lokalen Suche ist damit die Ableitung von Verbesserungsschritten mit globalen Auswirkungen möglich, da Restriktionen über den gesamten Fertigungsplan propagiert werden (constraint propagation) [JuLh02, HeMi09].

Einfache Suchverfahren finden selten alleinigen Einsatz. Oft sind sie Bausteine in komplexeren Optimierungsstrategien.

2.3.5 Metaheuristische Verbesserungsverfahren

Für kombinatorische Probleme existiert eine Reihe von Lösungsstrategien. Diese Strategien sind unabhängig von den konkreten Problemen und bestehen aus vordefinierten Schemata zur iterativen Verbesserung der Planvorschläge, weshalb sie auch Metaheuristiken genannt werden [BIRo03]. Viele der Metaheuristiken finden auch Einsatz in der Fertigungssteuerung [SBMR02]. Sie nutzen heuristische Konstruktions- und Verbesserungsverfahren als Basis.

Eine Klasse der Metaheuristiken ist dafür ausgelegt die einfachen iterativen Verbesserungsverfahren zu steuern. Zu den bekanntesten und häufig eingesetzten Metaheuristiken zählt die simulierte Abkühlung (simulated annealing). Iterationen eines Verbesserungsverfahrens dürfen sich bis zu einem bestimmten Kontrollparameter auch verschlechtern, um zu vermeiden, dass das Verfahren vorzeitig in einem lokalen Optimum endet. Nach jeder Iteration verringert sich der Kontrollparameter (Temperatur) und das Verfahren endet letztendlich in einem lokalen Optimum, woher auch der Name des Verfahrens stammt [LaAL92, NaDZ02].

Die Metaheuristik Tabu-Suche soll die erneute Untersuchung bereits untersuchter Lösungsräume in iterativen Suchverfahren vermeiden [Bran93, GeCe04]. Aus diesem Grund speichert sie die (zuletzt) erstellten Fertigungspläne, damit das Iterationsverfahren diese nicht erneut generiert.

Die Metaheuristiken Simulierte Abkühlung und Tabu-Suche sind in der Vergangenheit häufig eingesetzte Verfahren, da sie einfach zu implementieren sind, zu jedem Zeitpunkt abgebrochen werden können und den Ausgangsplan beispielsweise von Prioritätsregelverfahren signifikant verbessern können.

Komplexe naturanaloge Metaheuristiken bilden Evolutionsprozesse, bestimmte Verhaltensmuster in der Tierwelt oder andere ergebnisbringende Prozesse in der Natur nach [BlRo03, HaRC05, KCPL13]. Der genetische Algorithmus (GA) ist eine der ältesten Metaheuristiken, erzeugt sehr gute Ergebnisse für viele (nicht nur) kombinatorische Optimierungsprobleme [Gold89] und wurde in unterschiedlicher Art und Weise für Shop Scheduling Probleme adaptiert [NaYa91].

Das Ablaufschema des GA ist in Abbildung 2.18 skizziert. Der GA unterhält eine Menge (Population) von Individuen, welche die Problemlösungen repräsentieren. In jedem Iterationsschritt entstehen mittels Kreuzungen (crossover, recombination) von je zwei Individuen (parent) neue Individuen (sibling), welche die neue Generation bilden sollen. Anschließend Mutation verändert jedes Individuum in der neuen Generation. Nach einem Evaluierungsschritt bekommt jedes neues Individuum eine Bewertung (fitness). Anschließend entscheidet die ausgewählte Selektionsstrategie (selection), welche der Individuen der neuen und der alten Generation unter Berücksichtigung der Bewertung überleben und die Population für die nächste Iteration bilden.

Für den Einsatz eines GA für die Feinplanung ist die Repräsentation eines Fertigungsplans innerhalb des GA von ausschlaggebender Bedeutung. Die Repräsentation (engl. representation) ist die Kodierung der Individuen und bestimmt die Art der beiden verwendeten genetischen Operatoren – Kreuzung und Mutation. Im Evaluationsschritt erstellt oft ein einfaches Konstruktionsverfahren aus jedem Individuum einen Fertigungsplan und bewertet ihn anschließend.

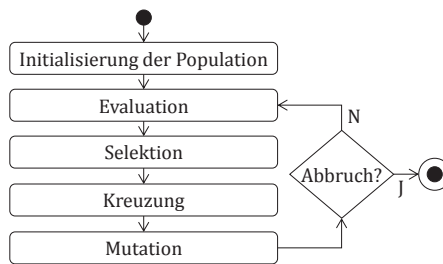


Abbildung 2.18: Ablaufschema des genetischen Algorithmus

Die Forschungswelt hat unterschiedliche Repräsentationen untersucht [ChGT96]. Der ursprüngliche GA kodiert die Lösungen in eine binäre Reihe $\{1|0\}^l$ mit fixierter Länge l . Unverändert kann diese Kodierung auf den disjunktiven Graph eines Shop Scheduling Problems angewendet werden. Dabei bildet jeder Wert aus der binären Reihe die Reihenfolge der Operationen einer bestimmten disjunktiven Kante: Bei 0 ist die Operation mit niedrigem Index vor der mit höherem, bei 1 umgekehrt [NaYa91]. Diese Information ist ausreichend für die Erstellung eines Fertigungsplans. Dementsprechend können auch die ursprünglichen genetischen Operatoren unverändert verwendet werden.

Eine weit verbreitete Repräsentation ist die Vergabe von Prioritäten für die einzelnen Aufträge (oder Operationen), die vom GT Algorithmus verarbeitet werden. Anstelle der binären Werte stehen die Indizes der Aufträge oder Operationen bzw. Zahlen für eine eindeutige Zuordnung zu den Indizes [GeTK94, HJPZ93]. In

diesem Fall ist $l = |O|$. Indizes am Anfang der Reihe haben eine höhere Priorität als solche am Ende. Aus der Repräsentation werden die Prioritäten der Operationen abgeleitet und der GT Algorithmus erstellt damit einen aktiven Fertigungsplan zur Bewertung. Aus diesem Grund liegen Operationen mit höherer Priorität nicht zwangsläufig vor solchen mit niedriger Priorität. Während die binäre Repräsentation aufgrund der Entstehung möglicher Zyklen nicht immer einen Fertigungsplan darstellt, kann aus dieser Repräsentationsart immer ein aktiver Fertigungsplan abgeleitet werden.

Noch komplexere Repräsentationen speichern auch die ausgewählte Maschine [PeMC08]. In diesem Fall bestimmt der GT Algorithmus nur die geplanten Start- und Endzeitpunkte, die Ressourcen selbst bestimmt der GA. Andere Repräsentationsarten kodieren die Startzeiten und die ausgewählte Kombination von Maschinen und zusätzlichen Ressourcen, so dass der Fertigungsplan ohne den GT Algorithmus daraus ableitbar ist [Wall96]. Eine weitere Repräsentation kodiert den Fertigungsplan als eine Folge von Prioritätsregeln der Länge $l = |O|$ (s. 2.3.3) [DoPe95]. Die erste Operation wird anhand der ersten Prioritätsregel gewählt, die zweite anhand der zweiten usw. Diese Repräsentation ist sehr effektiv, jedoch ist ein Individuum im Vergleich zu den oben aufgezählten Repräsentationen nicht aus jedem Fertigungsplan ableitbar. Nicht alle aktiven Fertigungspläne sind mittels einer eingeschränkten Anzahl an Prioritätsregeln repräsentierbar.

Die stark spezialisierten Repräsentationen erfordern auch stark modifizierte genetische Operatoren. Abbildung 2.19 zeigt einige häufig verwendete Operatoren. Dabei sind die Individuen repräsentiert mittels $\begin{pmatrix} \text{Operationsindex} \\ \text{Maschinenindex} \end{pmatrix}$. Mutationsoperatoren verändern die Ressource oder vertauschen Teile innerhalb der Reihe und verändern so die Priorität der Operationen. Zusätzlich ist auf der Abbildung das Ergebnis einer Mutation im Fertigungsplan dargestellt.

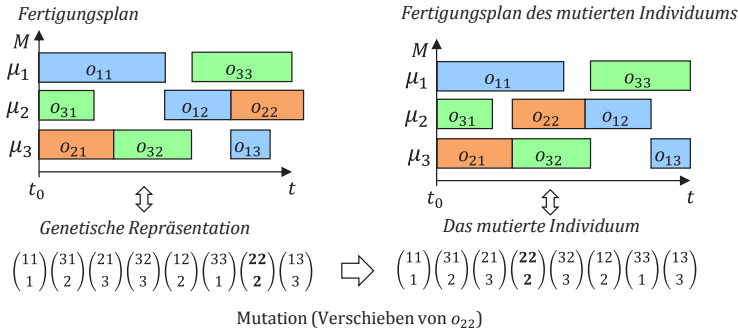


Abbildung 2.19: Beispiel einer Mutation innerhalb des GA und ihre Auswirkung auf den Fertigungsplan

Kreuzungsoperatoren erstellen ein neues Individuum aus Teilen zweier in der Population vorhandener Individuen. Ein weit verbreiteter Operator ist N-Point Crossover, der alternierend Teile aus den Eltern-Individuen entnimmt und das Nachkommen generiert (s. Abbildung 2.20). Dieser Kreuzungsoperator ist sehr verbreitet und mit verschiedener, auch intelligenter Auswahl der Kreuzungspunkte (durch die Zahlenfolge) zu finden [ChGT99]. Ein weiterer verbreiteter Kreuzungsoperator ist der Precedence Preservative Crossover (PPX). Eine Bitfolge der Länge l steuert, welche Gene aus welchem Elternteil vom Nachkommen vererbt werden.

Der Lösungsraum des GA mit dieser Repräsentation umfasst alle aktiven Fertigungspläne. Um den Lösungsraum zu reduzieren, wird oft die ursprüngliche Betrachtungsgrenze des GT Algorithmus mittels eines „look-ahead“ Parameters verringert [MaBi04]. Eine niedrigere Betrachtungsgrenze hinterlässt kleinere Lücken auf den Maschinen und führt zu kompakten Fertigungsplänen.

Diese immer noch nicht vollständige Aufzählung zeigt die breite Palette verschiedener Einsatzmöglichkeiten von genetischen Algorithmen. Je mehr Aspekte des Fertigungsplans die Repräsentation abbildet, desto größer ist der Suchraum des GA und desto

besser das theoretisch erreichbare Ergebnis. Ein Nachteil des größeren Suchraums ist die deutlich erhöhte Laufzeit bis der GA konvergiert.

Genetische Repräsentation der Eltern

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix}$$

1. Auswahl einer Zahlenfolge l_1, \dots, l_n mit $\sum l_i = l$, z.B. 3, 3, 2
2. Auswahl der ersten $l_1 = 3$ aus dem ersten Elternteil für das Nachkommen

Vorläufige genetische Repräsentation des Nachkommens

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix}$$

3. Auswahl der ersten $l_2 = 3$ aus dem zweiten Elternteil, welche nicht im Nachkommen schon enthalten sind und Hinzufügen zum Nachkommen

Vorläufige genetische Repräsentation des Nachkommens

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix}$$

4. Auswahl der ersten $l_3 = 2$ aus dem ersten Elternteil, welche nicht im Nachkommen schon enthalten sind und Hinzufügen zum Nachkommen

Genetische Repräsentation des Nachkommens

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix}$$

Abbildung 2.20: Beispiel einer Kreuzung innerhalb des GA

Der GA wird als ein Verbesserungsverfahren bezeichnet, weil er die initiale Population iterativ verbessert, obwohl er ein Konstruktionsverfahren nutzt. Wenn die initiale Population nicht zufällig erzeugt wird, sondern aus einem Fertigungsplan abgeleitet, wird dieser Fertigungsplan verbessert. Um eine Population mit guter Fitness bereits bei der Initialisierung zu erstellen, führen manche Variationen des GA den GT Algorithmus mit unterschiedlichen Prioritätsregeln aus. Die gewonnenen Individuen repräsentieren einerseits gute Lösungen und sind andererseits mit hoher Wahrscheinlichkeit sehr unterschiedlich (abhängig von den eingesetzten Prioritätsregeln). Dies erhöht die Wahrscheinlichkeit für erfolgreiche Kreuzungen zusätzlich.

Genetische Algorithmen sind auch aufgrund ihres Alters gut erforscht. Andere naturanaloge Verfahren, wie die Ameisenalgorithmen Ant Colony Optimization (ACO), Ant System (AS) und andere Variationen, sind nur halb so alt und wurden erst vor 20

Jahren für Shop Scheduling Probleme adaptiert [CDMT94]. Die Ameisenalgorithmen basieren auf der Selbstorganisationsfähigkeit der Ameisen, das Optimum aus der Weglänge zur Nahrung und der Nahrungsqualität zu finden. Im Shop Scheduling Problem wandert mindestens eine künstliche Ameise durch alle Knoten des disjunktiven Graphs und erstellt so einen Fertigungsplan [DoCG99, RoDi07].

Natural analoge Metaheuristiken haben in den letzten Jahren an Popularität gewonnen. Neuartige Verfahren sind das Artificial Immune System (AIS) und der Shuffled Frog Leaping Algorithmus (SFLA). Das AIS bildet einige der Prozesse im menschlichen Immunsystem ab [CoRC03]. Der SFLA teilt die Fertigungspläne (frogs) in Untermengen (memeplexes) und regelt den Austausch von Verbesserungen innerhalb dieser Untermengen [XWZW12]. Ein Grund für die zunehmende Untersuchung naturanaloger Verfahren ist die Suche nach einer effizienten Selbststeuerungsheuristik [LeBT12], die die Komplexität des Shop Scheduling Problems beherrschen soll.

Die Repräsentationen des Fertigungsplans in den naturanalogen Verfahren haben große Ähnlichkeiten; der Hauptunterschied zwischen diesen Verfahren liegt eher in den Verbesserungsschemata.

2.3.6 Hybride Verfahren

Jedes Verfahren hat seine Vorteile. Um diese zu bündeln, wurden auch Kombinationen aus mehreren klassischen Verfahren vorgeschlagen. Diese Verfahren werden hybrid genannt.

Die Lösungssuche optimaler Konstruktionsverfahren kann durch Verwendung von Heuristiken beschränkt werden. Ein Beispiel dafür ist das heuristische (truncated) Branch&Bound Verfahren, bei dem nicht alle Zweige untersucht werden [FrNS01]. Die

Qualität dieser Verfahren hängt direkt von der verwendeten Heuristik ab.

Einfache heuristische Konstruktions- und Verbesserungsverfahren können auf eine vom menschlichen Benutzer generierte Wissensbasis zugreifen und so Schritte ausführen, die an die konkrete Problemart angepasst sind [Smit92]. Die Schwierigkeit dabei ist die Extraktion und Repräsentation des menschlichen Wissens für das Verfahren.

Konstruktions- und Verbesserungsverfahren sind kombinierbar. Das relativ neue Verfahren Greedy Randomized Adaptive Search Procedure (GRASP) fügt zu den Verbesserungsiterationen noch Ergebnisse aus einfachen Konstruktionsverfahren wie z. B. aus dem GTA mit der Regel „Random“ hinzu [BHLR02].

Bei komplexen metaheuristischen Verfahren wird deren zunehmende Anpassung an die konkrete Problemstellung und Kombination mit anderen Verfahren beobachtet [PuRa05]. Anstelle von zufälligen Mutations- und Kreuzungsoperatoren bei genetischen Algorithmen werden immer öfter spezifische Operatoren für Shop oder Flow Shop Probleme verwendet [ChGT99]. Teilweise weichen diese Operatoren vom ursprünglichen Verfahrenskonzept ab und führen gezielte Änderungen an der genetischen Repräsentation durch [BrWA05, CCFC08, DoGi94]. Eine weitere Änderung ist die nachgeschaltete lokale Suche im Evaluierungsschritt des genetischen Verfahrens. Diese sog. memetischen Verfahren sollen durch Zufall generierte Vorschläge des integrierten genetischen Verfahrens aufwerten und so die Lösungsgüte erhöhen [GoMR05, QWHW08, HSEC09].

Die Repräsentationen vieler populationsbasierter metaheuristischer Verfahren sind sehr ähnlich, was die Kombinierbarkeit dieser Verfahren sehr erleichtert. Ein Beispiel für solche Verfah-

ren sind Kombinationen von ACO und GA [HuWL10] sowie evolutionäre Algorithmen und AIS [WaG004].

In den letzten Jahren nehmen die Versuche mit Kombinationen mehrerer klassischer Lösungsverfahren zu und erzielen auch deutliche Verbesserungen. Das ist einerseits ein Zeichen dafür, dass die klassischen Verfahren nicht zufriedenstellend sind, und andererseits dafür, dass das Forschungsfeld noch nicht ausgereizt ist.

2.3.7 Vergleichbarkeit von Verfahren

Viele der Feinplanungsverfahren können Lösungen für allgemeine Shop Scheduling Probleme erstellen. Während der Entwicklung und Parametrierung verwendete Testdaten haben große Auswirkungen auf die optimalen Einsatzbedingungen der Verfahren. Häufig erzielt ein Verfahren mit zu den Testdaten ähnlichen Problemstellungen gute Ergebnisse, während es bei abweichender Problemstellung nur mittelmäßig abschneidet.

Allgemein anerkannte Testdaten (Benchmarks) sollen die Vergleichbarkeit der Verfahren ermöglichen. Über die Jahre wurden unterschiedliche Benchmarks vorgeschlagen [BeGe12]. Besondere Aufmerksamkeit fanden die Testdaten von Lawrence [Lawr84], Hurink [HuJT94] sowie Dauxere-Perez und Paulli [Paul95]. Das besondere Merkmal der Testdaten ist die oft große Anzahl lokaler Optima, die vergleichsweise hohe Anzahl an zulässigen Fertigungsplänen zur Problemgröße und die schwierige Aufteilung in mehrere Unterprobleme. Weiterhin sind die Daten so konstruiert, dass verbreitete heuristische Prioritätsregeln eher unterdurchschnittliche Ergebnisse erzielen. Optimierungsfunktion ist bei den meisten Testdaten der späteste Fertigstellungszeitpunkt (C_{max}).

Ein Nachteil der verwendeten Benchmark-Probleme ist die starke Berücksichtigung bereits vorhandener Verfahren und die durch-

schnittliche Leistung handelsüblicher Rechner zum Zeitpunkt der Erstellung. Vor ca. 30 Jahren, als die Rechnerleistung eines handelsüblichen Rechners mit der einer heutigen Smart-Armbanduhr vergleichbar war, waren die Testdaten dementsprechend klein. Beispiele mit 5 Aufträgen und 3 Maschinen galten als groß. Die in letzter Zeit verwendeten Problemgrößen haben dagegen einen Umfang von 20-50 Aufträgen und 10-20 Maschinen.

Ungünstig ist außerdem, dass diese Problemgrößen zusammen mit der Zielfunktion C_{max} weit von praktischen Anwendungen entfernt sind. Aus diesem Grund werden viele Verfahren entweder mit echten erfassten Daten aus der Fertigung getestet oder mit Testdaten, die Ähnlichkeit zum angestrebten Anwendungsfall haben.

Generell gelingt die Auswahl eines geeigneten Algorithmus aufgrund der vielen verschiedenen Shop Scheduling Problemarten nur schlecht. Ab einer bestimmten Problemgröße verhalten sich problemspezifische Parametrierungen von Prioritätsregelverfahren oder genetischen Algorithmen bei einer breiten Palette von Problemarten überdurchschnittlich gut.

2.4 Das dynamische Shop Scheduling Problem

Das klassische Shop Scheduling Problem beschreibt eine mathematische Aufgabe mit unveränderten Problemdaten. Diese statische Beschreibung eignet sich zwar für wissenschaftliche Untersuchungen oder für die Bewertung von Investitionsentscheidungen, ist aber für die Abbildung der Steuerung einer laufenden Fertigung nicht ausreichend [MaLu93].

In der realen Fertigungsumgebung bleibt die mathematische Problemstellung der Fertigungssteuerung nur für kurze Zeit

konstant, denn es treten kontinuierlich Ereignisse auf, wie ein Beispiel in Abbildung 2.21 zeigt. In einer laufenden Fertigung werden bestehende Aufträge beendet, modifiziert, storniert und neue eingefügt. Dazu fallen Maschinen oder Personal aus. Auch andere Ereignisse wie Materialausschuss oder Nacharbeit führen zur Verlängerung der Bearbeitungsdauer der Operationen. Dadurch halten Operationen ihren geplanten Endzeitpunkt nicht ein. Infolgedessen blockieren Nachfolgeoperationen weitere Maschinen, was wiederum zu weiteren Verspätungen und organisatorischen Ausfallzeiten führt.

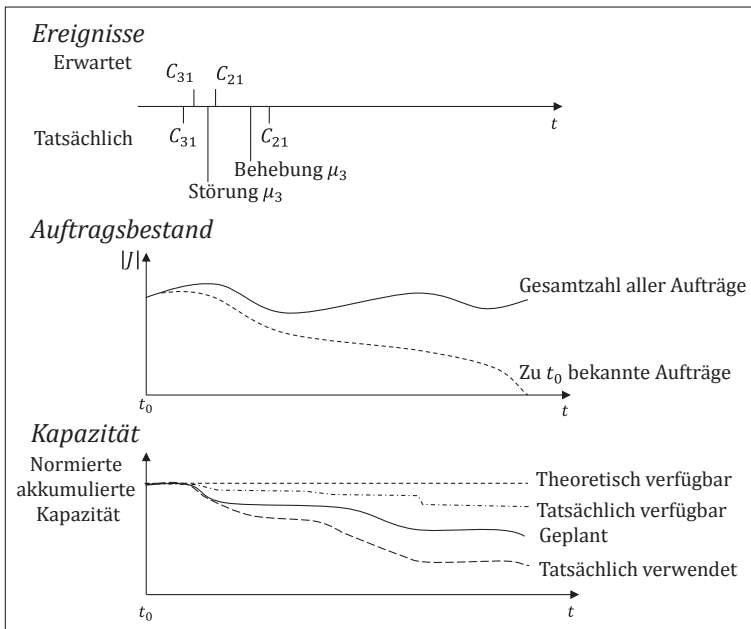


Abbildung 2.21: Beispielhafte Ereignisse in einer laufenden Fertigung

Ein auf die vorige Problemstellung optimierter Fertigungsplan kann sich mit einer leicht geänderten Problemstellung als undurchführbar erweisen [LaSe97]. Die Fertigung ist dann auf

lokale, häufig sehr suboptimale Entscheidungen in der Abarbeitungsreihenfolge der Operationen angewiesen.

„Online“ Feinplanungsverfahren sollen mit der hohen Dynamik in der Fertigung umgehen, um für einen durchführbaren Fertigungsplan bei Änderungen zu sorgen, an den die Fertigung sich halten kann. Aus diesem Grund wurden unterschiedliche Ansätze aufbereitet [OuPe09, PrST04], welche in drei Kategorien zusammengefasst werden können (s. Abbildung 2.22): Unterkapitel 2.4.1 befasst sich mit der Anpassung von Offline-Konstruktionsverfahren für das dynamische Shop Scheduling Problem, während 2.4.2 Online-Verbesserungsverfahren und 2.4.3 die Anwendung populationsbasierter Metaheuristiken beschreiben. Unterkapitel 2.4.4 gibt einen Überblick darüber, wie die durchzuführenden Veränderungen in einem Fertigungsplan gemessen werden, um dadurch die Stabilität des Fertigungsplans trotz einer hohen Anzahl an Umplanungen zu kontrollieren.

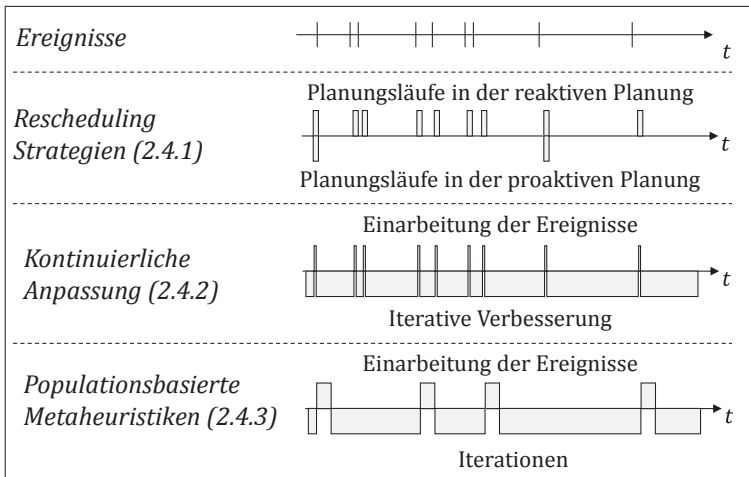


Abbildung 2.22: Gegenüberstellung der unterschiedlichen Ansätze für dynamische Shop Scheduling Probleme

2.4.1 Rescheduling Strategien

In der Forschungswelt wurden viele Verfahren für statische Problemstellungen untersucht. Besonders verbreitete einfache heuristische Konstruktionsverfahren können aufgrund ihrer kurzen Laufzeiten bei Bedarf nach einem Ereignisauftritt schnell einen neuen Fertigungsplan unter Berücksichtigung der geänderten Situation erstellen (rescheduling) [KuSa99, ViHL03].

In einer modernen Fertigung mit automatisierter Fortschrittserfassung treten dauernd Ereignisse auf. Die ständige Erstellung und Verteilung neuer Fertigungspläne würde die Unsicherheit und die Nervosität in der Fertigung und im Fertigungsplan (shop floor und schedule nervousness) steigern, was die Akzeptanz und das Vertrauen der Fertigungsverantwortlichen reduziert sowie weitere negative Effekte mit sich bringt [Puja04]. Um trotz Verwendung von Konstruktionsverfahren eine gewisse Planstabilität (schedule stability) bei gleichzeitiger Bewahrung der Planungsgüte zu erreichen, existieren die folgenden Strategien, welche die Parametrierung (how to schedule) und den Zeitpunkt (when to schedule) der Ausführung bestimmen [SaKi03].

Schon bei der Erstellung des Fertigungsplans kann mittels geeigneter Parametrierung auf dessen Robustheit geachtet werden (predictive scheduling) [JoWS94]. Robuste Fertigungspläne haben längere „Totzeitspannen“ zwischen aufeinanderfolgenden Operationen innerhalb eines Auftrags, so dass sich eine Verspätung beim Vorgänger nicht auf den Startzeitpunkt des Nachfolgers auswirkt. Eine weitere Möglichkeit ist die künstliche Verlängerung der Bearbeitungsdauer der Operationen um die Ausfallrate der ausführenden Maschinen. So berücksichtigt die Planung nicht die maximale theoretische Verfügbarkeit der Ressourcen, sondern die erwartete, und erstellt somit realitätsnahe Fertigungspläne [MeUz98]. Durch diese Maßnahmen haben Aufträge in robusten Fertigungsplänen oft deutlich höhere Durchlaufzeiten und damit

weniger Spielraum zwischen dem frühesten Startzeitpunkt und dem spätesten Fertigstellungsterminen. Dies reduziert die Anzahl gültiger Fertigungspläne und wirkt sich negativ auf die Planungsgüte aus.

Anstatt Puffer für eventuelle Störungen in der Ausführung im Plan einzubauen, kann bei Abweichungen das Konstruktionsverfahren nochmals ausgeführt werden (reactive scheduling). Um nicht bei jedem Ereignis ständig neue Pläne zu generieren, kann die Planung periodisch oder erst ab einem voreingestellten kumulierten Abweichungsgrad (perturbation) angestoßen werden [SaBa00]. Dadurch wird eine hohe Nervosität durch ständige Neuplanung vermieden. Bei dieser Strategie besitzen die Fertigungspläne eine höhere Planungsgüte, jedoch ist die tatsächliche Erreichung der in der Planung ermittelten Kennzahlen in der Ausführung nicht sichergestellt. Im Vergleich können robuste Fertigungspläne eine gewisse Anzahl an Fertigungsabweichungen „überleben“ und die ermittelte Planungsgüte konstant halten.

Diese beiden Hauptstrategien schließen sich nicht gegenseitig aus sondern sind miteinander kombinierbar [VoDH07]. Sie sind mit geringem Aufwand zu implementieren, was sich auf ihren Verbreitungsgrad auswirkt. Es bleibt dann eine Frage der Parametrierung, inwieweit der erstellte Fertigungsplan robust sein soll und bei welchem Grad der Abweichung die Neuplanung angestoßen wird bzw. ein neugenerierter Fertigungsplan übernommen wird.

2.4.2 Kontinuierliche Anpassung

Ein anderes Vorgehen für das dynamische Shop Scheduling Problem ist die Anpassung des Schedule bei Bedarf ohne dessen Neuerstellung. Heuristische Verbesserungsverfahren eignen sich von ihrer Natur her besonders gut für diese Aufgabe. Bei Abweichungen in der Ausführung können sie den Fertigungsplan nur soweit

notwendig iterativ anpassen und dadurch nur eine geringe Nervosität verursachen.

Schon die Anwendung einfacher Umplanungsoperatoren wie Links- und Rechtsverschiebung und Vertauschung zweier Operationen auf einer Maschine kann einen z. B. infolge einer Störung undurchführbaren Fertigungsplan reparieren [MoPV11, NoGr91]. Diese einfachen Operatoren sorgen dafür, dass die organisatorischen Ausfallzeiten wieder mit Operationen belegt werden.

Die heuristischen und metaheuristischen Suchverfahren erreichen noch bessere Ergebnisse in der Wiederherstellung der Plangüte nach Abweichungen [DoKT95, ZDDD93], da sie komplexere Umplanungen durchführen. Um ein performantes „Online“-Verhalten zu erreichen, müssen Verbesserungsverfahren aufgetretene Ereignisse in die Problemstellung aufnehmen, damit sie Abweichungen schon in der nächsten Iteration berücksichtigen können. Bei dieser Aufnahme sind die Aktualisierung von komplexen Hilfsdatenstrukturen und die Neuevaluierung der Kennzahlen des Fertigungsplans notwendig. Im Falle einer metaheuristischen Suchstrategie sind je nach Verfahren die zusätzlichen Daten der Metaheuristik (Tabu-Liste bei Tabu-Suche, Kontrollparameter bei Simulated Annealing) zu aktualisieren.

Oft betreffen die Ereignisse in einer Fertigung nur wenige Operationen, was eine komplette Neuerstellung des internen Datenmodells des eingesetzten Verfahrens nicht rechtfertigt. Aus diesem Grund existieren Methoden zur inkrementellen Berechnung (incremental computation) von verschiedenen Datenstrukturen, die Anpassungen im internen Datenmodell aus den Abweichungen zwischen Fertigungsplan und Ausführung ableiten und durchführen. Solche Methoden sind für die Aktualisierung von eingesetzten Hilfsstrukturen wie Constraint-Graphen, Kapazitätsgebirgen und Intervallbäumen vorgeschlagen worden [ChSm97, LLDV07]. Auch die Berechnung der Zielfunktionswerte kann sogar bei komplexen

Zielfunktionen basierend auf der Standardabweichung oder der Varianz inkrementell berechnet werden [Finc09].

Ansätze zur Anpassung des Fertigungsplans bei Abweichungen in der Fertigung kommen dem menschlichen Agieren näher. Diese Ansätze sind auch in Form einer selbstorganisierten Fertigungssteuerung mit Agenten zu finden [ShNo99]. Derselbe Agent führt sowohl die Datenerfassung als auch die Umplanung durch und reagiert damit schneller auf Störungen. Agentenbasierte Steuerungen brauchen jedoch spezialisierte Agenten zur Aufrechterhaltung des Fertigungsplans, damit sie die Fertigstellungstermine garantieren können.

2.4.3 Anwendung von populationsbasierten Verbesserungsverfahren

Wie in Unterkapitel 2.3 erläutert, erzeugen komplexe populationsbasierte Metaheuristiken die besten Ergebnisse für größere Shop Scheduling Probleme, weshalb auch einige Versuche gemacht wurden, diese für dynamische Problemstellungen anzupassen [BiMa99, ChSu01, RoDi00, RoDi07].

Diese Verfahren unterhalten eine größere Anzahl an Repräsentationen von Fertigungsplänen sowie weitere aus vorigen Iterationen abgeleitete bzw. angepasste Steuerungsparameter, welche die Einführung potentieller Abweichungen in das interne Datenmodell deutlich erschweren. Zusätzlich ist die Anlaufphase für diese Verfahren deutlich zeitintensiver als bei einfachen Suchverfahren, weshalb sie nicht ohne Modifikationen effizient für dynamische Probleme eingesetzt werden können.

Populationsbasierte Metaheuristiken optimieren vom tatsächlichen Geschehen entkoppelt und synchronisieren sich periodisch mit der aktualisierten Problemstellung. Entweder erfolgt die aufwändige Reevaluierung aller Individuen in der Population

unmittelbar nach der Aktualisierung oder die bestehende Population wird durch eine neue Generation ersetzt. Anders als bei den Suchverfahren ist hier die Gefahr von Planinstabilität groß, da die unterschiedlichen Fertigungspläne in der Population nicht unbedingt Ähnlichkeiten mit dem aktuellen Fertigungsplan haben müssen.

Stabilität bzw. Nervosität sind messbar und unterschiedliche Methoden werden im nächsten Unterkapitel erläutert. Die Kennzahl für Nervosität fügt eine neue Dimension in die Zielstellung des Verfahrens ein. Die Zielstellung ist dann immer multikriteriell und bringt die entsprechenden Nachteile (s. 2.2.3) mit sich. Die deutlich bessere Optimierung bei populationsbasierten Metaheuristiken als Vorteil rechtfertigt jedoch die höhere Komplexität des Shop Scheduling Problems. Besonders genetische Algorithmen zeigen gute Ergebnisse im Vergleich zu lokalen Suchverfahren und heuristischen Konstruktionsverfahren. Da die Laufzeit populationsbasierter Metaheuristiken sehr stark von der Problemgröße abhängt, finden sie in einer für eine dynamische Umgebung akzeptablen Rechenzeit weniger gute Fertigungspläne als beispielsweise heuristische Konstruktionsverfahren [BiMa99, OuPe09].

2.4.4 Quantifizierung der Planstabilität

Um eine gewisse Planstabilität bei dynamischen Verfahren sicherzustellen, wird die potentielle Nervosität bei Übernahme eines Planvorschlags quantifiziert. Ein Abstandsmaß zwischen dem aktuellen Fertigungsplan und dem Planvorschlag soll die Nervosität messen [ViHL03]. Darüber wird auf die Planstabilität geschlossen, weshalb dieses Abstandsmaß auch Stabilitätsmaß genannt wird. Für unterschiedliche Personengruppen in einem Unternehmen haben unterschiedliche Aspekte der Planstabilität eine Bedeutung. Beim Personal ist die wechselnde Schichtzuordnung nicht gewünscht. Bei den Vertriebsmitarbeitern sind sich ändernde

de Fertigstellungszeitpunkte der Aufträge und somit des Fertigerzeugnisses ein Grund zur Sorge, in der Logistik die wechselnden Startzeiten der Aufträge etc.

Eine Kennzahl für Nervosität, die alle Beteiligten zufriedenstellt, betrachtet alle Aspekte des Fertigungsplans: Start- bzw. Endzeitpunkte und die zugeordneten Ressourcen der einzelnen Operationen. Basierend auf den Abweichungen zu einem Referenz-Fertigungsplan (#) (s. Abbildung 2.23a) wird die Nervosität am häufigsten durch $\sum_{j,i} |S_{ji} - S_{ji}^{\#}|$ berechnet [WuSC93]. Als Referenz-Fertigungsplan dient der initiale oder der aktuell aktive Fertigungsplan. Andere Metriken verwenden nur die Anzahl der Umplanungen [ChUz92] oder nur die Anzahl der auf andere Maschinen verschobenen Operationen [AlAz03] als Stabilitätskennzahl. Bei Eintritt unvorhergesehener Ereignisse ist eine gewisse Perturbation im Fertigungsplan erwartet. Die Stabilität kann dadurch als Summe der Differenzen zu den erwarteten Startzeitpunkten $\sum_{j,i} |S_{ji} - S_{ji}^{erwartet}|$ ermittelt werden [HeLe05]. Der erwartete Startzeitpunkt hängt von den aufgrund von Störungen notwendigen Verschiebungen (der Vorgängeroperationen) ab.

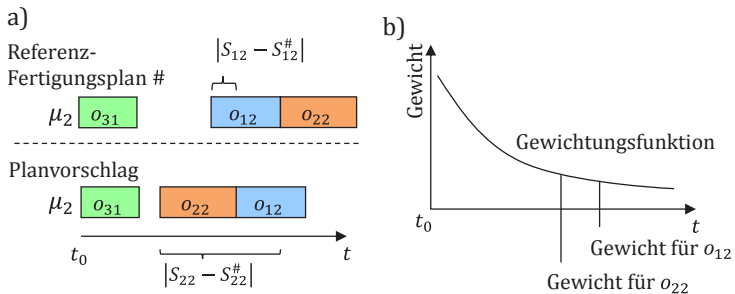


Abbildung 2.23: Abstand zwischen Operationen auf μ_2 der Fertigungspläne aus Abbildung 2.19 mit Gewichtung abhängig von t_0

Die Umplanung unterschiedlicher Operationen im Fertigungsplan verursacht nicht die gleiche Nervosität. Operationen, die weiter in der Zukunft starten, sind nur im Fertigungsplan des Planers

vorhanden. Ihre Verschiebung verursacht nur geringe oder gar keine Nervosität. In der Fertigung unmittelbar bevorstehende Operationen dagegen sind bei vielen Ressourcen angemeldet und es sind oft schon Vorbereitungsarbeiten durchgeführt. Deren Umplanung würde zu großer Unruhe führen. Zur Lösung dieser Probleme in der Praxis wurden zwei Lösungen vorgeschlagen: Bei der ersten werden die Operationen bis zu einer festdefinierten Entfernung (frozen horizon/interval) vom aktuellen Zeitpunkt aus der Umplanung ausgeschlossen [Bake77, ChJi07, LKLB94]. Bei der zweiten werden Operationen mit Startzeiten näher am aktuellen Zeitpunkt höher gewichtet, wie Abbildung 2.23b beispielhaft dargestellt [RaFK04].

Die verschiedenen Metriken für Stabilität, die auch in gewichteten Varianten existieren, zeigen, dass die Stabilität fast genau so schwierig zu messen ist wie die Termintreue selbst (vgl. 2.2.3), da sie sehr subjektiv ist [LeHe05]. Bis jetzt hat sich keine einheitliche Metrik etabliert.

Obwohl dynamische Shop Scheduling Probleme eine deutlich höhere Praxisrelevanz besitzen, sind sie längst nicht so ausgiebig untersucht wie statische Problemstellungen. Gründe dafür sind die fehlenden etablierten Benchmarks, die schwierigere Erstellung von Umgebungen für die Evaluation und die mangelhaften Methoden zur Kollaboration mit den Planern, wie aus dem nachfolgenden Unterkapitel ersichtlich wird.

2.5 Kollaboration zwischen Verfahren und Planern

Die Unterkapitel 2.3 und 2.4 beschreiben die rein algorithmische Lösung statischer bzw. dynamischer Shop Scheduling Probleme als mathematische Aufgaben, ohne dabei die Rolle des Menschen im Planungs- und Steuerungsprozess zu betrachten. Während

sehr viele wissenschaftliche Arbeiten die rein algorithmische Feinplanung untersucht haben, wird der menschliche Faktor nur in einem Bruchteil davon erwähnt [McWi06, S.177–178, Weze06] und die Lücke zwischen Theorie und Praxis ist seit Jahren enorm [King76, McWi99]. Dabei sind die Menschen in produzierenden Betrieben die Verantwortlichen und Entscheidungsträger, während die Verfahren lediglich ein Hilfsmittel darstellen. Besonders in der Feinplanung ist der Faktor Mensch aufgrund vielfältiger Einflüsse zu beachten [Macc01, S.181ff].

Damit ein Verfahren einen Mehrwert für den Menschen, den Planer, bringt, muss es dessen Planungsprozess unterstützen und positiv beeinflussen. Besonders in produzierenden Betrieben, in denen die Fertigung im Mittelpunkt steht, ist die Erstellung eines Fertigungsplans aufgrund des organisatorischen, technischen und soziologischen Kontextes eine sehr schwierige kognitive Aufgabe. Vor der Betrachtung der einzelnen untersuchten Kollaborationsarten zwischen Verfahren und Mensch in der Planung wird in Unterkapitel 2.5.1 der menschliche Planungsprozess in einem produzierenden Betrieb erläutert. Abschnitt 2.5.2 stellt dar, wie die bereits für statische und dynamische Problemstellungen untersuchten Verfahren eine Unterstützung leisten können. Abschnitt 2.5.3 befasst sich mit den sog. wissensbasierten Systemen, die das modellierte Wissen des Planers verwenden, um Fertigungspläne zu erzeugen, die näher an den Wünschen des Planers sind. Verfahren, die eigenständig Änderungen im Fertigungsplan durchführen und nicht einfach eine beratende Rolle spielen, sind in 2.5.4 zu finden.

2.5.1 Der Mensch im Planungsprozess

Aus kognitiver Sicht ist die Planung ein sehr komplexer Prozess, der Merkmale von Problemlösung und Entscheidung beinhaltet, sich jedoch auch davon unterscheidet [Jorn06, S.107–112]. Der

Planungsprozess ist sehr individuell. Wie empirische Studien zeigen, existiert kein vorgefertigtes Schema, nach dem ein Plan entsteht [Jorn06, S.101ff]. Es ist deshalb nicht möglich, den kognitiven Planungsprozess mittels eines Verfahrens nachzubilden.

Der Mensch verfügt über viel Wissen, das in heutigen Informationssystemen nicht modelliert oder nicht modellierbar ist. Nach außen sind die Planer sehr gut vernetzt und pflegen engen Kontakt mit Materialeinkauf und Vertrieb [GaFW10, S.21, McWi06]. Sie wissen, ob Materialbestellungen aufgestockt oder früher geliefert werden können, ob und welche Aufträge wirklich dringend sind und welche verschoben oder sogar aufgestockt werden können. Die Planer wissen von Aufträgen, die aufgrund von saisonalen Schwankungen oder mündlichen Verhandlungen nicht in Informationssystemen eingetragen sind, und reservieren Fertigungskapazität dafür. Somit haben sie theoretisch deutlich mehr Spielraum als ein algorithmisches Verfahren.

Auch nach innen sind die Planer sehr gut vernetzt, arbeiten eng mit Schichtleitern, Maschinenoperatoren und anderem Personal zusammen und wissen um deren Besonderheiten und die des Maschinenparks [SnWe10]. Die Planer wissen, wann Sonderschichten und Überstunden organisierbar sind, welche Aufträge im Hinblick auf den Kunden und die erforderliche Qualität mit welchem Personal und welchen Maschinen besser zu fertigen sind. Denn während es in einem neuen Maschinenpark keine erheblichen Unterschiede gibt, weisen veraltete und teilweise ausgetauschte Maschinen deutliche Qualitäts- und Quantitätsdifferenzen auf. Nur ein kleiner Anteil dieses Wissens ist zurzeit in Planungssystemen modellierbar (vgl. 2.2.2).

In Organisationen sind oft mehrere Planer zum Teil in verschiedenen Schichten in den Prozess involviert. Sie führen eigenmächtig Planänderungen durch. Dieser sog. mixed-initiative Vorgang ist nicht ohne eine enge Abstimmung unter den Planern möglich.

Neben dem oben beschriebenen Wissen verfügen die Planer über die soziale Kompetenz, die für den Ablauf des Planungsprozesses erforderlich ist [McWi06].

Selbst ein sehr fortgeschrittenes Verfahren verfügt nicht annähernd über dieses Wissen und diese Fähigkeiten. Der Mensch ist im Planungsprozess unersetzbar und wird in absehbarer Zeit nicht von einem Verfahren abgelöst werden können. Die steigende Komplexität in der Fertigung, die Verkürzung der Produktionszyklen und der Kostendruck zwingen den Planer zur Verwendung neuer technischer Hilfsmittel, die in den nachfolgenden Abschnitten beschrieben werden.

2.5.2 Entscheidungsunterstützung in der Planung

Die Planung ist ein sehr rechenintensiver Prozess, der selbst bei kleinen Problemstellungen hunderte einfacher mathematischer Operationen erfordert [Jorn06, S.118]. Aus diesem Grund nutzen die verschiedenen Planer oft grafische Darstellungen und arbeiten weniger mit Zahlen. In den innerhalb des Planungsprozesses zu treffenden Entscheidungen kann der Planer von einfachen Planungswerkzeugen (Leitständen) und Entscheidungsunterstützungssystemen (Decision Support System, DSS) begleitet werden, deren Hilfe sich auf die Planerstellung, die Visualisierung und die manuellen Umplanungen beschränkt [FrRBAT0, TBBL05, Weze06].

Der Planer muss zur Planerstellung aus den verfügbaren Algorithmen auswählen, sie parametrieren und die erstellten Lösungen einander gegenüber stellen und eine auswählen. Er interagiert dabei nicht direkt mit dem Verfahren, sondern mit dessen Input und Output. Aus diesem Grund sind auch unmodifizierte Verfahren für statische Problemstellungen geeignet. Meist werden heuristische Konstruktionsverfahren hierfür eingesetzt (s. 2.5.2). Komplexere und zeitintensive Verfahren wie genetische Algo-

rithmen oder heuristisches Branch&Bound werden dagegen oft in nächtlichen Planungsläufen eingesetzt [WeCH10, S.342–343].

Die Auswahl des Fertigungsplans aus einem Verfahren stellt eine gute Basis für manuelle Interaktionen dar. Unterschiedliche Interaktionskomponenten wie Gantt Diagramme, stetige und diskrete Intervall-Gebirge für die unterschiedlichen Planungsaufösungen unterstützen den Planer im Planungsprozess (s. Abbildung 2.24) [HBBW01, S.130–170]. Oft sind nachgeschaltete iterative Verbesserungsverfahren nur bei großen Planungsproblemen erwünscht. Denn sobald der Fertigungsplan sich in einem lokalen Optimum befindet, würde jede Benutzerinteraktion zu einer Verschlechterung der Kennzahlen führen, die wiederum nur durch viele weitere Interaktionen zu verbessern sind. Um diesen Teufelskreis zu vermeiden, wird der Planer seine Interaktionen auf das Notwendige reduzieren.

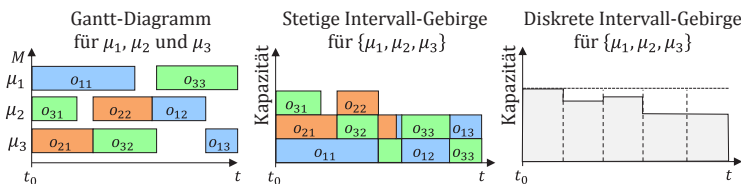


Abbildung 2.24: Visuelle Komponenten zur Darstellung des Fertigungsplans: Gantt Diagramm, stetige und diskrete Intervall-Gebirge

Nach der Auswahl eines Fertigungsplans beschränkt sich die algorithmische Unterstützung auf die Aktualisierung der Bedienoberfläche und auf Hinweise für unzulässige, nicht durchführbare Stellen im Fertigungsplan. Fortgeschrittene DSS können bei Bedarf mit einfachen Operationen die Zulässigkeit des Fertigungsplans (oft bei Verschlechterung der Kennzahlen) wiederherstellen.

Dieser rudimentäre Ansatz ist mit den bereits für statische Problemstellungen untersuchten Algorithmen möglich und deshalb

sehr beliebt. Der Planer übernimmt die Anpassung des Fertigungsplans bei Ereignissen, während das Verfahren einen initialen Fertigungsplan erstellt. Mehr Einflussmöglichkeiten auf das Verfahren hat der Planer in wissensbasierten Systemen.

2.5.3 Wissensbasierte Planung

Feinplanungsverfahren wurden ursprünglich für allgemeine Planungsprobleme entwickelt und bieten einen gewissen Konfigurationsspielraum für spezifische Probleme. Wie in 2.5.1 erwähnt, existiert ein Teil des Fertigungsmodells bzw. Planungsproblems nur als Wissen bei den Planern. Die Idee hinter der wissensbasierten Planung ist es, dieses Wissen und ihre Präferenzen in eine für das Verfahren verständliche Form zu übertragen. So soll das Verfahren die zusätzlichen Restriktionen der Planer einhalten und den Fertigungsplan nach ihrer Art und Weise erstellen bzw. modifizieren, so dass Interaktionen auf das Minimum reduziert werden.

Wissensbasierte Planungssysteme haben ein flexibles erweiterbares Planungsmodell und zusätzlich ein flexibles Verfahren, das damit effektiv umgehen kann. Die Planer sollen das Planungsmodell mithilfe einer dafür geeigneten (meist grafischen) Benutzeroberfläche erweitern. Die Erweiterungen des Modells beziehen sich eher auf unterschiedliche Restriktionen zu den Planungsobjekten (Maschinen, Aufträge, Operationen, Ressourcen) und weniger auf die Planungsobjekte selbst, wie die Fokussierung in der Forschung zeigt [DoGV98, FoSm84, SaBr97, SmLB96, SMM090]. Verschiedene Stufen von Restriktionen beschreiben die Einhaltungsstufen hard, soft und preference-based. Entsprechend bestimmen sie die Zulässigkeit, die primären und die sekundären Optimierungsziele. Wissensbasierte Systeme betrachten das Planungsproblem meistens als Bedingungserfüllungsproblem (Constraint Satisfaction Problem, CSP) [BaSR10, DoKa01] und

weniger als ein Optimierungsproblem, weshalb die Optimierung von Zielkriterien wie Rüstzeitminimierung, die nicht durch Restriktionen abbildbar sind, nicht möglich ist.

Wissensbasierte Planungssysteme sollten die Lücke zwischen den theoretischen Planungsmodellen und den realen Fertigungsorganisationen schließen. Die notwendigen GUIs erfordern im Vergleich zu den interaktiven Planungssystemen weitere grafische Komponenten zur Wissensmodellierung. GUIs für Feinplanungssysteme sind bekannter Weise sehr aufwändig zu entwickeln, weshalb die Forschung dieses Gebiet oft meidet oder nur eingeschränkt untersucht.

2.5.4 Mixed-Initiative Mensch-Maschine Feinplanung

In den vorigen zwei Ansätzen führen Planungsverfahren kleinere und nur kontrollierte Änderungen im Fertigungsplan durch. Größere Änderungen oder die Planerstellung finden nur auf Initiative des Planers statt. Bei einer mixed-initiative Interaktion initiieren verschiedene Beteiligte Planänderungen. Da sie in einer gemeinsamen Datenbasis, dem Fertigungsplan, ihre Interaktionen durchführen, ist für eine harmonische Zusammenarbeit Absprachen erforderlich.

In einer mixed-initiative Planung ohne Verfahren können die Planer mittels der natürlichen Sprache ihre Interaktionsbereiche organisatorisch abstimmen, damit sie sich nicht überlappen [Gaze06]. Die Absprache erfolgt größtenteils informell. Fortgeschrittene u. a. auch kommerzielle Systeme erlauben das Sperren von Ressourcen und/oder Zeitbereichen für die exklusive Planung oder die Ausführung von Verfahren auf einem eingeschränkten reservierten Planbereich und unterstützen so die informelle Absprache.

In einer mixed-initiative Planung ohne Planer kommunizieren die planenden Einheiten (meist Software-Agenten) mittels einer vorher definierten formalen Sprache (Protokoll) [ShWH06, SHYN06]. Aufgrund der heute fortgeschrittenen Kommunikationstechnik sind sie in der Lage, große Informationsvolumina auszutauschen und ihre Planungsschritte untereinander abzustimmen. Oft existiert kein Fertigungsplan bzw. ist er auf die Agenten verteilt und deshalb ist eine zentrale Einheit für die Zusammenführung der Teilpläne notwendig, um Garantien an übergeordnete Instanzen abgeben zu können [Gaze06, S.162–163]. Die Kommunikation zwischen den Agenten erfolgt mittels einer festdefinierten formalen Sprache, weshalb unvorhergesehene Situationen oder Interaktionen zur Verkomplizierung von sonst einfachen und überschaubaren Planungsproblemen führen [BoJV06]. Aus diesem Grund ist die Forschung mit der Suche von optimalen Absprachemechanismen beschäftigt und greift hierfür oft auf Muster aus biologischen Systemen (Ameisen, Bienen...) oder der Marktwirtschaft (Handelsplätze) zurück [LeBT12, RWM005].

Eine mixed-initiative Interaktion mit Planern und Verfahren bringt noch größere Herausforderungen in Bezug auf den Absprachemechanismus mit sich [WeCH10]. Der Planer sieht das Verfahren als ein Hilfswerkzeug, das ihn während der Planung nach Wunsch und Bedarf unterstützt. Da die menschliche Planung ein sehr komplexer und individueller Prozess ist, ist die erwünschte Unterstützung entsprechend schwierig formal zu definieren. Je mehr Veränderungen ein Verfahren im Fertigungsplan eigenständig durchführen darf, desto mehr muss seine Aktion mit den Planern im Einklang sein. Seine Aktionen müssen nachvollziehbar sein und Verbesserungen bringen. Gleichzeitig dürfen sie mit den eventuellen Aktionen des Planers nicht kollidieren.

Die Anforderungen an das Verhalten des Verfahrens werden von einfachen Suchverfahren zum Großteil erfüllt (vgl. 2.3.4). Diese Suchverfahren agieren meist deterministisch nach vorgegebenen

Regeln. Sind die Regeln einfach, so sind die durchgeführten Änderungen auch für die Planer nachvollziehbar. Die meisten Regeln beschränken sich auf das Austauschen einiger Operationen oder die Wiederherstellung der Zulässigkeit des Fertigungsplans [OwST88, Smit95, Weze06].

Entwicklungen aus anderen Domänen, in denen die Verfahren einen größeren Freiheitsgrad im Vorschlagen und Übernehmen von Änderungen haben [CBCC04, WeJo09], sind in der Fertigungssteuerung noch nicht angekommen. In der Fertigungssteuerung haben auch kleine Umplanungen oft weitreichende Auswirkungen im Fertigungsplan, da der Fertigungsplan meist sehr schwierig in unabhängige Bereiche aufzuteilen ist (schedule decomposition) [OvUz97]. Deshalb wird die eigenständige Verfahrensaktion künstlich eingeschränkt.

Ein Hauptgrund für den zurückhaltenden Forschungsfortschritt in der Mensch-Maschine Interaktion in der Fertigungssteuerung liegt in der schwierigen Erfahrungssammlung und Erprobung von Prototypen [Weze06]. Prototypen für intelligente Planung sollen bestehende Fertigungsabläufe nicht verschlechtern und produzierende Betriebe bei der Erprobung nicht übermäßig überfordern. Wie aus dem nächsten Unterkapitel ersichtlich wird, ist selbst der Einsatz von am Markt etablierten Systemen eine große Herausforderung für die Unternehmen.

2.6 Planung in der Unternehmens-IT

In der Unternehmens-IT bekommen Planungsverfahren unterschiedliche Aufgaben. So finden sie beispielsweise Einsatz im Versorgungskettenmanagement (Supply Chain Management), in Investitionsrechnungen oder in der Materialplanung. Im Vergleich zu diesen Bereichen besitzt die Fertigungssteuerung eine höhere Dynamik und Komplexität. Um sie in den Griff zu bekommen, ist

eine algorithmische Unterstützung in verschiedenen kommerziellen Systemen implementiert. Einen kurzen Überblick über diese Systeme gibt Abschnitt 2.6.1. Der nachfolgende Abschnitt 2.6.2 listet die in der Praxis eingesetzten Verfahren auf. In 2.6.3 werden die Gründe und Hindernisse für den Einsatz solcher Systeme und Verfahren in der Fertigungssteuerung untersucht.

2.6.1 Fertigungseinplanung in Softwaresystemen

Das Konzept für die IT-gestützte Fertigungssteuerung befindet sich in ununterbrochener Entwicklung (s. 2.1.1). Um dieses Konzept zu unterstützen, wurden über die Jahre Funktionen und Verfahren für die Fertigungssteuerung in unterschiedlichen IT-Systemen implementiert.

Heute bilden verschiedene IT-Systeme für die Unterstützung der Fertigungssteuerung aus dem Forschungsstand der letzten ca. 30 Jahre einen komplexen Mix in der Unternehmens-IT. Ursprünglich wurde die Fertigungssteuerung in IT-Systemen im Unternehmen nur rudimentär implementiert. Die oft mangelnde algorithmische Unterstützung und die mangelnde Berücksichtigung von Ressourcenkapazitäten haben die Produktionsplanungs- und Steuerungssysteme (PPS-Systemen) etabliert, die stärker auf die Anforderungen aus der Fertigung eingehen sollten [FaFG94]. Da diese Systeme Restriktionen und Besonderheiten in der Feinplanung nur sehr eingeschränkt berücksichtigen, wurde die Fertigungssteuerung und -feinplanung teilweise in MES verlagert und mit Feinplanungsverfahren ausgestattet. Weil MES zu nahe an der Fertigung ist und eine zu kurze Sicht auf die Planung hat, wurden Systeme für Advanced Planning and Scheduling (APS) implementiert [MeWR08]. APS Systeme sollen sowohl die kurzfristige als auch die langfristige Planung erstellen.

Zurzeit ist kein einheitliches Konzept für die Fertigungssteuerung durchsetzbar, da sie mit dem im jeweiligen Unternehmen vorhan-

den Mix aus MES, PPS, ERP und/oder APS Systemen realisiert wird. Die unterschiedlichen Systeme übernehmen unterschiedliche Teilaufgaben der Fertigungssteuerung und sind oft hierarchisch angeordnet. Oft sitzen die ERP Systeme auf der obersten Ebene der Systemhierarchie und erstellen die Fertigungsaufträge, während untergeordnete Systeme die Fertigung in unterschiedlichen Auflösungen (resolution) planen. Obwohl die Systeme sich in Funktionalität und Auflösung deutlich unterscheiden, verwenden sie oft die gleichen Grundtypen von Verfahren, die im folgenden Abschnitt beschrieben werden.

2.6.2 Eingesetzte Feinplanungsverfahren in der Praxis

Forschungslabore haben eine breitere Palette an unterschiedlichen Planungsverfahren untersucht (vgl. 2.3 und 2.4). Doch ganz wenige Algorithmen finden Einsatz in kommerziellen Produkten. Hier werden die verbreiteten Algorithmen [Boze12, Thie11, S.476–648] aufgezählt.

Einfache heuristische Konstruktionsverfahren

Die einfachen heuristischen Konstruktionsverfahren sind breit studiert und ausführlich in der Literatur beschrieben (vgl. 2.3.3). Sie sind auch Bestandteil der Inhalte von Wirtschaftsstudiengängen und ohne viel Aufwand in einem MES, aber auch in einem ERP oder APS System implementierbar. Mit Hilfe von einfachen Regeln wird die Priorität der Operation auf der Maschine ermittelt. Operationen mit höherer Priorität starten früher als andere. Systeme wie SAP ERP haben über 50 Heuristiken [Hopp12, S.693–750]. Oft bietet jede Heuristik über Parameter auch selbst weitere Einstellmöglichkeiten. Der Planer muss allerdings die passende Heuristik samt Parametrierung für seine Fertigung durch Ausprobieren selber finden. Manche Systeme wie Dualis ERP iterieren

über alle Heuristiken automatisch. Da Heuristiken oft sehr suboptimale Ergebnisse produzieren, existieren unterschiedliche Regelsätze für spezielle Fertigungsarten wie z. B. die Wafer Fertigung [MöFM13, SaVW11]. Diese Regelsätze sind sehr komplex und müssen aufwändig an die Fabrik angepasst werden.

Genetische Algorithmen

Genetische Algorithmen (vgl. 2.3.5) finden Einsatz in unterschiedlichen Bereichen von der Finanzwirtschaft über die Ingenieurwissenschaften bis hin zu den Naturwissenschaften und seit einigen Jahren auch in einigen kommerziellen Systemen für die Fertigungssteuerung. Systeme wie SAP APO setzen GA zur Generierung von Fertigungsplänen ein [GüGN06]. Bei SAP APO z. B. wird das Verfahren mit großen Pufferzeiten zwischen den Operationen initialisiert, um einen robusten Fertigungsplan zu erzeugen. Sie werden sukzessiv verringert, falls keine zulässige Lösung gefunden wird. Der Lauf von GA umfasst oft mehrere Stunden und wird deshalb nachts durchgeführt. Andere Systeme wie MPDV Mikrolab nutzt GA zur Einstellung der Parametrierung von heuristischen Konstruktionsverfahren.

Mathematische Optimierungsmethoden

Eine Reihe von universellen kommerziellen Produkten mit mathematischen Optimierungen hat sich auch im Bereich der Fertigungsfeinplanung spezialisiert. Firmen wie ILOG, Gurobi oder Quintiq bieten als Dienstleistung die problemspezifische Abbildung des konkreten Fertigungsmodells in einem mathematischen Modell [Pape94]. Zum Beispiel werden zurzeit Fabrikbereiche für die Wafer Herstellung von IBM und Taiwan Semiconductor Manufacturing Company (TSMC) mit ILOG CPLEX optimiert.

Außer diesen Verfahren gibt es viele einzelne projektspezifische Prototypen wie das Reparaturplanungssystem des Space Shuttles

[ZDDD93], die die Entwicklung hin zu einem Produkt nicht geschafft haben. Trotz enormer nachgewiesener Ersparnisse ist der Weg hin zu einem Massenprodukt sehr schwierig.

2.6.3 Hindernisse für den Einsatz von Feinplanungssystemen

Trotz der potentiellen bedeutenden Ersparnisse werden die Komponenten für die Fertigungssteuerung in Software-Systemen innerhalb der Unternehmens IT am wenigsten eingesetzt [Jorn06, S.117]. Die Gründe dafür können grob in verfahrensbezogen und nicht verfahrensbezogen aufgeteilt werden.

Die nicht verfahrensbezogenen Gründe liegen in den Anforderungen, die ein Verfahren an die Stamm- und Bewegungsdaten stellt. Diese Daten bilden die Problemstellung für das Shop Scheduling Problem. Sind sie fehlerhaft bzw. bilden sie die Realität fehlerhaft ab, so ist auch der erstellte Plan in der Realität nicht durchführbar. Häufige Fehler sind fehlende oder falsch eingegebene Ausführungszeiten von Operationen [ChHT09]. Noch schwieriger ist es, wenn durch die Manipulation dieser Daten die Fertigungskosten in ERP Systemen bestimmt werden. Oft gibt es sogar Abweichungen von über 70% zwischen berechneten und tatsächlichen Ausführungszeiten [DiBS10]. Der Erfolg der Einführung eines Feinplanungssystems hängt von der Qualität der Stamm- und Bewegungsdaten ab.

Für adäquate Aktionen sowohl des Feinplanungsverfahrens als auch des Planers ist zusätzlich die Kenntnis über den aktuellen Fertigungsstand notwendig [Loos99]. Oft meldet das Personal nur einen Teil der Störungen und führt die Auftragsmeldungen erst am Ende der Schicht aus. Nicht selten wird die Schichtdauer auf die fertigen Prozesse verteilt. Planungssysteme bieten Mechanismen für die Anpassung der Prozesszeiten aufgrund von erfassten Fertigungsmeldungen. Die fehlenden Störungen und die falsch

erfassten Prozesszeiten stören diese Mechanismen zur Berechnung und geben weiterhin auch keine Informationen über die tatsächlichen Leerzeiten der Maschinen. Aus den oben genannten Gründen planen Verfahren längst beendete Prozesse um oder berechnen das erwartete Ende eines Prozesses falsch. Dies führt zur Entkopplung der Feinplanungssysteme von der Fertigung und letztendlich von deren unmittelbarer Steuerung, die ein essenzieller Bestandteil des dynamischen Planungsprozesses ist [Hoc06].

Die genannten Hindernisse sind oft nur mit viel Aufwand zu überwinden. Das erschwert die Inbetriebnahme solcher Systeme. Das Hauptproblem liegt jedoch in den eingesetzten Verfahren mit all den Nachteilen, die in den letzten drei Unterkapiteln aufgezählt wurden. Die einfachen heuristischen Konstruktionsverfahren produzieren Ergebnisse, die selbst von unerfahrenen Neulingen zu übertreffen sind [Kuo99] und den Planungsprozess sogar eher behindern [Baek99]. Genetische Algorithmen und mathematische Optimierungsmethoden sind aufwändig einzustellen, haben sehr lange Laufzeiten und sind zurzeit nur in einigen wenigen Systemen verbreitet.

Die Übernahme der fortgeschrittenen Planungssysteme Preactor und ILOG durch die Konzerne Siemens bzw. IBM spiegelt die Markterwartung des lang vorhergesagten Durchbruchs im Bereich der fortgeschrittenen algorithmischen Fertigsteuerung wieder. Das aktuelle Konzept für die Interaktion zwischen Verfahren und Planer sowie für die Rolle des Verfahrens selbst ist nicht mehr zeitgemäß, wie aus dem nächsten Kapitel ersichtlich wird.

3 Analyse und Anforderungsermittlung

Das vorige Kapitel beschreibt die verschiedenen Forschungsfelder und die aktuellen Planvorschläge im Bereich der Fertigungssteuerung. Eine besondere Herausforderung bilden permanentlaufende Feinplanungsverfahren für eine laufende Fertigung unter Einbeziehung von Benutzerinteraktionen, da sie die Herausforderungen mehrerer Forschungsfelder bewältigen müssen. Die Feinspezifizierung der Anforderungen an diese Feinplanungsverfahren erfolgt nach einer genauen Untersuchung des heutigen und des gewünschten Umgangs mit Feinplanungssystemen.

Unterkapitel 3.1 analysiert die aktuell implementierten Konzepte für Fertigungssteuerung und die Rolle des Feinplanungsverfahrens darin. Zum Vergleich werden in 3.2 neuartige Fertigungssteuerungskonzepte sowie das gewünschte intuitive Verhalten der Feinplanungsverfahren gegenüber gestellt. Daraus werden die Anforderungen an die moderne algorithmische Fertigungssteuerung in 3.3 abgeleitet. 3.4 listet die technischen Anforderungen und Beschränkungen aus dem aktuellen Stand der Technik auf.

3.1 Feinplanungsverfahren in der Fertigungssteuerung heute

Eine Untersuchung von verfügbaren Systemen mit Fertigungssteuerungskomponenten [Boze12, Thie11, S.476–648] zeigt große Übereinstimmungen in der Integration der Fertigungssteuerung in der Unternehmens-IT und im Einsatz der Feinplanungsverfahren (vgl. 2.6). Der Grund dafür ist die Evolution und die

allgemein akzeptierte Definition der Fertigungssteuerung als Regelkreis (vgl. 2.1.1).

Alle Systeme zur Fertigungssteuerung können zwischen der Automatisierungsebene und dem ERP System eingeordnet werden (s. Abbildung 3.1). Die einzelnen Maschinen sind meist direkt an die Fertigungssteuerung angebunden und das ERP System hat nur über die Ebene 3 Zugang zur Fertigung [Ansi13a, Vdi07]. Die Maschinen melden den aktuellen Fertigungsstand und -fortschritt, das ERP-System die vorhandene und geplante Kapazitätsverfügbarkeit sowie den Auftragsbestand. Auf Grundlage dieser Information erstellt die Fertigungssteuerung den Fertigungsplan und teilt ihn ihren Kommunikationspartnern bei Bedarf mit. Jede Person in der IT-Landschaft sieht nur den Teil des Fertigungsplans, den ihm sein IT-System anbietet. Änderungen im Fertigungsplan sind auf Ebene 3 vorgesehen [Ansi13a, Vdi07].

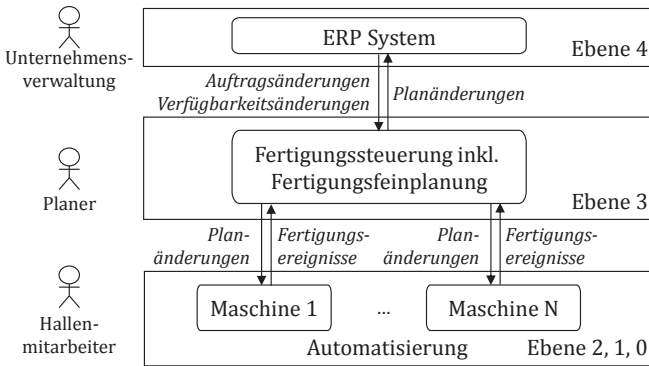


Abbildung 3.1: Grobe Abbildung und Einordnung der Fertigungssteuerung nach dem ISA-95 Standard [Ansi13a]

Da die Feinplanung oft als Aufruffunktion und nicht als kontinuierlich laufender Dienst angesehen wird, ist sie nicht selbstständig und erfordert das explizite Eingreifen von außen. Unabhängig vom Umfang der grafischen Darstellungsmöglichkeiten oder der algorithmischen Ausprägung weisen alle heutigen Feinplanungs-

systeme ein sehr ähnliches Verhalten auf (vgl. 2.6) und stellen den Planer in den Mittelpunkt [RHMD10a, S.315]. Dieses Verhalten ist in allen Feinplanungssystemen zu finden und resultiert aus dem Wunsch des Planers die volle Kontrolle über das Planungsergebnis zu besitzen.

Feinplanungssysteme sind auf den Aufruf des Feinplanungsverfahrens (H1) und auf die Verwendung der GUI für Umplanungen (H2) angelegt (s. Abbildung 3.2). Dementsprechend sind sie genau dafür optimiert. Ein dritter Anwendungsfall (H3) ist die nur bedingte Verwendung des Feinplanungssystems, falls eine schnelle Reaktion erforderlich ist.

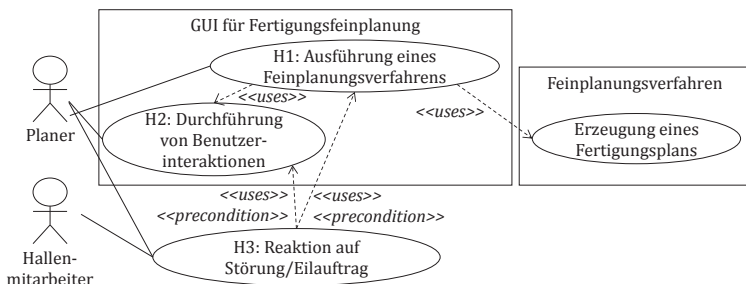


Abbildung 3.2: Anwendungsfälle in bestehenden Fertigungssteuerungssystemen

Anwendungsfall H1: Ausführung eines Feinplanungsverfahrens

Der Einsatz der Feinplanungsverfahren erfordert viele manuelle Aktionen seitens des Planers (s. Abbildung 3.3). Der Planer wählt das Feinplanungsverfahren und führt eine Parametrierung abhängig von der gewünschten Zielsetzung aus. Das ausgewählte Feinplanungsverfahren schlägt in einem vertretbaren zeitlichen Rahmen einen oder mehrere Fertigungspläne vor. Der Benutzer überprüft die vorgelegten Kennzahlen (KPIs) des erstellten Fertigungsplans/der erstellten Fertigungspläne [RHMD10b]. Entweder wählt er einen Fertigungsplan aus oder er parametriert die Fein-

planungsverfahren mit anderen Werten. Diese Entscheidung wird meist subjektiv getroffen. Nach der Auswahl führt der Benutzer eine visuelle Prüfung der Belegungen der einzelnen Operationen durch, wobei er den Fertigungsplan nach seinen subjektiven Kriterien evaluiert. Der Grund dafür ist beim Planer vorhandenes Wissen über die Fertigung, das nicht im Planungsproblem modellierbar ist (vgl. 2.5.1). Nach dieser Beurteilung hat er zwar immer noch die Möglichkeit einen neuen Fertigungsplan zu generieren, doch meistens konzentriert sich der Planer relativ schnell auf einen Fertigungsplan und setzt seine Arbeit damit fort [WeCH10, S.343].

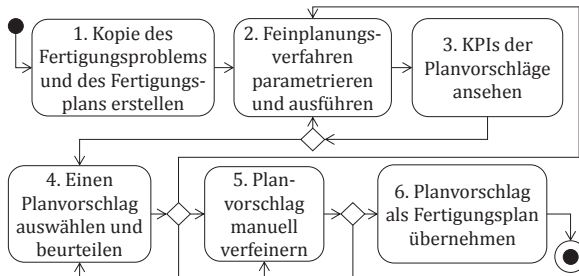


Abbildung 3.3: Fertigungsfeinplanung mittels Planungsverfahren (Anwendungsfall H1)

Dieser Ablauf überfordert den Planer oft und bringt einige bedeutende Nachteile mit sich:

- Der Planer muss über die entsprechenden Kenntnisse in Bezug auf das Verhalten der Feinplanungsverfahren verfügen, um die gesetzten Optimierungsziele korrekt in deren Parametrierung umzusetzen (Schritt 2) und so ein optimales Planungsergebnis zu erreichen.
- Die Feinplanungsverfahren untersuchen wenige mögliche Fertigungspläne (Schritt 2) und dienen oft der Erstellung eines Basisplans für manuelle Verbesserung. Denn die Feinplanungsverfahren sollen in einer für den Benutzer akzeptablen Zeit von einigen Minuten ein Ergebnis liefern.

Dadurch nutzen die Feinplanungsverfahren nur ca. 1-2% der verfügbaren Rechenzeit [FoSm84]. Eine Ausnahme bilden hier nächtliche Planungsläufe, die komplexe Feinplanungsverfahren ausführen (z. B. SAP). Hier erhält der Planer einen deutlich besseren Fertigungsplan, kann aber die Auswirkung der Parametrierung des Feinplanungsverfahrens erst am nächsten Tag beurteilen.

- Der Ablauf ist historisch bedingt nur für einen einzelnen Benutzer konzipiert. Bei der Verwendung eines Feinplanungssystems durch mehrere Planer in größeren Betrieben kann eine Übernahme eines Fertigungsplans (Schritt 6) möglicherweise Veränderungen anderer Benutzer überschreiben (Schritt 5). Deshalb müssen die Planer ihre Interaktionen per E-Mail, in Sitzungen oder telefonisch koordinieren.
- Das Feinplanungssystem sieht den Planer als einzigen Benutzer vor. Andere Fertigungsverantwortliche haben über den Planer nur einen eingeschränkten und indirekten Einfluss auf den Fertigungsplan.

Anwendungsfall H2: Durchführung von Benutzerinteraktionen

Direkte Anpassungen am Fertigungsplan geschehen über Interaktionen mit dem Feinplanungssystem (s. Abbildung 3.4). Der Benutzer ändert die Ausführungszeiten oder Ressourcenbelegungen von Operationen in seiner Kopie des Fertigungsplans. Entscheidet sich der Benutzer nach den Interaktionen für die Ausführung eines Feinplanungsverfahrens, muss er sich wieder in einen neuen Fertigungsplan einarbeiten, mit allen Nachteilen wie im zuvor beschriebenen Anwendungsfall H1.

Bei diesem Ablauf sind die folgenden Nachteile zu beobachten:

- Während der manuellen Umplanung (Schritt 2) hat der Planer keine Unterstützung seitens des Feinplanungsver-

fahrens. Lediglich einfache Routinen stellen die Zulässigkeit des Fertigungsplans bei Bedarf her.

- Eine fortgeschrittene algorithmische Unterstützung während der Interaktionen ist nicht ohne eine Neuausführung eines Feinplanungsverfahrens (Schritt 4.) möglich.

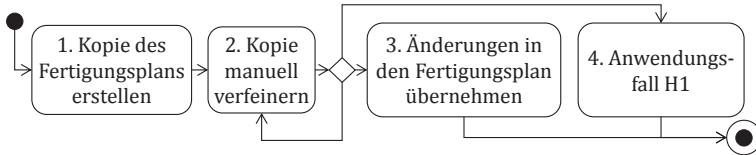


Abbildung 3.4: Anwendungsfall H2

Anwendungsfall H3: Reaktion auf Störung / Eilauftrag

Die Abweichungen vom geplanten Ablauf sind Teil des Tagesgeschäfts. Eilaufträge, Maschinenstörungen oder Materialverspätungen können den aktuellen Fertigungsplan unausführbar bzw. unzulässig machen und eine Kettenreaktion aus Stillständen quer durch die Fertigung verursachen. Deshalb sind verantwortliche Hallenmitarbeiter gezwungen, schnell eine bessere Abarbeitungsreihenfolge für die betroffenen Maschinen zu bestimmen. Falls zu diesem Zeitpunkt ein Planer (s. Abbildung 3.5) und ausreichend Zeit für einen der anderen beiden Anwendungsfälle zur Verfügung stehen, kann das Feinplanungssystem dabei unterstützen. Dies geschieht über den Planer. Andernfalls ist der Hallenmitarbeiter auf eine Entscheidung unter der alleinigen Berücksichtigung des eigenen Verantwortungsbereichs angewiesen.

Der Ablauf in Abbildung 3.5 beleuchtet die Schwächen bei der Realisierung des Regelkreises in der Fertigungssteuerung mittels heutiger Feinplanungssysteme:

- Feinplanungsverfahren sind nicht selbstständig reaktionsfähig. Sie nehmen nur durch Aktionen des Planers am Regelkreis der Fertigungssteuerung teil.

- Eine adäquate Berichtigung des Fertigungsplans unter Betrachtung der kompletten Fertigungssituation ist bei Verwendung von Feinplanungsverfahren nur in den seltensten Fällen möglich. Beispielsweise ist die Fertigung während Nacht- oder Spätschichten auf Entscheidungen vor Ort angewiesen.

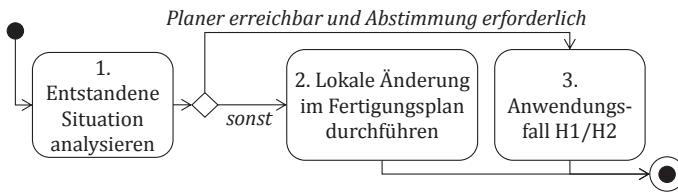


Abbildung 3.5: Reaktion auf Störung / Eilauftrag (Anwendungsfall H3)

Das Verhalten heutiger Feinplanungssysteme weist in allen drei Anwendungsfällen viele Schwächen auf [Weze06, S.206–207]. Diese Systeme sind nicht autonom, erfordern den regelmäßigen Einsatz eines Planers und sind in der praktischen Anwendung nur ein Hilfswerkzeug für die Feinplanung. Das äußert sich auch in deren schlechter Akzeptanz in der Industrie.

3.2 Die gewünschte Rolle der Feinplanungsverfahren in der Fertigungssteuerung

Die zunehmende Vernetzung, die Standardisierung der Interoperabilität und die zunehmende Intelligenz in der Fertigung lösen bestehende langjährige IT-Konzepte ab [BaDi11a]. Unter den Bezeichnungen Industrie 4.0 [KaWH12] sowie Connected Manufacturing [HKSC06] oder Industrial Internet [EvAn12] sind Tendenzen und zukünftige Organisationsformen in der Fertigungssteuerung beschrieben. Direkte Kommunikationsmöglichkeiten zwischen Information bereitstellenden und fordernden Systemen

ersetzen die etablierte Hierarchisierung der fertigungsunterstützenden Systeme.

Abbildung 3.6 zeigt eine moderne Vernetzung von Ressourcen und IT-Systemen. Die Anbindung an ein Netzwerk oder an das Internet erlaubt den Datenaustausch zwischen ERP und den Ressourcen oder zwischen den Ressourcen selbst über standardisierte Schnittstellen [VKBW09]. Intelligente Maschinensteuerungen führen dabei eine beschränkte Reihenfolgenoptimierung der anstehenden Operationen unter Berücksichtigung des Fertigungszustands durch. Beispiel dafür ist die Verschnittoptimierung in der Textilindustrie oder die nahezu rüstoffreie Reihenfolgenbildung von Platinenbestückungsautomaten in der Elektronikfertigung.

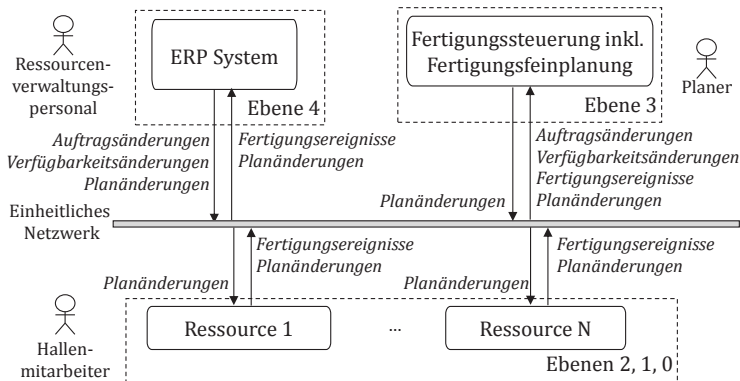


Abbildung 3.6: Vernetzung in der Unternehmens-IT nach Industrie 4.0

Im Gesamteffekt agieren entsprechend der Industrie 4.0 deutlich mehr Akteure innerhalb der Fertigungssteuerung, die durch bidirektionale Abstimmungen untereinander den Fertigungsablauf und -plan beeinflussen können. Eine verteilte selbstorganisierte Fertigungssteuerung weist nach der heutigen Forschung deutliche Schwächen auf (vgl. 2.5.4). Da auch die Verantwortlichen für die Fertigungssteuerung die Oberhand über den Ferti-

gungsablauf behalten wollen, müssen die Abläufe innerhalb der Fertigungssteuerung überdacht werden.

Planänderungen finden nicht mehr nur auf Initiative des Planers statt. Eine höhere Anzahl an Akteuren wird zukünftig involviert (s. Abbildung 3.7). Der Automatisierungsgrad soll gesteigert werden und die Feinplanungsverfahren sollen selbstständiger agieren [Mönc06, S.43–46]. Die Rolle des Menschen in den Abläufen ist das Ergebnis der Analyse vieler Gespräche mit Planern aus unterschiedlichen Fertigungsarten und -branchen, die im Rahmen dieser Arbeit geführt wurden. Die Fertigungsarten reichen von Serienfertigung über Auftragsfertigung bis zur einmaligen kundenspezifischen Auftragsfertigung. Obwohl die befragten Fertigungsverantwortlichen dazu aus unterschiedlichen Branchen wie z. B. Maschinenbau, Elektronikindustrie, Anlagenbau oder Textilindustrie stammen und deren Optimierungsziele sich stark unterscheiden, decken sich ihre Wünsche hinsichtlich des gewünschten Verhaltens und der Unterstützung seitens des Feinplanungssystems. Nachfolgend wird die gewünschte Rolle des Feinplanungssystems in den drei Anwendungsfällen (F1, F2 und F3) in der modernen Fertigungsorganisation erläutert:

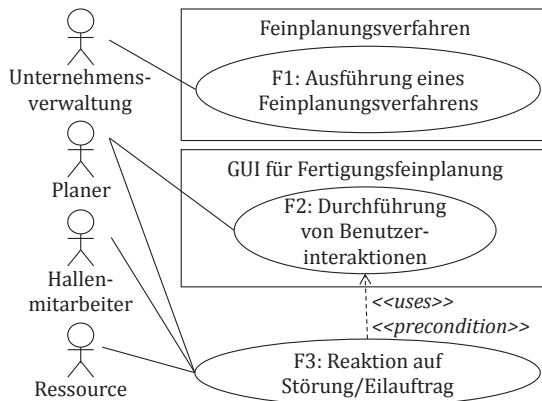


Abbildung 3.7: Anwendungsfälle in zukünftigen Fertigungssteuerungssystemen

Anwendungsfall F1: Ausführung eines Feinplanungsverfahrens

Das Feinplanungsverfahren agiert selbstständig als Dienst und untersucht ununterbrochen den Lösungsraum nach besseren Fertigungsplänen unter Berücksichtigung der eingegebenen Ziele (s. Abbildung 3.8). Es führt nützliche und nachvollziehbare Änderungen im Fertigungsplan durch. Der Fertigungsfortschritt wird in der Feinplanung bei Bedarf mit einbezogen. Auch Reihenfolgenänderungen ausgehend von den Ressourcen und Benutzerinteraktionen fließen ein.

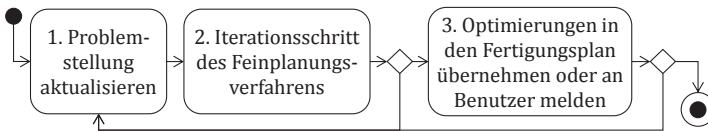


Abbildung 3.8: Anwendungsfall F1

Im Gegensatz zu Anwendungsfall H1 aus dem vorigen Unterkapitel bringt dieser Anwendungsfall mehrere Vorteile für den Planer und die Fertigung:

- Die Ausführung des Feinplanungsverfahrens ist unabhängig vom Planer. Der Planer steuert das Feinplanungsverfahren nur indirekt über seine Zielsetzung oder seine manuellen Interaktionen.
- Da das Feinplanungsverfahren ständig im Hintergrund abläuft, untersucht es deutlich mehr denkbare Fertigungspläne, was sich in der Optimierungsqualität widerspiegelt. Theoretisch wird im Vergleich zu H1 die restlichen 98% der insgesamt verfügbaren Rechenzeit zusätzlich genutzt.
- Das Feinplanungsverfahren führt nur notwendige Änderungen im Fertigungsplan durch und berücksichtigt die Interaktionen der Planer. Der Fertigungsplan bleibt damit stabil.
- Fertigungsverantwortliche und Ressourcen mit Planungsin-telligenz beteiligen sich direkt an der Feinplanung.

Anwendungsfall F2: Durchführung von Benutzerinteraktionen

In Anwendungsfall F2 manipuliert der Planer den Fertigungsplan (Abbildung 3.9). Seine Aktionen sind dabei unmittelbar für die anderen Benutzer sichtbar. Im Hintergrund führt das Feinplanungsverfahren Änderungen durch, die im Einklang mit den getätigten Benutzerinteraktionen sind.

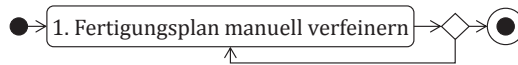


Abbildung 3.9: Anwendungsfall F2

Dieser Ablauf ist deutlich einfacher als der Ablauf von H2 und hat die folgenden Vorteile:

- Der Planer hat während seiner Interaktionen eine Unterstützung von einem im Hintergrund laufenden Feinplanungsverfahren, das nicht nur die Zulässigkeit des Fertigungsplans herstellt, sondern bei Bedarf kleine nachvollziehbare Optimierungen durchführt.
- Der Planer konzentriert sich nur auf die Planänderungen und nicht auf die Bedienung der Algorithmik.

Anwendungsfall F3: Reaktion auf Störung / Eilauftrag

Das Eintreten eines Fertigungsereignisses, das eine dringende Anpassung des Fertigungsplans erfordert, ist bereits durch die permanente Optimierung in F1 abgedeckt. Parallel dazu kann ein Planer oder Fertigungsverantwortlicher gezielt eingreifen (F2), um die Planänderungen zu beeinflussen (s. Abbildung 3.10).

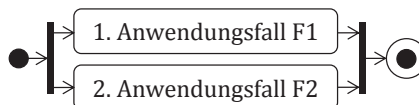


Abbildung 3.10: Anwendungsfall F3

Im Unterschied zu H3 bietet F3 die folgenden Vorteile:

- Die Feinplanungsverfahren sind reaktionsfähig und am Regelkreis der Fertigungssteuerung aktiv beteiligt.
- Die Fertigungsverantwortlichen können bei einem Ereigniseintritt ohne Umweg über einen Planer eine lokale Reihenfolgenänderung durchführen, die in die Problemstellung des Feinplanungsverfahrens einfließt und im Planungslauf sofort berücksichtigt wird.

Wie aus den Anwendungsfällen zu entnehmen ist, wünscht sich der Planer eine gründliche Verbesserung in der Funktionalität der Feinplanungssysteme. Der erste Anwendungsfall schildert den Wunsch nach mehr Autonomie und einem größeren Beitrag von Seiten der Feinplanungsalgorithmen. Der zweite Anwendungsfall schildert die gewünschte Kooperations- und Assistenzfähigkeit der Algorithmik. Aus dem dritten Fall geht die Forderung nach einer adäquaten Reaktionsfähigkeit hervor. Zusammengefasst soll das Feinplanungssystem ein echter Bestandteil des Regelkreises der Fertigungssteuerung werden.

3.3 Anforderungen an ein fortschrittliches Feinplanungssystem

Aus der Gegenüberstellung der Abläufe in der heutigen und der gewünschten Fertigungssteuerung in 3.1 bzw. 3.2 leiten sich die Anforderungen an bestehenden Feinplanungssysteme ab. Die vom aktuellen Stand der Technik (vgl. 2.5) nicht ausreichend erfüllten Anforderungen sind anschließend beschrieben:

Anforderung A1: Nachvollziehbarkeit algorithmischer Planänderungen für den Benutzer

Eine ständig laufende algorithmische Optimierung im Hintergrund soll im auszuführenden Fertigungsplan nachvollziehbare Modifikationen vornehmen. Dadurch wird das Feinplanungsverfahren aus Sicht des Planers verständlich und es baut sich Vertrauen in das Feinplanungsverfahren auf. Nur wenn der Planer der Algorithmik vertraut, wird er autonome Änderungen im auszuführenden Plan akzeptieren.

Anforderung A2: Kollaboration zwischen mehreren Benutzern und Feinplanungsverfahren

Die Planer wollen stets den aktuellen Fertigungsablauf verfolgen können und sich von ihren Aktionen gegenseitig in Kenntnis setzen. Gleichzeitig verlangen sie eine algorithmische Unterstützung. Das Feinplanungsverfahren soll mit initiierten Veränderungen von unterschiedlichen Seiten (Benutzer und andere Feinplanungsverfahren) im selben Fertigungsplan umgehen können (sog. mixed-initiative scheduling, vgl. 2.5.4).

Anforderung A3: Behandeln von Fragmenten des Fertigungsplans abhängig von ihrer Relevanz zum aktuellen Zeitpunkt

Wie ein Planer muss auch das Feinplanungsverfahren entscheiden, welche Probleme zum aktuellen Zeitpunkt bestehen. Besonders Probleme, die den unmittelbar auszuführenden Bereich des Fertigungsplans betreffen, muss das Feinplanungsverfahren mit höherer Priorität behandeln. Zusätzlich darf es die Probleme im Bereich der langfristigen Vorschau nicht außer Acht lassen. Diese Anforderung ist entscheidend für die Reaktionsfähigkeit des Feinplanungssystems in der unmittelbaren Fertigungssteuerung.

Anforderung A4: Durchführen einer angemessenen Anzahl an Änderungen im aktuellen Fertigungsplan

Auf leistungsfähigen Rechnern können Feinplanungsverfahren tausende Fertigungspläne pro Minute vorschlagen, die sich stark

voneinander unterscheiden und deren Übernahme die Planstabilität stark beeinträchtigen würde (vgl. 2.4). Im Zuge der Kooperation mit dem Planer sollen die Feinplanungsverfahren besonders im unmittelbar auszuführenden Fragment wenige Änderungen selbstständig durchführen, um die Nervosität in der Fertigung und bei den Planern zu reduzieren. Im Gegensatz dazu darf das Feinplanungsverfahren im Bereich der langfristigen Vorschau mehr Umplanungsvarianten untersuchen. In Kombination mit Anforderung A3 lässt sich die in Feinplanungsverfahren oft eingesetzte Sperre für Änderungen im unmittelbar auszuführenden Fragment (frozen zone) aufheben.

Anforderung A5: Berücksichtigung der Benutzerinteraktion und Assistenz

Auch wenn die Änderungsinitiativen sowohl vom Planer als auch vom Feinplanungsverfahren ausgehen, hat das Feinplanungsverfahren die Rolle des Unterstützenden. Der Planer verfügt über Wissen, beispielsweise über die aktuelle Fertigungssituation und bevorstehende Aufträge, das in der Problemstellung nicht abbildbar ist (vgl. 2.5.1). Der Scheduler soll deshalb die Änderung des Benutzers bei seiner Umplanung beachten. Auch wenn der Fertigungsplan vor der Interaktion in einem lokalen Optimum war und durch die Interaktion die Optimierungskriterien deutlich verschlechtert wurden, muss das Feinplanungsverfahren versuchen seine Interaktion beizubehalten. Ggf. soll es dem Planer durch zusätzliche Umplanungen assistieren, wenn dies notwendig ist, um die Planzulässigkeit nach den Änderungen des Planers wieder herzustellen [RFWF10].

3.4 Technische Anforderungen

Die Rechnersysteme sind die Ausführungsplattform der Feinplanungsverfahren. Die heute angebotene Rechenleistung eines

leistungsfähigen Rechnersystems reicht aufgrund der kombinatorischen Explosion des Planungsproblems nur für die Erstellung eines kleinen Anteils aller möglichen Fertigungspläne aus (vgl. 2.2). Die fehlende Rechenkapazität ist eine große Behinderung für die algorithmische Feinplanung [Weze06, S.206–207].

Auch wenn man das 50 Jahre alte Mooresche Gesetz [Mack11] zur Verdoppelung der Transistordichte alle 20 Monate berücksichtigt, wird die Rechenleistung in absehbarer Zeit nicht für die optimale Lösung von heutigen Planungsproblemen im für die Praxis typischen Umfang ausreichen. Die heutige Rechenleistung reicht nur für die Erstellung, die Bewertung und den Vergleich eines sehr kleinen Anteils aller möglichen Fertigungspläne in einer für den Planer akzeptablen Zeit. Durch permanent laufende Feinplanungsverfahren kann die komplette verfügbare Rechenleistung verwendet werden.

Der Fertigungszustand und die Umplanungen anderer Planer sollen dem Planer mit minimaler Verzögerung mitgeteilt werden. Falls er den aktuellen Fertigungsplan ändert, erfahren das betroffene Personal sowie die anderen Planer davon. Dieses Konzept entspricht dem seit ca. 50 Jahren angestrebten simultanen Arbeiten mehrerer Benutzer an einem Dokument, in diesem Fall ein Fertigungsplan [Enge68]. Das Konzept unterscheidet sich deutlich vom klassischen Weg über die zyklisch oder manuell angestoßenen Abrufe des Fertigungszustands und die explizite Freigabe der eigenen Umplanungen an die anderen Planer.

Im Grunde genommen beschreiben die hier geschilderten technischen Anforderungen die allgemeinen Erwartungen an die Entwicklung neuer Systeme. Deren Nichterfüllung kann zu fehlender Akzeptanz des Feinplanungssystems oder zu Qualitätseinbußen im Ergebnis führen.

4 Ableitung eines Konzepts für eine Feinplanungsplattform

Die im vorigen Kapitel abgeleiteten Anforderungen zeichnen das Bild eines intelligenten Software-Systems, das den Planer unterstützt. Diese Unterstützung erfordert eine nachvollziehbare Bewertung der Fertigungssituation und eine nachvollziehbare Auswahl der Aktionen in akzeptabler Zeit. Wie in Kapitel 2 festgestellt, mangelt es nicht an leistungsfähigen Feinplanungsverfahren, sondern an deren Kollaboration mit dem Planer. Deshalb wird in diesem Kapitel eine intelligente Feinplanungsplattform konzipiert, die Feinplanungsverfahren lenkt, um die gestellten Anforderungen zu erfüllen.

Unterkapitel 4.1 führt das Gesamtbild der Feinplanungsplattform ein. Anschließend werden ihre einzelnen Aspekte abgeleitet, beginnend mit den Bewertungskomponenten in 4.2 und gefolgt von der Vergleichsprozedur für Fertigungspläne in 4.3. Unterkapitel 4.4 zeigt, wie die Feinplanungsplattform Feinplanungsverfahren bei der Optimierung unterstützen kann. Die Koordination von gleichzeitig initiierten Umplanungen seitens des Feinplanungsverfahrens und des Planers wird in 4.5 beschrieben.

4.1 Gesamtbild der Feinplanungsplattform

Die in 3.3 ermittelten Anforderungen legen den Schwerpunkt auf die Auswahl der Fertigungspläne und die dafür notwendige Bewertung. Diese Funktionen sind für einen Großteil der Feinplanungsverfahren notwendig aber nicht verfahrensspezifisch. Andere Funktionen, die allein das Bedienungserlebnis betreffen,

wie beispielsweise die Kollaboration mehrerer Planer, sind auch außerhalb der eigentlichen Optimierung realisierbar.

Nicht verfahrensspezifische Funktionalität wird häufig in Feinplanungsplattformen ausgelagert [Casa01, DoGV98, KLMM02, LLDV07]. Im Forschungsbereich liegt der Schwerpunkt auf der Untersuchung von Feinplanungsverfahren und Wiederholbarkeit der Ergebnisse unter Verwendung von deterministischen Pseudo-Zufallszahlgeneratoren [WBKK10]. Andere Feinplanungsplattformen konzentrieren sich auf den Planer und seine Arbeit mit verschiedenen Szenarien und Was-wäre-wenn-Analyse [SMM090] und bieten eine mächtige Benutzeroberfläche für Interaktionen.

Die gemeinsame Funktionalität führt auch hier zur Entscheidung, sie in eine Feinplanungsplattform auszulagern. Somit steht sie allen an der Fertigungsplanung Beteiligten zur Verfügung (s. Abbildung 4.1). Die Feinplanungsplattform integriert die beiden Hauptfunktionen Bewertung und Vergleich (für die Auswahl) von Fertigungsplänen, übernimmt die Kommunikation zwischen den Teilnehmern sowie die Zusammenführung und die Verteilung der Änderungen in den Fertigungsplänen. Die Feinplanungsverfahren sind austauschbar, kombinierbar und können von gemeinsam verwendeten Funktionen wie Bewertung und Vergleich von Fertigungsplänen profitieren.

Die Planer können die Bewertung und den Vergleich somit auch unabhängig von der Algorithmik verwenden, wenn sie einen Fertigungsplan nach nicht für die Planauswahl relevanten Kriterien auswerten wollen. Auch können sie an einem gemeinsamen Fertigungsplan arbeiten, ohne die algorithmische Optimierung in Anspruch nehmen zu müssen. Ferner soll die Feinplanungsplattform relevante Meldungen über den Fertigungsfortschritt überwachen und relevante Änderungen im Fertigungsplan unmittelbar zusammenführen. Außerdem verteilt sie den aktuellen Fertigungsplan.

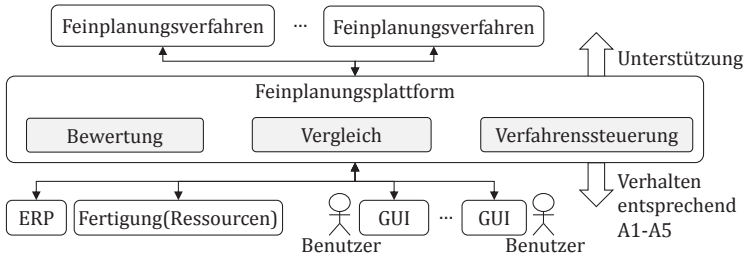


Abbildung 4.1: Grobe Übersicht der Feinplanungsplattform

Wie aus den Anwendungsfällen zum gewünschten Verhalten der Algorithmik (vgl. 3.2) hervorgeht, will sich der Benutzer auf manuelle Umplanungen konzentrieren und erwartet von der Algorithmik eine höhere Selbstständigkeit. Deshalb muss die Feinplanungsplattform die Ausführung und die Steuerung der Feinplanungsverfahren übernehmen.

Um eine Zusammenarbeit von mehreren Benutzern an einem Fertigungsplan in Echtzeit anzubieten, verteilt die Feinplanungsplattform jede Änderung im Fertigungsplan an alle Betroffenen unmittelbar nach Eintritt unabhängig von ihrer Herkunft. So sehen die Planer gegenseitig ihre Umplanungen in Echtzeit. Diese neuartige Eigenschaft in Feinplanungssystemen ist für die Mehrbenutzer-Bedienung unumgänglich.

4.2 Aspekte der Bewertungskomponenten

Die Bewertungskomponente muss die Berechnung notwendiger Bewertungskriterien für den Vergleich der Fertigungspläne durchführen. Neben den Kennzahlen für Termintreue und Rüstzeiten (vgl. 2.2.3) werden weitere für die Praxis relevante Kennzahlen wie die durchschnittliche Durchlaufzeit der Aufträge ermittelt. Die Durchlaufzeit wird als $\emptyset DLZ := \frac{1}{n} \sum (C_j - S_j)$ oder als $\frac{1}{n} \sum F_j$ abhängig von der Fertigungsart berechnet, wobei

die zweite Variante eigentlich eine menschenfreundliche Variante von $\sum C_j$ ist.

Anders als ein Feinplanungsverfahren interessiert sich ein Planer für deutlich mehr Bewertungskriterien wie Rüstgrad, Beleggrad, Belegnutzgrad, Auslastung etc. [Iso14, Vdma09], deren Bewertungswert einen deutlich benutzerfreundlicheren Wertebereich hat als die absoluten oder gewichteten Summen der Zielfunktionen in der Forschung. Diese Bewertungskriterien beeinflussen den Planer bei seinen Umplanungen und müssen deshalb auch von der Bewertungskomponente bereitgestellt werden.

Ein wesentlicher Teil der Bewertungskomponente besteht in der Erweiterung der oben genannten klassischen Bewertungskriterien sowie der Einführung neuer Bewertungskriterien, um einen Vergleich von Fertigungsplänen gemäß der ermittelten Anforderungen zu ermöglichen. Das Feinplanungsproblem ist ein komplexes multikriterielles Optimierungsproblem und jedes zusätzlich berücksichtigte Kriterium erhöht den Komplexitätsgrad. Die unterschiedlichen Aspekte der Anforderungen werden deshalb nachfolgend, soweit möglich, in die klassischen Bewertungskriterien integriert.

4.2.1 Integration des aktuellen Zeitpunkts

Die herkömmlichen Bewertungskriterien sind theoretischer oder statistischer Natur und die Dispositionsobjekte fließen unabhängig von ihrer Relevanz für die aktuelle Situation ein. Ereignisse in einer laufenden Fertigung verursachen Unstimmigkeiten im Fertigungsplan, die aus aktueller Sicht unterschiedliche Bedeutung besitzen. Dieser vernachlässigte Aspekt wird hier in die Bewertung integriert.

Anhand eines Beispiels in Abbildung 4.2 wird die Bedeutung des aktuellen Zeitpunkts t_0 erläutert. Der dargestellte Fertigungsplan

beinhaltet einige verspätete Operationen sowie reihenfolgenabhängige Rüstzeiten. Feinplanungsverfahren haben eine deutlich längere verfügbare Rechenzeit für die Optimierung von Fragmenten des Fertigungsplans, die in der Zukunft auszuführen sind, als für die Optimierung von Fragmenten, die gerade den Fertigungsablauf steuern.

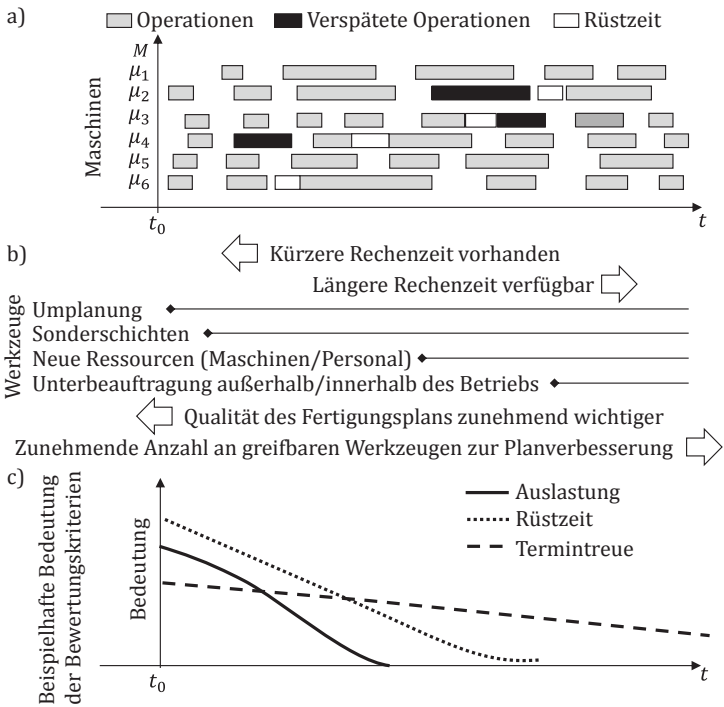


Abbildung 4.2: Der Fertigungsplan aus Sicht des aktuellen Zeitpunkts t_0 nach [DiBa12]

Die algorithmische Optimierung ist nur ein Mittel, um den Fertigungsplan zu verbessern. Einem Planer stehen weitere Werkzeuge zur Verfügung. Er kann die Bearbeitung bestimmter Operationen oder ganzer Aufträge innerhalb oder außerhalb des Betriebs auslagern. Auch kann er zusätzliche Schichten kurzzeitig einfüh-

ren oder sogar neue Fertigungsanlagen oder Personal beschaffen. Je näher ein Fragment des Fertigungsplans am aktuellen Zeitpunkt liegt, desto weniger Werkzeuge sind nutzbar, wie Abbildung 4.2b zeigt. Das unmittelbar auszuführende Fragment ist nur mittels Planänderung zu verbessern.

Einen weiteren Punkt bilden die unterschiedlichen verfolgten Optimierungsziele abhängig vom aktuellen Zeitpunkt (s. Abbildung 4.2c). In der langfristigen Planung ist die Termineinhaltung ein primäres Ziel. Die Auslastung von Maschinen und Personal beginnt erst in der mittelfristigen Planung eine Rolle zu spielen. In der unmittelbaren Fertigungssteuerung dagegen sind andere fertigungsspezifische Kennzahlen wie die reihenfolgenabhängige Rüstzeit relevant. Die Optimierung dieser Kennzahlen führt oft zu großen finanziellen Ersparnissen, weshalb Planer und Verantwortliche bei einer Verbesserung dieser Kriterien oft auch zusätzliche Terminverletzungen oder die suboptimale Auslastung von Ressourcen in Kauf nehmen.

Die Relevanzkurven aus Abbildung 4.2c können mathematisch in Form einer Gewichtungsfunktion abgebildet werden. Üblicherweise dienen Gewichtungsfunktionen zur Ermittlung einer Gewichtung für die Bewertung der einzelnen betroffenen Objekte (vgl. 2.2.3). Zum Beispiel werden die einzelnen Verspätungen der Aufträge T_j mit ihrer Gewichtung w_j multipliziert und so entsteht die gewichtete Summe der Verspätungen $\sum T_j$. Die Relevanzkurven dagegen drücken die Relevanz eines Bewertungskriteriums zu einem Zeitpunkt aus, weshalb sie als Gewichtungsfunktion die einzelnen Zeitstempel und nicht deren Differenz gewichten müssen. Anstelle der einfachen Differenz $d_j - C_j$ zur Berechnung von T_j für einen verspäteten Auftrag j kann das entstandene Volumen zwischen der Relevanzkurve $g(t)$ und den betroffenen Zeitpunkten $\int_{a_j}^{C_j} g(t)dt$ verwendet werden. Das Integral bringt deutliche Vorteile gegenüber der einfachen Gewichtung der Differenz von

Zeitstempeln. Eine Verdopplung dieser Differenz führt nicht zu einer Verdopplung der über das Intergral gewichteten Differenz. Die Gewichtung ist präzise steuerbar.

Die so berechnete Verspätung ist in Abbildung 4.3 anhand eines Beispiels mit zwei Aufträgen im Vergleich mit der klassischen Variante zu sehen. Die zeitbehaftete Verspätung eines Auftrags hängt von der Relevanz des betroffenen Zeitintervalls ab, da hauptsächlich eine Umplanung in diesem Intervall diese Verspätung reduzieren würde. Das Beispiel aus Abbildung 4.3 zeigt, dass es darüber hinaus möglich ist, eine kürzere Verspätung je nach Relevanz zum aktuellen Zeitpunkt höher zu bewerten, da dringender Handlungsbedarf besteht.

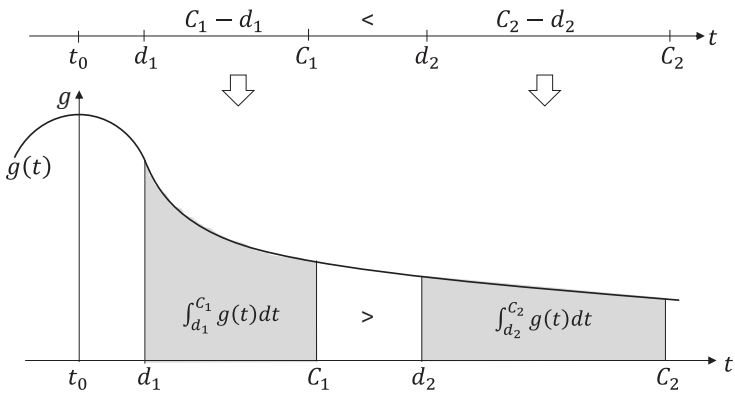


Abbildung 4.3: Gewichtung der Verspätung nach klassischer und zeitbehafteter Art [DiBa12]

Viele Bewertungskriterien für die Termintreue können mittels Ersetzung der klassischen Varianten $T_j = \max\{d_j - C_j, 0\}$ und $E_j = \max\{C_j - d_j, 0\}$ durch die zeitbehafteten Varianten $\max\{\int_{d_j}^{C_j} g(t) dt, 0\}$ bzw. $\max\{\int_{C_j}^{d_j} g(t) dt, 0\}$ umgeschrieben werden. Tabelle 4.1 gibt einen Auszug der Zielfunktionen aus Abschnitt 2.2.3 zusammen mit ihren zeitbehafteten Varianten. Die

Standardabweichung σ_T und Varianz σ_T^2 der Verspätung sind ebenfalls konvertierbar. Weiterhin lassen sich andere Bewertungskriterien umschreiben, die auf Basis einer Differenz zwischen zwei Zeitwerten berechnet werden. Beispiele dafür sind die durchschnittliche Durchlaufzeit und die Summe der Rüstzeiten TST . Bei der Summe der Rüstkosten ist die direkte Integration von $g(t)$ nicht möglich. Weil die Rüstkosten in bestimmten Zeitintervallen anfallen, werden sie pro o_{ji} mit $\int_{YS_{ji}}^{YC_{ji}} g(t)dt / (YC_{ji} - YS_{ji})$ gewichtet. Diese Gewichtung drückt die normierte Relevanz des Zeitintervalls (YS_{ji}, YC_{ji}) aus. Somit sind auch andere Kennzahlen ohne direkte Verwendung von Zeitstempeln konvertierbar.

Tabelle 4.1: Erweiterung klassischer Zielfunktionen aus 2.2.3 um die Relevanz zum aktuellen Zeitpunkt

<i>Zielfunktion</i>	<i>klassisch</i>	<i>zeitbehaftet</i>
ΣE_j	$\Sigma \max\{d_j - C_j, 0\}$	$\Sigma \max\left\{\int_{C_j}^{d_j} g(t)dt, 0\right\}$
ΣT_j	$\Sigma \max\{C_j - d_j, 0\}$	$\Sigma \max\left\{\int_{d_j}^{C_j} g(t)dt, 0\right\}$
$1/n \Sigma F_j$	$1/n \Sigma (C_j - r_j)$	$1/n \Sigma \int_{r_j}^{C_j} g(t)dt$
$\emptyset DLZ$	$1/n \Sigma (C_j - S_j)$	$1/n \Sigma \int_{S_j}^{C_j} g(t)dt$
TST	$\Sigma_{j,i} (YC_{ji} - YS_{ji})$	$\Sigma_{j,i} \int_{YS_{ji}}^{YC_{ji}} g(t)dt$
TSC	$\Sigma_{j,i} YR_{ji}$	$\Sigma_{j,i} \frac{\int_{YS_{ji}}^{YC_{ji}} g(t)dt \cdot YR_{ji}}{YC_{ji} - YS_{ji}}$

Fast alle nach [Iso14, Vdma09] standardisierten und für die Feinplanung relevanten Bewertungskennzahlen sind auf diese Art modifizierbar. Diese Definitionsart der zeitbehafteten Varianten ergibt bei $g(t) \equiv 1$ eine Übereinstimmung mit den klassischen Kennzahlen. Da die Bewertungskriterien sich in ihrer Priorisierung entlang der Zeitachse unterscheiden, ist es notwendig unter-

schiedliche Relevanzfunktionen $g(t)$ pro Kennzahl anzuwenden. Die unterschiedliche Relevanz der Kriterien (wie z. B. in Abbildung 4.2c) ist somit modellierbar und Anforderung A3 ist erfüllt.

4.2.2 Quantifizierung der Planstabilität

Wie in Abschnitt 2.4.4 erläutert, existieren bereits etablierte Methoden zur Quantifizierung der Planstabilität bzw. der verursachten Nervosität als das gegenteilige Bewertungskriterium. Wie aus der typischen Berechnung der Planstabilität als $\sum_{j,i} |S_{ji} - S_{ji}^{\#}|$ ersichtlich ist, handelt es sich um ein relatives Bewertungskriterium, das einen bestimmten Fertigungsplan nur in der Gegenüberstellung mit einem anderen Fertigungsplan $P_{\#}$ als Basis (Referenzplan) quantifizieren kann. Als Referenzplan dient oft der dem Benutzer zum aktuellen Zeitpunkt vorliegende Fertigungsplan.

Die Bedeutung des aktuellen Zeitpunkts in der Evaluierung des Fertigungsplans wurde im Zusammenhang mit der Quantifizierung der Planstabilität schon von [PfKM06, RaFK04] erkannt (vgl. 2.4.4). Anstatt die Relevanzfunktion in die Berechnung zu integrieren wurden die Differenzen der Zeitstempel einfach gewichtet. Die verwendeten Gewichtungsfunktionen haben als Eingangsparmeter $(S_{ji} + S_{ji}^{\#})/2$ bzw. $\min\{S_{ji}, S_{ji}^{\#}\}$ und zeigten einen positiven Einfluss auf die Simulationsergebnisse. Die Art der Gewichtungsfunktionen wurde jedoch zu Demonstrationszwecken gewählt und die praktische Relevanz blieb im Hintergrund.

Wie im vorigen Abschnitt kann die Formel $\sum_{j,i} \left| \int_{S_{ji}}^{S_{ji}^{\#}} g(t) dt \right|$ die klassische Planstabilität $\sum_{j,i} |S_{ji} - S_{ji}^{\#}|$ ersetzen. Ein weiterer Aspekt der Planstabilität sind die geplanten Maschinen μ_{ji} bei mehreren zulässigen Maschinen pro o_{ji} ($|M_{ji}| > 1$) in der Problemstellung (Flexible Shop Scheduling Probleme). Während die obige Formel den stattgefundenen Maschinenwechsel ignoriert,

wurden in der Literatur Kennzahlen für die Planstabilität vorgeschlagen, die allein auf der Anzahl geänderter Maschinen basieren und die Ausführungszeiten ignorieren [AlAz03]. Da in der Praxis meistens mehrere Maschinen für eine Operation zur Auswahl stehen, muss der Maschinenwechsel aus den nachfolgenden Gründen in der Planstabilität mitbetrachtet werden.

Auch wenn die eingeplanten Ausführungszeiten unberührt bleiben, kann ein Wechsel der eingeplanten Maschinen (und Ressourcen) zu Unruhe in der Fertigung und Planung führen. Einerseits muss das Personal Vorbereitungsarbeiten vor dem geplanten Startzeitpunkt durchführen, die je nach Fertigungsart, eingesetzten Ressourcen und der innerbetrieblichen Logistik einer gewissen Vorlaufzeit bedürfen. In der Planung dagegen bilden die Planer oft manuell günstige Rüstfolgen, die auch so bestehen bleiben sollen. Der Ressourcenwechsel von o_{ji} kann deshalb als

$\int_{\min\{s_{ji}, s_{ji}^{\#}\}}^{\min\{s_{ji}, s_{ji}^{\#}\}} g(t) dt$ quantifiziert werden, wobei v_{ji} die notwendige Vorlaufzeit der Operation o_{ji} , darstellt.

Als nächstes muss der Ressourcenwechsel (RW) in die gewichtete Intervallsumme eingebracht werden. Bei Kombination der beiden Summen entsteht für die Quantifizierung der Planstabilität

die Formel $\vartheta(P) := \sum_{j,i} \int_{\min\{s_{ji}, s_{ji}^{\#}\} - vx_{ji}}^{\max\{s_{ji}, s_{ji}^{\#}\}} g(t) dt$ mit $vx_{ji} := \begin{cases} v_{ji}: RW \\ 0: sonst \end{cases}$.

Dieses Kriterium quantifiziert die Unterschiede zwischen zwei Fertigungsplänen. Für die Entscheidung über eine Übernahme als auszuführender Fertigungsplan ist diese Information ungenügend und es müssen weitere Aspekte mit einbezogen werden.

4.2.3 Akkumulierte Planstabilität (Shop Floor Nervousness)

Das Kriterium für Planstabilität quantifiziert die potentiell entstehende Nervosität im Falle, dass der Referenzplan als aktuell auszuführender Plan durch den zu bewertenden Plan ersetzt wird. Es berücksichtigt nicht, ob in der Vergangenheit stattgefundene Übernahmen bereits eine gewisse Nervosität verursacht haben. Dabei können viele Übernahmen in einem kurzen Zeitintervall aufsummiert zu einer merklichen Nervosität führen.

Die Problematik wird anhand des Beispiels in Abbildung 4.4 erläutert, die die entstandene Nervosität bei der Übernahme von k Fertigungsplänen zeigt. k ist der Einfachheit halber gerade. Die Länge der Pfeile zwischen den Fertigungsplänen P_0 bis P_k repräsentiert die entstandene Nervosität bei der Übernahme. Die Pfeile mit durchgängiger Linie repräsentieren die Übernahme von jedem Fertigungsplan in der Reihe und solche mit gestrichelter Linie das Überspringen von jedem zweiten beim Übernehmen. Da jede Übernahme zu einer Nervosität führt, akkumuliert sich eine höhere Nervosität bei häufigeren Übernahmen.

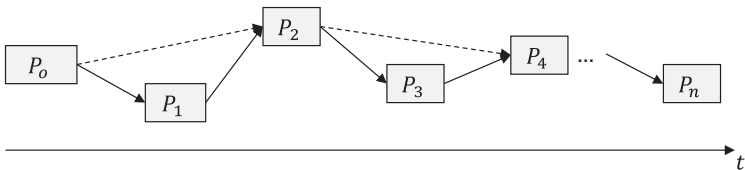


Abbildung 4.4: Grafische Darstellung der Nervosität bei der Übernahme von Fertigungsplänen

Dieser Effekt wird nachfolgend mathematisch bestätigt. Die Zielfunktion für die Berechnung der Planstabilität $\vartheta(P_i)$ für den Fertigungsplan P_i kann auch als eine Art „Abstandsmaß“ $\vartheta^a(P_i, P_{i-1})$ zwischen P_i und der vorher übernommenen P_{i-1} angesehen werden. Die Funktion PS ist wie im vorletzten Ab-

schnitt definiert (sowie in den meisten Ausführungen in der Literatur) symmetrisch und erfüllt die Dreiecksungleichung. Für alle $0 < i < k$ gilt $\vartheta^a(P_i, P_{i-1}) + \vartheta^a(P_{i+1}, P_i) \geq \vartheta^a(P_{i+1}, P_{i-1})$ und somit auch $\sum_{i=1}^k \vartheta^a(P_i, P_{i-1}) \geq \sum_{i=1}^{k/2} \vartheta^a(P_{i,2}, P_{i,2-2})$.

Um das Problem mit den vorigen Übernahmen zu umgehen, schlagen manche Varianten von ϑ die Verwendung ein und desselben Fertigungsplans als Referenzplan $P_{\#}$ vor (vgl. 2.4.4). Das schränkt den Abstand zu $P_{\#}$ ein, kann aber zu bis zu einem doppelten Abstand zum Vorgänger als Nebeneffekt haben und damit zu einer hohen realen akkumulierten Nervosität führen.

Um die Akkumulierung der Nervosität bei der Planübernahme mit zu betrachten (Anforderung A4), muss sie vorher quantifiziert werden. Unmittelbar stattgefundenene Übernahmen haben einen höheren Einfluss auf die Nervosität und sollen auch eine deutlich höhere Gewichtung bekommen (s. Beispiel in Abbildung 4.5). Denn der Planer sowie andere involvierte Personen müssen sich nach einer Übernahme im neuen Fertigungsplan wieder finden. Übernahmen in der Vergangenheit dagegen haben eine deutlich geringere Auswirkung auf die Gesamtnervosität, da die übernommenen Pläne teils schon umgesetzt wurden und sich die Planer dort eingearbeitet haben.

Die akkumulierte Nervosität kann als gewichtete Summe der verursachten Nervosität der vergangenen Übernahmen berechnet werden. Eine weitere Variante von g benutzt die Übernahmezeitpunkte u_i zur Ermittlung der Gewichtung. In Folge dessen wird für die akkumulierte Nervosität bzw. Planstabilität bei der Übernahme von P_k durch die folgende Formel definiert:

$$\theta(P_k) := \sum_{i=1}^k (g^u(u_i) \cdot \vartheta^a(P_i, P_{i-1}))$$

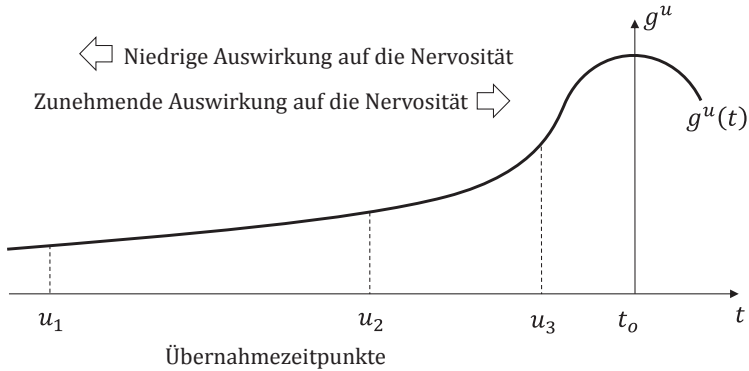


Abbildung 4.5: Schematische Gewichtung der entstandenen Nervosität für drei stattgefundene Übernahmen

Diese Kennzahl liefert eine Entscheidungsgrundlage für zukünftige Übernahmen und auch dafür, ob und inwieweit eine Planübernahme in der aktuellen Situation aus Sicht der Planer und Fertigungsverantwortlichen erwünscht oder zu verkräften ist.

4.2.4 Einhaltung von Benutzerinteraktionen

Die Planer führen aus verschiedenen Anlässen manuelle Plananpassungen durch (vgl. 2.5). Da die Planer über weit mehr Kenntnis über die Fertigung verfügen als im Feinplanungsverfahren modelliert, sind ihre Interaktionen nicht immer im Einklang mit der algorithmischen Feinplanung. Obwohl die Interaktionen aus menschlicher Sicht eine Verbesserung bringen, können sie zu einer Verschlechterung des Zielfunktionswerts führen, da die menschliche Bewertungsart oft nicht algorithmisch nachbildbar ist.

Die Interaktionen sind der Beitrag des Planers in der Planerstellung. Es ist bei Vorschlägen seitens des Feinplanungsverfahrens besonders wichtig, dass dieser Beitrag möglichst bestehen bleibt (Anforderung A5). Abbildung 4.6 stellt ein Fragment eines Ferti-

gungsplans $P_{\#}$ dar, das in den Vorschlägen P_1 und P_2 leicht modifiziert enthalten ist. Aus Sicht der Kennzahl für Planstabilität ist P_2 näher an $P_{\#}$. Jedoch ist die vom Planer verschobene Operation o_{12} (in grau) umdisponiert. Hier wiegt das Interesse des Planers an der Beibehaltung seines Beitrags stärker als die Sicherstellung der Planstabilität.

Auch Umplanungen von intelligenten Ressourcen (vgl. 3.2) sind wie Benutzerinteraktionen zu behandeln. Die Ressourcen kennen ihr internes Modell und können z. B. sehr günstige Rüstreihenfolgen bilden oder Tätigkeiten energieeffizient bündeln. Für ein Feinplanungsverfahren sind diese Planänderungen ein weiterer externer Eingriff, der nach Möglichkeit zu berücksichtigen ist.

Aus Abbildung 4.6 wird ebenfalls deutlich, dass die Einhaltung (bzw. die Verletzung) der Benutzerinteraktion als Variante der Quantifizierung der Planstabilität (bzw. Nervosität) angesehen werden kann, die sich nur auf die vom Benutzer geänderte Operationen erstreckt. Wie für die Planstabilität sollen auch für die Einhaltung der Benutzerinteraktionen dieselben Aspekte der betroffenen Operationen in neuen Vorschlägen enthalten bleiben, nämlich die Planzeiten S_{ji} und C_{ji} , die ausführenden Maschinen und die zusätzlichen Ressourcen. Die Formel für die Einhaltung der Benutzerinteraktionen muss einen weiteren interaktionsspezifischen Aspekt integrieren, dessen Bedeutung nachfolgend erklärt wird.

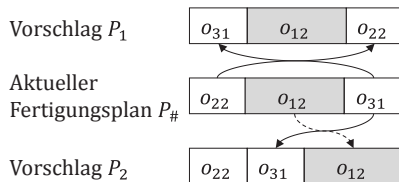


Abbildung 4.6: Planstabilität und Einhaltung der Benutzerinteraktion auf o_{12} in zwei Planvorschlägen

Oft führt der Planer mehrere einzelne Interaktionen durch, um eine beabsichtigte Planänderung komplett durchzuführen. Während der Ausführung der Interaktion kann der Fertigungsplan auch kurzzeitig unzulässig werden und sein Zielfunktionswert kann sich temporär reduzieren. Schaltet sich ein Feinplanungsverfahren in dieser Zeit ein, besteht die Gefahr, dass die gerade getätigte Interaktion zurückgesetzt wird, da dies aus algorithmischer Sicht meistens der kürzeste Weg zu einer Planverbesserung und Wiederherstellung der Planzulässigkeit ist. Diese Gefahr ist bei einem im lokalen Optimum liegenden Ausgangsfertigungsplan besonders groß. Erst wenn der Anwender eine gewisse Anzahl an Umplanungen durchgeführt hat, würde sich der Algorithmus mit höherer Wahrscheinlichkeit eine andere Lösung suchen.

Die algorithmische Optimierung soll die Wünsche der Planer beachten, auch dann wenn sie zu einer Verschlechterung des Zielfunktionswerts führen. Besonders kurz zuvor durchgeführte Interaktionen sollen erhalten bleiben, damit der Planer alle seine beabsichtigten Interaktionen ungehindert durchführen kann. Vor längerer Zeit durchgeführte Interaktionen sind aus mehreren Gründen niedriger zu gewichten. In dynamischen Shop Scheduling Problemen wechselt die Problemstellung ständig: Neue Aufträge kommen hinzu, andere werden fertiggestellt oder es stehen aufgrund geänderter Schichtpläne andere Kapazitäten zur Verfügung (vgl. 2.4).

Abbildung 4.7 zeigt, wie sich die Auftragsmenge erneuert. Bei der Durchführung einer Interaktion zum Zeitpunkt t_1 kannte der Benutzer nur ca. 25% der Aufträge in t_0 . Auch wenn sich die Problemstellung nur geringfügig ändert, kann sich die Planungsstrategie des Benutzers stark verändern. Auch das Gedächtnis als subjektiver Faktor ist hier involviert. An ältere Interaktionen erinnern sich die Benutzer weniger als an jüngere.

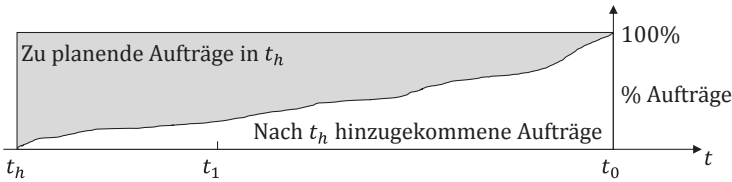


Abbildung 4.7: Erneuerung der Auftragsmenge entlang der Zeit

Eine zusätzliche Gewichtungsfunktion wird benötigt, welche die Interaktionen anhand ihres Alters gewichten soll. Abbildung 4.8 zeigt die Gewichtung von drei Interaktionen. Die skizzierte Gewichtungsfunktion unterscheidet sich unwesentlich von der aus Abbildung 4.5 und spiegelt die Kurve aus Abbildung 4.3. Jedoch dienen alle drei Gewichtungsfunktionen einem anderen Zweck und unterscheiden sich in der Ausführung grundsätzlich voneinander.

Wie auch bei der Planstabilität, spielt die Relevanz zum aktuellen Zeitpunkt eine Rolle, jedoch aus anderen Gründen. Interaktionen mit höherer Relevanz zum aktuellen Zeitpunkt müssen entsprechend hoch gewichtet werden. Die algorithmische Optimierung kann außerdem in der Lage sein, in akzeptabler Zeit einen zulässigen Fertigungsplan zu erzeugen, der alle getätigten Interaktionen beibehält. Der Planer hat deutlich mehr Zeit neue Interaktionen zu tätigen, die die langfristige Planung betreffen, und kann einige von der Optimierung zurückgesetzte Interaktionen erneut oder anders ausführen. Anders sieht es mit Interaktionen aus, welche die unmittelbare Fertigungssteuerung betreffen. Der Planer muss sich darauf verlassen, dass der Algorithmus diese Interaktionen beibehält. Auch sind dort algorithmische Ressourcenwechsel kritischer zu betrachten, weil sie vom Planer optimierte Operationsbündel (batches) oder günstige Rüstreihenfolgen zerbricht.

Für $B_k \in B$ als die Menge der Benutzerinteraktionen, die jeweils zu einer o_{ji} verweisen, b_k als der Interaktionszeitpunkt von B_k , ergibt sich die folgende Formel:

$$\vartheta^b(P) := \sum_{B_k \in B} \left(g^b(b_k) \cdot \int_{\min\{s_{ji}, s_{ji}^k\} - vx_{ji}}^{\max\{s_{ji}, s_{ji}^k\}} g(t) dt \right), \quad vx_{ji} := \begin{cases} v_{ji}: RW \\ 0: sonst \end{cases}$$

Im Wesentlichen unterscheiden sich diese Formeln alleine anhand der Referenzoperationen. Bei der Quantifizierung der Planstabilität wird der aktuelle Fertigungsplan als Referenz genommen. Bei der Quantifizierung der Beibehaltung der Benutzerinteraktionen muss die Bewertung ein Protokoll der getätigten Interaktionen führen, um sie als Referenz nutzen zu können.

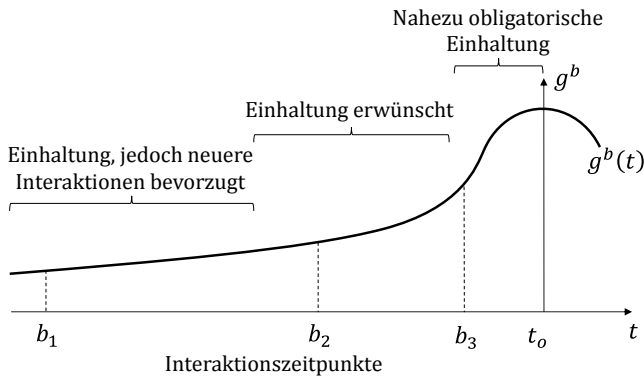


Abbildung 4.8: Schematische Gewichtung der Benutzerinteraktionen nach den Interaktionszeitpunkten

Dieses Kriterium strebt an, die teils subjektiv empfundene Berücksichtigung der Interaktion seitens der algorithmischen Optimierung zu quantifizieren. In der Berechnung fließen identifizierte Aspekte wie das Alter einer Interaktion und die Relevanz der betroffenen Operationen zum aktuellen Zeitpunkt. Im Gegensatz zur Quantifizierung der Planstabilität ist hier keine nachträgliche Akkumulierung der Kriteriumswerte der vergangenen Übernahmen erforderlich. Der Algorithmus darf nicht behindert werden, in weiteren Optimierungsschritten die Einhaltung der Interaktionen zu verbessern.

4.2.5 Bewertung von Fragmenten des Fertigungsplans

Nach Anforderung A1 aus Unterkapitel 3.3 sollen Feinplanungsverfahren nachvollziehbare Änderungen im Plan durchführen. Der Benutzer erwartet, dass ein Optimierer ähnlich wie ein Mensch verbesserungsbedürftige Fragmente des Fertigungsplans aufspürt und optimiert. Das ist bei den stochastischen Feinplanungsverfahren keine Selbstverständlichkeit und selbst intelligente deterministische Algorithmen sind schwer nachvollziehbar (vgl. 2.5). Die Bewertungskomponente muss den Fertigungsplan erst in Fragmente aufteilen, diese Fragmente bewerten und deren Bewertungen einander gegenüberstellen, um die Fragmente mit einem hohen Optimierungspotential auszufiltern.

Im Allgemeinen besteht ein Fragment des Fertigungsplans aus einer oder mehreren Operationen. Es kann kein Fragment ohne Operationen existieren, weil ein leerer Fertigungsplan keine Dispositionsobjekte (Aufträge und Operationen) besitzt und deshalb nicht optimierbar ist. Die atomaren Fragmente sind die Operationen. Die Mehrheit der Bewertungskriterien iterieren entweder über diese atomaren Fragmente oder über die Aufträge, die auch als aus mehreren Operationen bestehende Fragmente angesehen werden können. Implizit bekommt jedes Fragment eine Bewertung je nach Zielfunktion, wie z. B. die Verspätung T_j oder die Terminabweichung L_j für das Fragment Auftrag j oder die Rüstdauer ($YC_{ji} - YS_{ji}$) des Fragments o_{ji} .

Die Operationen können auch anhand unterschiedlicher Fragmentierungsmerkmale einem Fragment zugewiesen werden. Je nach Bedarf können die Operationen im gesamten Fertigungsplan anhand ihrer Ausführungswoche oder des erstellten Ausgangsmaterials aufgeteilt werden (s. Abbildung 4.9). Dabei kann die Fragmentierung hierarchisch erfolgen. Zum Beispiel werden die

Operationen zuerst nach Ausführungswoche fragmentiert und anschließend nach Art der eingesetzten Maschine.

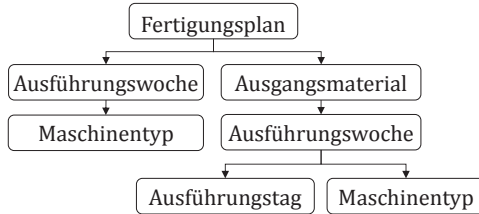


Abbildung 4.9: Beispielhafte Nachschaltung von Fragmentierungsmerkmalen

Die Fragmentierung nach z. B. Ausführungswoche ist nicht immer eindeutig, wenn das Ausführungsintervall einer Operation sich über mehrere Wochen erstreckt. Die Operation wird mehreren Fragmenten zugeordnet und die entstandenen Fragmente sind folglich nicht disjunkt. Eine nicht disjunkte Fragmentierung entsteht auch bei der parallelen Anwendung mehrerer Fragmentierungskriterien. Abbildung 4.10 zeigt eine beispielhafte Fragmentierung des Fertigungsplans nach Ausführungswoche und Maschinentyp. Es ist ersichtlich, dass die einzelnen verspäteten Aufträge (Repräsentierung durch die Menge ihrer Operationen) auch in mehreren Fragmenten vorkommen. Deshalb entspricht $\sum T_j$ des gesamten Fertigungsplans nicht der Summe der $\sum T_j$ der einzelnen Fragmente.

Ziel der Fragmentierung ist es, dem Feinplanungsverfahren einen Hinweis bereitzustellen, wo der Fertigungsplan Optimierungspotential besitzt. Unterschiedliche Fragmentierungen sind für unterschiedliche Optimierungsziele relevant. Um die Gesamtverspätung oder die durchschnittliche Durchlaufzeit zu reduzieren, ist es z. B. notwendig, den Fertigungsplan nach Auftragszugehörigkeit zu fragmentieren. Diese Fragmentierung ist schon implizit in der Modellierung des Shop Scheduling Problem vorhanden und deshalb leicht erzielbar. Anhand der Bewertung der einzelnen Kriterien kann das Feinplanungsverfahren durch Sortierung der

Verspätung der einzelnen Fragmente (Aufträge) erkennen, welches Fragment die höchste Verspätung aufweist.

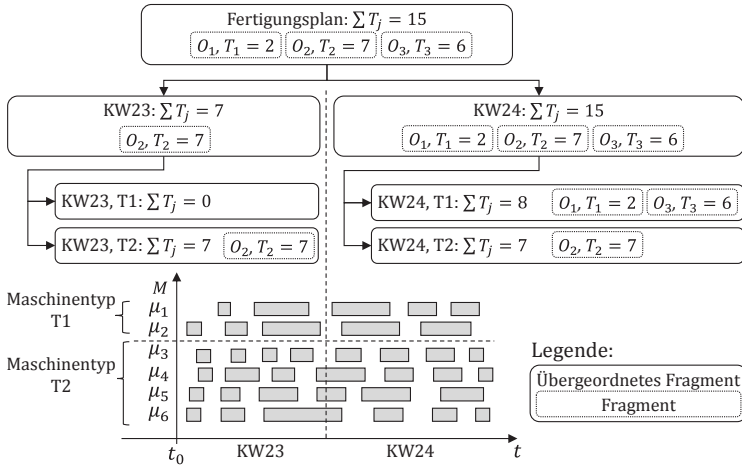


Abbildung 4.10: Fragmente samt Operationen aus der Bewertung nach $\sum T_j$

Für andere Optimierungsziele wie die Erhöhung der Maschinenauslastung ist dagegen eine zeitbezogene Fragmentierung notwendig. Für die Reduzierung der Gesamtrüstzeit ist ferner auch Fragmentierung nach sich gegenseitig ersetzenden Maschinen, also nach Maschinentyp, notwendig. Eine Optimierung der Rüstreihenfolge soll alle alternativen Maschinen gleichzeitig betrachten, weil einzelnen Maschinen zwar schlechte Rüstfolgen von Operationen zugewiesen werden können, während andere Alternativen ohne Umrüstung auskommen.

Bei einer multikriteriellen Zielsetzung kommt es deshalb oft auf die Nutzung mehrerer unterschiedlicher parallel eingesetzten Fragmentierungen an. Die Ermittlung des Optimierungspotentials der einzelnen Fragmente wird im nächsten Abschnitt genauer betrachtet.

4.2.6 Bewertungskriterien für den Vergleich von Fragmenten

Bewertungskriterien, die als Zielfunktionen dienen, sind für den Vergleich mehrerer Lösungen ein und desselben Planungsproblems ausgelegt. Solche Bewertungskriterien sind die Summe der Verspätungen $\sum T_j$, die Summe der Fertigstellungszeitpunkte $\sum C_j$ oder die Summe der Rüstzeiten TST (vgl. 2.2.3). Eingesetzt für Fragmente des Fertigungsplans oder für abgelaufene Fertigungsperioden, haben sie einen rein informativen Charakter. Denn die unterschiedlichen Fertigungsperioden oder Fragmente besitzen möglicherweise eine unterschiedliche Anzahl an Aufträgen, unterschiedliche Kapazitätsverfügbarkeiten usw. Auch andere Bewertungskriterien wie der Beleggrad, Rüstgrad oder Prozessgrad [Vdma09], die häufig als informative Kennzahlen oder als sekundäre Optimierungsziele dienen, haben dieselbe Einschränkung.

Etwas Verbesserung bringen Kennzahlen, die den Durchschnitt, die Varianz oder die Standardabweichung aus den obigen Kriterien bilden, da sie sie in gewisser Weise normieren. Mit diesen fortgeschrittenen Kennzahlen sind auch (abgelaufene) Fertigungsperioden in gewissem Maß vergleichbar. Fragmente bestehend aus unterschiedlichen Maschinen oder Aufträgen können jedoch ganz unterschiedliche Durchschnitts- oder Varianzwerte aufweisen. Es ist deshalb eine Weiterentwicklung dieser Kennzahlen notwendig, um eine Vergleichbarkeit in der Feinplanung zu ermöglichen.

Die Minimierung einer Zielfunktion für ein konkretes Shop Scheduling Problem ist oft nur bis zu einem gewissen Grad möglich. Während einer Optimierung weist die berechnete untere Schranke der gesetzten Zielfunktion (vgl. 2.2.4) auf das restliche theoretische Optimierungspotential hin. Um dieses Optimierungspotential in den Fragmenten aufzuspüren, müssen die Kriterien für den Vergleich von Fragmenten es widerspiegeln.

Ein hohes Optimierungspotential ist nicht immer äquivalent zu einem großen Wert der klassischen Kennzahlen. Zum Beispiel kann die durchschnittliche Durchlaufzeit eines Fragments am längsten sein, jedoch kann sie mit der unteren Schranke der durchschnittlichen Durchlaufzeit übereinstimmen. Das Gleiche gilt für sehr hohe Verspätungen von Aufträgen, deren spätester Fertigstellungszeitpunkt weit in der Vergangenheit liegt ($d_j \ll t_0$). Im aktuellen Feinplanungsproblem ist eventuell kaum mehr Spielraum für Verbesserung vorhanden.

Auch wenn die Zielfunktion aus klassischen Kennzahlen besteht, ist eine Bewertung der Fragmente hinsichtlich des Optimierungspotentials von großem Vorteil. Tabelle 4.2 stellt für ausgewählte und bekannte Bewertungskriterien die Schwächen bei der Ermittlung des Optimierungspotentials dar und zeigt beispielhafte Vorschläge für die Erhöhung der Genauigkeit bei der Potentialermittlung. Dabei repräsentiert YZ_{ji} den benötigten Rüstzustand von o_{ji} . Die Funktion $C_j^\theta(t)$ ermittelt den Endzeitpunkt für den Startzeitpunkt t unter der Annahme, dass die Problemstellung nur den Auftrag j enthält.

Wie der Tabelle zu entnehmen ist, ist das Optimierungspotential für Fragmente in vielen Fällen deutlich genauer ermittelbar. Bei Kriterien wie TST und TSC , die stark von der Maschinenart oder der eingesetzten Fertigungstechnologie abhängen, kann ein Vergleich des Optimierungspotentials von zwei Fragmenten mit verschiedenartigen Maschinen nur bedingt erfolgen. Maschinen mit mehreren verschiedenartigen Operationen haben oft, auch wenn die Operationen optimal angeordnet sind, eine höhere Rüstzeit als Maschinen mit nur wenigen unterschiedlichen Operationen. Bei Ermittlung des Optimierungspotentials von TST und TSC wird zusätzlich die Anzahl verschiedenartiger Operationen $|\cup_{j,i}\{YZ_{ji}\}|$ berücksichtigt. In manchen Fällen würde zusätzlich die durchschnittliche Umrüstdauer die Aussagekraft steigern.

Tabelle 4.2: Erweiterung klassischer Zielfunktionen aus 2.2.3 und 4.2.1 um das Optimierungspotential (OP)

Ziel- funkt- ion	Klassisch Zeitbehaftet	Klassisch mit OP Zeitbehaftet mit OP
ΣE_j	$\Sigma \max\{d_j - c_j, 0\}$ $\Sigma \max\left\{\int_{d_j}^{c_j} g(t)dt, 0\right\}$	<i>unverändert</i>
ΣT_j	$\Sigma \max\{c_j - d_j, 0\}$ $\Sigma \max\left\{\int_{d_j}^{c_j} g(t)dt, 0\right\}$	$\Sigma \max\{c_j - \max\{d_j, c_j^\theta(t_0)\}, 0\}$ $\Sigma \max\left\{\int_{\max\{d_j, c_j^\theta(t_0)\}}^{c_j} g(t)dt, 0\right\}$
$1/n \Sigma F_j$	$1/n \Sigma (c_j - r_j)$ $1/n \Sigma \int_{r_j}^{c_j} g(t)dt$	$\Sigma (c_j - c_j^\theta(r_j))$ $\Sigma \int_{c_j^\theta(r_j)}^{c_j} g(t)dt$
$\emptyset DLZ$	$1/n \Sigma (c_j - s_j)$ $1/n \Sigma \int_{s_j}^{c_j} g(t)dt$	$\Sigma (c_j - c_j^\theta(s_j))$ $\Sigma \int_{c_j^\theta(s_j)}^{c_j} g(t)dt$
<i>TST</i>	$\Sigma_{j,i} (Y C_{ji} - Y S_{ji})$ $\Sigma_{j,i} \int_{Y S_{ji}}^{Y C_{ji}} g(t)dt$	$(\Sigma_{j,i} (Y C_{ji} - Y S_{ji})) / U_{j,i}\{Y Z_{ji}\} $ $/ U_{j,i}\{Y Z_{ji}\} $
<i>TSC</i>	$\Sigma_{j,i} Y R_{ji}$ $\Sigma_{j,i} \frac{\int_{Y S_{ji}}^{Y C_{ji}} g(t)dt \cdot Y R_{ji}}{Y C_{ji} - Y S_{ji}}$	$(\Sigma_{j,i} Y R_{ji}) / U_{j,i}\{Y Z_{ji}\} $ $\left(\Sigma_{j,i} \frac{\int_{Y S_{ji}}^{Y C_{ji}} g(t)dt \cdot Y R_{ji}}{Y C_{ji} - Y S_{ji}} \right) / U_{j,i}\{Y Z_{ji}\} $

Die Überlegungen in diesem Abschnitt zeigen, dass für die Bewertung der einzelnen Fragmente weiterentwickelte Bewertungskriterien verwendet werden müssen, um eine gute Vergleichbarkeit der Fragmente zu gewährleisten. Für jedes klassische Bewertungskriterium muss das passende Kriterium für Fragmente nach einer Analyse abgeleitet werden.

Die in diesem Unterkapitel beschriebene Bewertungskomponente bietet eine umfassende Evaluierung des Fertigungsplans nach den verschiedenen notwendigen Kriterien wie Planstabilität, akkumulierte Planstabilität, Einhaltung der Benutzerinteraktionen und selbstverständlich den klassischen Kennzahlen. Dabei wurde in allen implementierten Kriterien die Relevanz zum aktuellen Zeitpunkt integriert. Zusätzlich führt die Bewertungskomponente eine Fragmentierung des Fertigungsplans nach unterschiedlichen Merkmalen durch. Abhängig von der gesetzten Zielfunktion ermittelt sie das Optimierungspotential der entstandenen Fragmente mittels einer Weiterentwicklung der klassischen Kennzahlen. Die nächsten Unterkapitel zeigen, wie diese Ergebnisse möglichst gut verwertet werden.

4.3 Vergleich von Fertigungsplänen

Für die Anbindung unterschiedlicher Algorithmen an das Framework ist eine Unifizierung der Vergleichsfunktionen vorteilhaft. Der multikriterielle Vergleich ist im Allgemeinen ein unlösbares Problem. Jedoch haben sich einige Methoden etabliert, um einen Fertigungsplan aus einer Menge von Fertigungsplänen algorithmisch auszuwählen (vgl. 2.2.3).

Die Auswahl des besten Planvorschlags anhand der Benutzerwünsche ist für eine korrekte verfahrensübergreifende Optimierung notwendig. Neben der Verbesserung der Kennzahlen sind für eine neue Übernahme auch die Einhaltung der Benutzerinteraktionen und die verursachte Nervosität wichtig, da der neue Fertigungsplan auch für alle Benutzer und die Fertigung gilt. Falls die Optimierung nicht als untergeordneter, sondern als gleichstehender Partner agiert, erwarten die Planer auch selbstständig durchgeführte Änderungen im aktuellen Plan bzw. eigeninitiierte Übernahmen.

Der Vergleich von Fertigungsplänen wird durch die Berücksichtigung der neuen Kennzahlen θ und ϑ^b multikriteriell. Diese Kennzahlen werden im Vergleich sowohl als Restriktionen als auch als Optimierungsziele einfließen. Es dürfen nur automatische Übernahmen stattfinden, wenn bestimmte vom Benutzer vorkonfigurierte Grenzwerte (GW) für diese Kennzahlen eingehalten werden. Bei sehr kleinen Werten dieser Kennzahlen ist eine Übernahme unproblematisch. Abbildung 4.11 zeigt schematisch, wie die Fertigungspläne P_1 bis P_4 automatisch übernommen werden. Je näher ein Fertigungsplan am Mittelpunkt der linken Kreise liegt, desto weniger Nervosität verursacht er, und je näher er am Mittelpunkt der rechten Kreise liegt, desto stärker hält er die Benutzerinteraktionen ein.

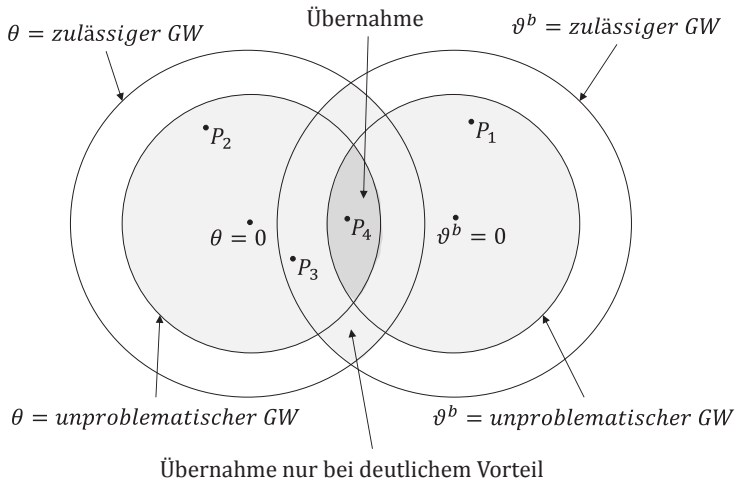


Abbildung 4.11: Übernahme abhängig von θ und ϑ^b

Falls die Werte von θ und ϑ^b innerhalb der zulässigen Grenzen liegen oder nur einer der Werte im unproblematischen Bereich liegt, kann nicht eindeutig über eine Übernahme entschieden werden, da die Veränderungen in der Fertigung und für den Planer sehr spürbar werden. Eine Übernahme findet dann nur

statt, wenn ein deutlich besserer Planvorschlag sie rechtfertigt. Dem Benutzer soll die Möglichkeit vorbehalten bleiben, diese Übernahme manuell auszulösen. In diesem Fall trägt der Benutzer auch die volle Verantwortung für die entstandene Unruhe.

Die hier beschriebene Vergleichskomponente nutzt die zwei zusätzlichen Kennzahlen für die akkumulierte Planstabilität und die Einhaltung der Benutzerinteraktionen, um ungeeignete Planübernahmen intelligent auszufiltern, die ein nach klassischer Zielfunktion optimierendes Feinplanungsverfahren durchführen würde. Die Vergleichskomponente kann auch von fortgeschrittenen Feinplanungsverfahren wie GA oder Ameisenalgorithmen verwendet werden, um die eingebaute Vergleichsroutine zu ersetzen.

4.4 Unterstützung der Feinplanungsverfahren bei der Lösungssuche

Lange Rechenzeiten stellen heute immer noch eine sehr große Behinderung für den Einsatz fortgeschrittener Feinplanungsverfahren für umfangreiche Planungsprobleme (vgl. 3.4) dar. Die zusätzlichen Bewertungskriterien (vgl. 4.2) und die Vergleichsfunktion (vgl. 4.3) erhöhen die Komplexität des Planungsproblems und stellen noch höhere Anforderungen an die algorithmische Optimierung. Eine Verkürzung der Optimierungszeit kann mithilfe von Wissen erreicht werden, das die Bewertungskomponente während der Evaluierung sammelt.

Jedes Feinplanungsverfahren lässt sich von außen ansteuern, jedoch nur zu einem unterschiedlichen Ausprägungsgrad. Falls ein Feinplanungsverfahren nicht von außen auf bestimmten Fragmenten im Fertigungsplan steuerbar ist, kann es über die Eingangsparameter, den Fertigungsplan und die Problemstellung beeinflusst werden. Die Feinplanungsplattform kann aus zusam-

menhängenden Fragmenten mit Optimierungspotential eine abgespeckte Problemstellung konstruieren und sie als Eingangsparameter übergeben. Da weniger Dispositionsobjekte in der abgespeckten Problemstellung existieren, kann die Optimierungszeit deutlich verkürzt werden. Verarbeitet ein Feinplanungsverfahren externe Optimierungshinweise, so kann die Feinplanungsplattform Fragmente mit Optimierungspotential ausfiltern und ihm übergeben. Das würde die Analyse oder den „Zufall“ im fortschrittlichen Feinplanungsverfahren ersetzen. Diese beiden Alternativen für die Lenkung eines Feinplanungsverfahrens können auch in Kombination eingesetzt werden.

4.4.1 Hinweise auf Fragmente mit Optimierungspotential

Die Mehrheit der Feinplanungsverfahren kann durch leichte Modifizierung oder von Natur aus Optimierungshinweise verarbeiten. Das Schema in Abbildung 4.12 verallgemeinert den internen Ablauf dieser Feinplanungsverfahren. Ausgehend von einem Fertigungsplan wählt das Feinplanungsverfahren eine oder mehrere Operationen aus. Diese Auswahl findet durch eine bestimmte Heuristik oder Zufall statt. Danach ermittelt das Feinplanungsverfahren für die ausgewählten Operationen die möglichen Umplanungen auf alternative Ressourcen oder zu anderen Ausführungszeiten. Aus diesen möglichen Umplanungen wählt das Feinplanungsverfahren wieder durch Heuristik oder Zufall eine oder mehrere Umplanungen. Am Ende führt es die ausgewählte Umplanung durch.

Dieses Schema kann lokale Suchverfahren sowie deren Derivate Simulated Annealing und Tabu Search sehr genau abbilden (vgl. 2.3.4 und 2.3.5). Diese Feinplanungsverfahren verwenden das Planungsmodell direkt. Metaheuristische Verbesserungsverfahren wie die genetischen Algorithmen oder Ameisenalgorithmen

verwenden zwar Metamodelle, können aber die Optimierungshinweise für die Veränderungen des internen Metamodells nutzen. Ein GA kann in der Mutationsphase gezielt statt zufällig die Gene verändern, die Operationen in den Optimierungshinweisen repräsentieren. In der Kreuzungsphase kann der GA die Übertragung dieser Gene in die nächste Generation nach Möglichkeit vermeiden. Bei den Ameisenalgorithmen dagegen kann der künstliche Weg (die Pheromonspuren) der Ameise(n) durch die Optimierungshinweise verändert werden.

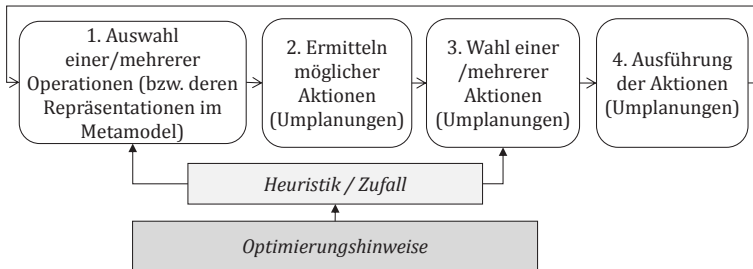


Abbildung 4.12: Allgemeiner Ablauf von Feinplanungsverfahren

Konstruktionsverfahren (vgl. 2.3.2 und 2.3.3) können die Optimierungshinweise während der Erstellung des Fertigungsplans nutzen. Der GT Algorithmus kann Operationen in den Optimierungshinweisen höher/niedriger priorisieren und sie je nach Hinweisart früher oder später einplanen. Die Heuristik beim heuristischen Branch-and-Bound kann so modifiziert werden, dass Zweige mit Operationen in den Optimierungshinweisen zuerst untersucht werden.

Um eine optimale Unterstützung der Feinplanungsverfahren zu gewährleisten, müssen Optimierungshinweise passend zur gesetzten Zielfunktion geliefert werden. In Abschnitt 4.2.6 wurden Bewertungskriterien speziell für den Vergleich von Fragmenten definiert. Bei Verwendung unterschiedlicher Fragmentierungen oder bei multikriterieller Zielsetzung entsteht eine deutlich höhere Anzahl von sich teilweise überlappenden Fragmenten, als es

Operationen im Fertigungsplan gibt. Daraus müssen die brauchbaren ausgefiltert und als Optimierungshinweise an die Feinplanungsverfahren bereitgestellt werden.

Bei der Vorbereitung der passenden Optimierungshinweise werden zuerst alle Fragmente (pro Optimierungsziel bei multikriterieller Zielsetzung) analysiert. Die vielversprechendsten davon werden als Top-Fragmente markiert (s. Abbildung 4.13). Die Top-Fragmente müssen nicht zwangsläufig die Zielfunktion negativ beeinflussen. Auch zu früh endende Aufträge, wie z. B. Auftrag 3 der Abbildung, kann eine potentielle Optimierungsstelle sein. Denn durch seine Umplanung auf einen späteren Zeitpunkt könnte theoretisch Kapazität für verspätete Aufträge freigegeben werden. Alle Top-Fragmente zu den gesetzten Optimierungszielen müssen miteinander verglichen werden, um die Optimierungshinweise auszufiltern, die danach als Optimierungshinweise dienen werden. Da im Vergleich der Fragmente die gleichen Ziele wie im Planvergleich verfolgt werden, kann an dieser Stelle eine leicht abgewandelte Version des Planvergleichs herangezogen werden.

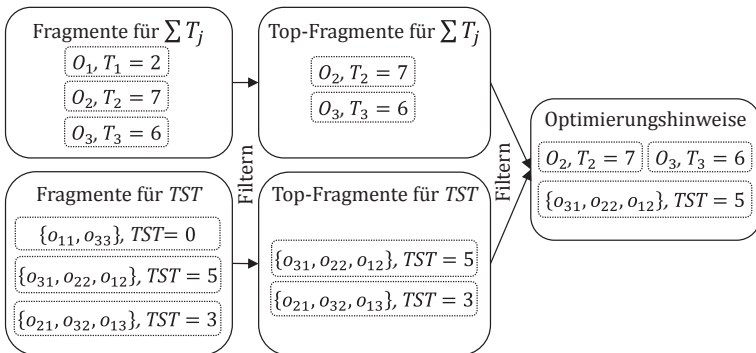


Abbildung 4.13: Ableitung von Optimierungshinweisen für die Unterstützung von Feinplanungsverfahren

Diese Optimierungshinweise sind eine zusätzliche Hilfe. Durch sie sollen die Feinplanungsverfahren die Rechenzeit verkürzen. Die alleinige Behandlung der Optimierungsstellen führt theoretisch nicht immer zu einer Verbesserung, da das Shop Scheduling Problem äußerst komplex ist und sehr viele Zusammenhänge zwischen den beteiligten Objekten enthält, die teilweise für die Bewertung unbekannt sein können. In den meisten Fällen werden jedoch eine deutliche Verbesserung des Endergebnisses und ein nachvollziehbares Verhalten erwartet.

4.4.2 Dynamische Dekomposition des Fertigungsplans

Eine Alternative zur Bereitstellung von Optimierungshinweisen bildet die Lenkung eines Feinplanungsverfahrens durch den Eingangsparameter. Das Feinplanungsverfahren befasst sich nur mit einem zuvor ausgewählten Abschnitt aus dem gesamten Fertigungsplan. Wenn die Feinplanungsplattform suboptimale Fragmente ausfindig macht und an ein oder mehrere Feinplanungsverfahren zur parallelen Bearbeitung sendet, wird sie die Optimierungszeit deutlich verkürzen. Wie in Abschnitt 2.2.4 beschrieben, hängt die Optimierungszeit stark vom Problemumfang ab. Eine Reduzierung des Umfangs erhöht die Performanz der deterministischen Feinplanungsverfahren und die Wahrscheinlichkeit für bessere Ergebnisse bei den stochastischen Feinplanungsverfahren. Zu diesem Zweck sollte die Feinplanungsplattform in der Lage sein suboptimale Abschnitte des Fertigungsplans einzugrenzen.

Für die Identifikation von suboptimalen Teilen innerhalb des Fertigungsplans kann das Ergebnis der Bewertung hilfreich sein. Die Fragmentierungsfunktionen der Bewertung können den Fertigungsplan nicht nur semantisch nach z. B. Auftrag, Kunde oder erstelltem Material fragmentieren, sondern auch „topolo-

gisch“: Die Fragmente können alle Operationen aus einem bestimmten Zeitintervall und auf sich gegenseitig ersetzenden Maschinen enthalten. Ein Feinplanungsverfahren hat in solchen Fragmenten ausreichenden Optimierungsspielraum und kann sie deshalb isoliert optimieren. Die Fragmentierungsfunktionen erstellen eine Hierarchie aus bewerteten Fragmenten mit dem gesamten Fertigungsplan als Wurzel (vgl. 4.2.5). In Abbildung 4.14 ist die hierarchische Anordnung der übergeordneten Fragmente aus Abbildung 4.10 zu sehen. Durch die Durchsuchung der Knoten des entstandenen Baums aus Fragmenten können suboptimale Fragmente identifiziert werden. Die Abbildung zeigt eine Tiefensuche, die anhand der Bewertung der Fragmente erfolgt.

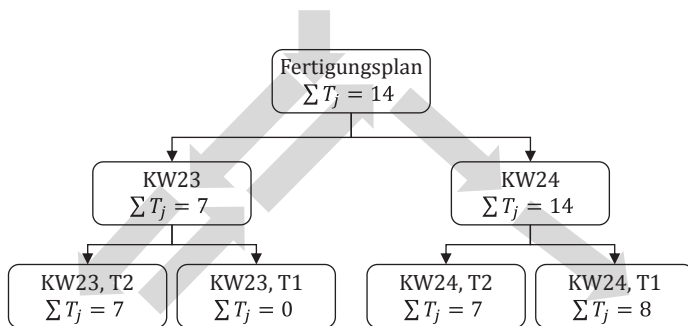


Abbildung 4.14: Untersuchung der Fragmenthierarchie

Bei Auswahl mehrerer suboptimaler Fragmente ist es günstig, die zeitlich kompatiblen Fragmente aus den nachfolgenden oder denselben Zeitabschnitten auszuwählen, da als nächstes ein für das Feinplanungsverfahren bequemer Teil des Fertigungsplans isoliert werden muss. Innerhalb der ausgewählten übergeordneten Fragmente weisen die Bewertungen der atomaren Fragmente auf Schwachstellen hin. Um diese Schwachstellen soll ein genügend großer Teil des Fertigungsplans eingegrenzt werden. Abbildung 4.15a zeigt einen Fertigungsplan samt aller Beziehungen zwischen den Operationen. Über diese Beziehungen werden alle Operationen markiert, welche mit den Schwachstellen verbunden

sind, so dass ein zusammenhängender Teilgraph entsteht. Anschließend werden aus den markierten Operationen über deren Beziehungen weitere Operationen solange markiert bis die geforderte Markierungsgröße erreicht ist.

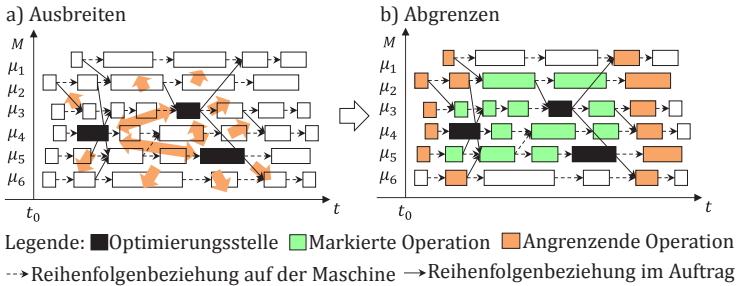


Abbildung 4.15: Markierung von einem Teil des Fertigungsplans

In Abbildung 4.15b ist die Markierung samt angrenzenden Operationen zu sehen. Aus dieser Markierung lässt sich ein abgespecktes Feinplanungsproblem konstruieren, das einer beliebigen Verfahrensinstanz zugeteilt werden kann.

Die Untersuchung der Fragmenthierarchie und die Markierung von Operationen können so gestaltet werden, dass sie vom Benutzer veränderte Fragmente bzw. Operationen meiden. So können auch die Benutzerinteraktionen implizit eingehalten werden. Eine Verfahrensinstanz ändert nur ihr zugeteilte Operationen und verursacht deshalb deutlich geringere Nervosität. Da sie isoliert abläuft, ist es möglich eine weitere Verfahrensinstanz auf einem anderen disjunkten Teil des Fertigungsplans zu starten, was im nächsten Abschnitt beschrieben wird.

4.4.3 Parallele Ausführung von Verfahrensinstanzen

Eine Verfahrensinstanz kann nur auf einer Untermenge der Dispositionsobjekte ausgeführt werden. Eine brauchbare Dekomposi-

tion des Feinplanungsproblems wie etwa bei der dynamischen Programmierung ist normalerweise nur in wenigen Fällen und unter bestimmten Bedingungen möglich [AyKU03, OvUz97]. Nach der Dekomposition entstandene Teile können eventuell optimal gelöst werden, aber bei ihrer Zusammenführung entsteht in der Regel kein optimaler Fertigungsplan. Die Dekomposition schränkt den Lösungsraum stark ein, da Operationen aus einem Teil des Fertigungsplans nicht in einen anderen wandern können. Diese Nachteile betreffen die einmalige (statische) Dekomposition.

Es ist jedoch möglich disjunkte Teile des Fertigungsplans gleichzeitig zu optimieren, wie der nachfolgende Ansatz zeigt. Abbildung 4.16a zeigt zwei parallele Optimierungsprozesse, die je einen Teil des Fertigungsplans eingegrenzt haben und ihn optimieren. Der restliche Teil des Fertigungsplans ist frei für die weitere Optimierung. Ein weiterer selbstständiger Prozess kann einen Teil für sich eingrenzen (b) und ihn ausschließlich verändern (c). In nachfolgenden Optimierungszyklen kann der Fertigungsplan anders aufgeteilt werden, sodass die neuen Teile sich mit den alten Teilen überschneiden. So kann eine Operation nach einer gewissen Anzahl an Aufteilungen bei Bedarf theoretisch auf eine beliebige Ressource und in einem beliebigen Zeitintervall umdisponiert werden.

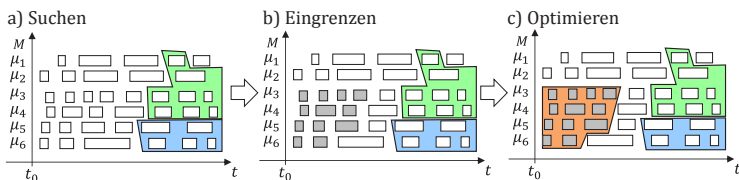


Abbildung 4.16: Beispielhafte Markierung bei mehreren Verfahrensinstanzen

Durch die Eingrenzung bekommt die Verfahrensinstanz exklusive Schreiberechte auf die betroffenen Objekte. Die angrenzenden Operationen werden als schreibgeschützt vermerkt, da sie als

abhängige Objekte den Lösungsraum beeinflussen. Die schreibgeschützten Operationen können unter den Verfahrensinstanzen geteilt werden.

Des Weiteren muss der Fertigungsplan nicht vollständig aufgeteilt werden. Die Verfahrensinstanzen können nur einige wenige suboptimale Teile optimieren. Während dieser Optimierung kann die Bewertung suboptimale Fragmente im zurzeit nicht markierten Teil des Fertigungsplans identifizieren. Falls sie solche findet, kann sie einen Teil des Fertigungsplans, der eine Untermenge der suboptimalen Fragmente enthält, eingrenzen und zur Optimierung an eine weitere Instanz des Feinplanungsverfahrens übergeben.

Nach diesem Schema erfolgt eine performante Eingrenzung von beliebigen Teilen des Fertigungsplans. Die Teile können unberührt von einer beliebigen Verfahrensinstanz optimiert werden, ohne davon Kopien zu erzeugen. Somit wird die Eingabemenge reduziert und die Gesamtoptimierung kann drastisch beschleunigt werden.

4.5 Koordination gleichzeitiger Aktionen von Benutzern und Algorithmen

Mehrere Benutzer und auch eventuell mehrere Feinplanungsverfahren greifen auf ein und denselben Fertigungsplan zu (Anforderung A2). Zudem läuft auch die Fertigung nach ihm und meldet alle Abweichungen zurück. Übergeordnete Software-Systeme überspielen modifizierte Stamm- und Bewegungsdaten, was eine Plananpassung erfordert.

Die im vorigen Kapitel vorgestellten Komponenten der Feinplanungsplattform reichen alleine nicht für die Planoptimierung in Kollaboration mit Benutzern in einer dynamischen Umgebung. Die

angebotenen Funktionen wie die Bewertung, die Unterstützung von Feinplanungsverfahren und der Vergleich von Plänen müssen im Falle des angestrebten gleichzeitigen Zugriffs koordiniert werden. Eine Optimierung soll nicht gleichzeitig vom Benutzer und dem Feinplanungsverfahren auf demselben Teil des Fertigungsplans erfolgen. Gleichzeitig soll die Feinplanungsplattform den Benutzern eine komfortable, verzögerungsfreie Bedienung ermöglichen. Hierfür werden deshalb Ansätze für die Kollaboration zwischen Verfahrensinstanzen und Planern abgeleitet.

4.5.1 Verarbeitung von externen Ereignissen

Die Feinplanungsplattform agiert in einer dynamischen Umgebung, in der sich ständig etwas ändert. Wie in Unterkapitel 3.2 festgestellt, sind mehrere Beteiligte in den Planungsprozess involviert. Sie müssen ihre Änderungsereignisse möglichst unmittelbar nach Eintreten verarbeiten und in das Datenmodell einpflegen, um stets einen aktuellen Stand für die algorithmische Optimierung und die Benutzer anzubieten. Verwenden die agierenden Parteien einen veralteten Stand, so sind falsche Entscheidungen wahrscheinlich. Die Feinplanungsplattform kann diese Ereignisse nicht immer sofort bearbeiten, da sie in unregelmäßigen Zeitabständen und Umfang ankommen. Sie kann die Ereignisse in einer Warteschlange speichern (s. Abbildung 4.17) und bei erster Gelegenheit nach dem Windhundprinzip bearbeiten.

Falls ein angekommenes Ereignis Bereiche des Fertigungsplans betrifft, die keine Verfahrensinstanz optimiert, kann es ohne Behinderung in das interne Planungsmodell eingefügt werden. Falls das Ereignis Ressourcenverfügbarkeiten, Operationen, Aufträge oder andere planungsrelevante Objekte auf bereits von einer Verfahrensinstanz reservierte Teile referenziert, muss diese Verfahrensinstanz ihre Optimierung abbrechen.

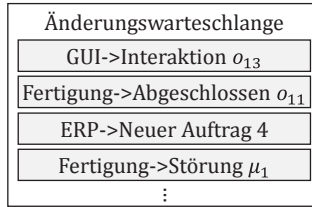


Abbildung 4.17: Warteschlange für Ereignisse in der Feinplanungsplattform

Im ungünstigsten Fall betrifft ein Ereignis mehrere Verfahrensinstanzen. Die Feinplanungsplattform kann das Ereignis dann nicht als Ganzes bearbeiten und zerlegt es. Abbildung 4.18a zeigt das Erzeugen zweier atomarer Ereignisse aus einer Benutzerinteraktion. Die Feinplanungsplattform kann die erste betroffene Verfahrensinstanz unterbrechen und die Änderung im ersten atomaren Ereignis durchführen (s. Abbildung 4.18b). Anschließend kann sie die zweite Verfahrensinstanz unterbrechen und den zweiten Teil bearbeiten (s. Abbildung 4.18c).

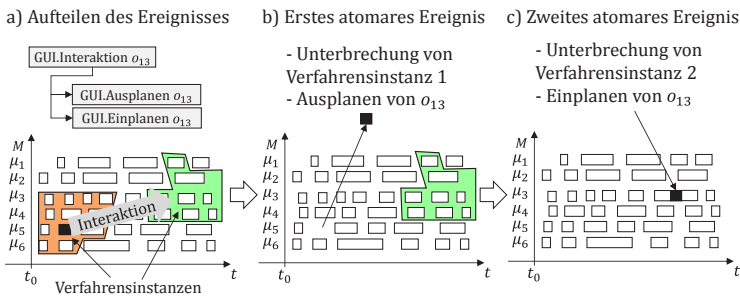


Abbildung 4.18: Verarbeitung einer Benutzerinteraktion bei mehreren Verfahrensinstanzen

Falls die betroffene Stelle in der Abgrenzung eines markierten Teils liegt und somit schreibgeschützt ist, müssen eventuell mehrere Instanzen abgebrochen werden. Gleichzeitig muss sichergestellt werden, dass neue Teile die Operation nicht ein weiteres Mal für den Lesezugriff markieren.

Je nach Ereignisart können die betroffenen Operationen mit hoher oder niedriger Wahrscheinlichkeit in der nächsten Markierung enthalten sein. Kurz zuvor vom Benutzer geänderte Operationen und die abhängigen Objekte sollen aus einer Reservierung ausgeschlossen sein, damit die beabsichtigte Interaktionsfolge ohne Störung durchgeführt werden kann. Störungen oder Änderungen in der Auftragsmenge oder im spätesten Fertigstellungszeitpunkt müssen oft sofort in der Feinplanung betrachtet werden.

Die Abwicklung jedes Ereignisses erfordert im ungünstigsten Fall die Unterbrechung einer Verfahrensinstanz, die möglicherweise einen Optimierungsfortschritt geleistet hat, wodurch Rechenzeit verloren geht. Der nächste Abschnitt zeigt, wie die Rechenzeit für die Feinplanung optimal von mehreren Instanzen verwendet werden kann.

4.5.2 Koordinierung mehrerer Verfahrensinstanzen

In der Praxis sind Shop Scheduling Probleme deutlich umfangreicher als die theoretischen Benchmarks und überfordern sogar einfache Konstruktionsverfahren. Die Rechenleistung der Systeme steigt zwar ununterbrochen, jedoch trägt die Anzahl der Prozessorkerne zu diesem Wachstum in letzter Zeit besonders bei. Durch Umbau können viele Feinplanungsverfahren die Gegebenheiten der Multiprozessorsysteme ausnutzen [AiBR03, LiGI97, LuAl13, SeAy07]. Die Problemgröße bleibt jedoch weiterhin eine Behinderung. Die Ausführung mehrerer Verfahrensinstanzen auf Teilen des Fertigungsplans würde einerseits alle verfügbaren Prozessorkerne auslasten. Andererseits bilden die Fragmente Problemstellungen mit kleinerer Problemgröße, die sich auf die Optimierungszeit positiv auswirkt.

Die Prozessoren eines Rechensystems sind eine beschränkte Ressource, die auch andere Dienste innerhalb und außerhalb der Feinplanungsplattform zeitweise benötigen. Die Verfahrens-

instanzen sollen optimal die verfügbare Rechenzeit verwenden, unnötige Wartezeiten im Koordinationsprozess vermeiden und gleichzeitig andere wichtige Dienste nicht ausbremsen. Die Feinplanungsplattform muss deshalb in der Lage sein, Verfahrensinstanzen bei Bedarf zu unterbrechen bzw. auszuschalten und im Falle freier Rechenkapazität zu aktivieren bzw. neue anzulegen. Gleichzeitig sollen Verfahrensinstanzen den Planer im Planungsprozess nicht z. B. durch Verursachung langer Reaktionszeiten behindern.

Jede Verfahrensinstanz optimiert einen abgegrenzten Teil des Fertigungsplans. Statt der Feinplanungsplattform kann die Verfahrensinstanz die Markierung selbst durchführen und nach dem Optimierungslauf ihre Optimierungsergebnisse in den Fertigungsplan selbst einpflegen. Falls die Verfahrensinstanzen sich während dieser zwei Aufgaben nicht gegenseitig stören, ist keine separate Koordinierungsinstanz notwendig.

Es ist möglich, die Verfahrensinstanzen unabhängig voneinander, selbstständig und gleichberechtigt zu gestalten. Damit sie parallel ablaufen können müssen sie sich alle an die gleiche Regel halten: Es darf nur eine Instanz gleichzeitig auf nicht markierte Teile des Fertigungsplans zugreifen. Die anderen Instanzen dürfen während dieser Zeit nur die zuvor markierten Teile verändern.

Der Ablauf aus Abbildung 4.19 erfüllt die oben gesetzten Vorgaben und nutzt gleichzeitig die komplette verfügbare Rechenzeit. Zuerst wird eine Verfahrensinstanz erstellt bzw. aktiviert (1), die exklusive Rechte auf den Fertigungsplan einfordert. Nach deren Erhalt findet (2) und markiert (3) sie einen (optimierungsbedürftigen) Teil des Fertigungsplans. Nach der Freigabe der exklusiven Rechte (4) werden nicht unterbrechbare Verfahrensschritte solange durchgeführt, bis ein Abbruchgrund (6) vorliegt oder zuvor asynchron angeforderte (7, 8) exklusive Rechte zugeteilt (9) wurden. Nach dem Abbruch arbeitet die Verfahrensinstanz die

Ereignisse aus der Ereigniswarteschlange ab, die keine markierten Teile von anderen Instanzen betreffen. Sobald eine andere Verfahrensinstanz die Abarbeitung ausbremst, wird ihr zum Abbruch ihrer Optimierung geraten (10). Danach entscheidet die Instanz, ob die beabsichtigten Änderungen nach der Abarbeitung der Ereignisse noch Verbesserung im Fertigungsplan bringen und trägt sie ggf. dort ein (11). Anschließend aktualisiert die Verfahrensinstanz die zugehörigen Hilfsdatenstrukturen sowie die Planbewertung. Je nach externem Bedarf von Rechenleistung (12) sucht sich die Verfahrensinstanz einen neuen Teil zum Optimieren (2) oder wird nach der Freigabe der exklusiven Rechte beendet bzw. unterbrochen (13).

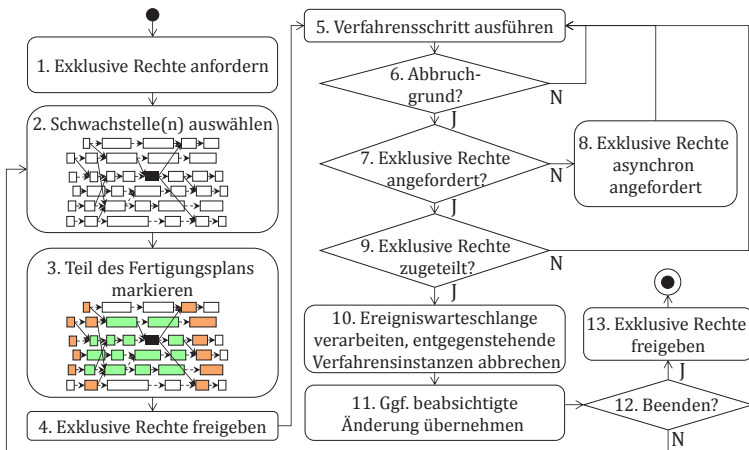


Abbildung 4.19: Aktivitätsdiagramm einer Verfahrensinstanz [DiBS11]

Für den Abbruch des Optimierungslaufs existieren verschiedenartige Gründe. Interne Abbruchgründe sind rein vom Optimierungslauf abhängig. Ein solcher Abbruchgrund ist die Erreichung einer vorgegeben Verbesserung des Zielfunktionswertes des markierten Teils. Ein weiterer interner Abbruchgrund kann die maximale Optimierungsdauer pro Teil des Fertigungsplans sein, nach dessen Ablauf sich die Verfahrensinstanz einen neuen (optimierungsbe-

dürftigen) Teil aussuchen muss. Diese Bedingung ist notwendig, damit die Verfahrensinstanz nicht an der Optimierung eines einzigen Teils hängen bleibt.

Dagegen existieren auch externe, von der Außenwelt abhängige Abbruchgründe, wie die Änderung der Aufgabenstellung, Benutzerinteraktionen oder Fortschrittmeldungen aus der Fertigung, die die markierten Teile betreffen. Auch kann das System Rechenressourcen brauchen und deswegen eine Verfahrensinstanz unterbrechen bzw. beenden.

Dieser Ablauf erzielt ein menschenähnliches Verhalten in der Optimierung. Jede Verfahrensinstanz sucht sich genau wie ein Planer bestimmte optimierungsbedürftige Stellen im Fertigungsplan. Sie optimiert diese Stellen solange bis eine gewünschte Verbesserung erzielt wird oder keine solche in Sicht ist. Gleichzeitig nimmt sie Rücksicht auf die Benutzerinteraktion und passt ihre Optimierung danach an. Dadurch sind die erzielten Ergebnisse mit größerer Wahrscheinlichkeit vorhersagbar und somit akzeptabler für den Menschen.

5 Die Feinplanungsplattform im Hinblick auf Performanz und Skalierbarkeit

Das vorige Kapitel stellt das Rahmenkonzept für eine Feinplanungsplattform vor, die durch vielfältige Funktionen eine dynamische Fertigungssteuerung unter Einbeziehung der Planer realisiert. Die Feinplanungsplattform deckt alle in 3.3 gestellten Anforderungen ab, um Feinplanungsverfahren intelligent zu lenken und gleichzeitig die getätigten Benutzerinteraktionen zu beachten. Diese Aufgaben stellen hohe Anforderungen an die Performanz, da sie den gesamten Optimierungsprozess und das Bedienerlebnis der Planer direkt beeinflussen. Hinsichtlich der Skalierbarkeit soll die erwartete Reaktionszeit weitgehend vom Umfang des Feinplanungsproblems unabhängig sein.

Das Gesamtkonzept und die einzelnen Komponenten der Feinplanungsplattform werden in Unterkapitel 5.1 vorgestellt. Ein wichtiger Baustein ist die inkrementell berechnete Planbewertung, die ausführlich in 5.2 beschrieben wird. Anschließend wird in 5.3 der Planvergleich erläutert. Unterkapitel 5.4 zeigt, wie die Optimierungshinweise unter der Verwendung von Zwischenergebnissen aus der inkrementell berechneten Planbewertung abgeleitet werden. Für die Evaluierung der Feinplanungsplattform wird ein genetischer Algorithmus als verbreiteter Stellvertreter für iterative Verfahren in 5.4.2 leicht modifiziert, um die bereitgestellten Optimierungshinweise effizient verwenden zu können.

5.1 Die Feinplanungsplattform in einer dynamischen Umgebung

Die Feinplanungsplattform ist gleichzeitig eine Ausführungsumgebung für Verfahren und eine Kollaborationsplattform für Planer in einer laufenden Fertigung. Sie implementiert die drei Kernfunktionen für die Optimierung: Bewertung und Vergleich von Fertigungsplänen sowie Verfahrenssteuerung. Kernfunktionen sowie die im vorigen Kapitel ausgearbeiteten Ansätze sind in Abbildung 5.1 zusammengefasst.

Die Feinplanungsplattform kapselt die Feinplanungsverfahren und Verfahrensinstanzen und filtert deren Planvorschläge, so dass im aktuellen Fertigungsplan aus menschlicher Sicht nur intelligente Änderungen durchgeführt werden. Der Planer lenkt die Verfahren nur über die Definition der Zielfunktion und über seine Benutzerinteraktionen. Die Feinplanungsplattform entlastet ihn bei der aufwändigen Parametrierung und Ausführung der Verfahren sowie der Inspektion und der Auswahl der erstellten Fertigungspläne.

Alle Fertigungsereignisse, Änderungen in der Problemstellung und Umplanungen der Akteure (ERP, Ressourcen, Planer) fließen zunächst in den aktuellen Fertigungsplan. Der aktuelle Fertigungsplan dient somit als gemeinsames Kommunikationsmittel. Durch dieses Mittel werden die Ergebnisse der angeschlossenen Verfahren allen zugänglich.

Feinplanungssysteme sind mit der vielfältigen Verarbeitung und Verteilung von großen Datenmengen konfrontiert. Obwohl Fertigungsereignisse meistens nur sehr wenige Dispositionsobjekte betreffen, wird oft der komplette Fertigungsplan neu verarbeitet. Das mag unkritisch und ausreichend für statische Problemstellungen in einer langfristigen Planung sein, für umfangreiche dynamische Planungsprobleme ist es aufgrund der angeforderten Reakti-

onszeit ein Ausschlusskriterium. Um dies zu umgehen, können die Kernfunktionen durch Ansätze für inkrementelle Berechnung realisiert werden, die nur die veränderte Untermenge der Daten betrachten.

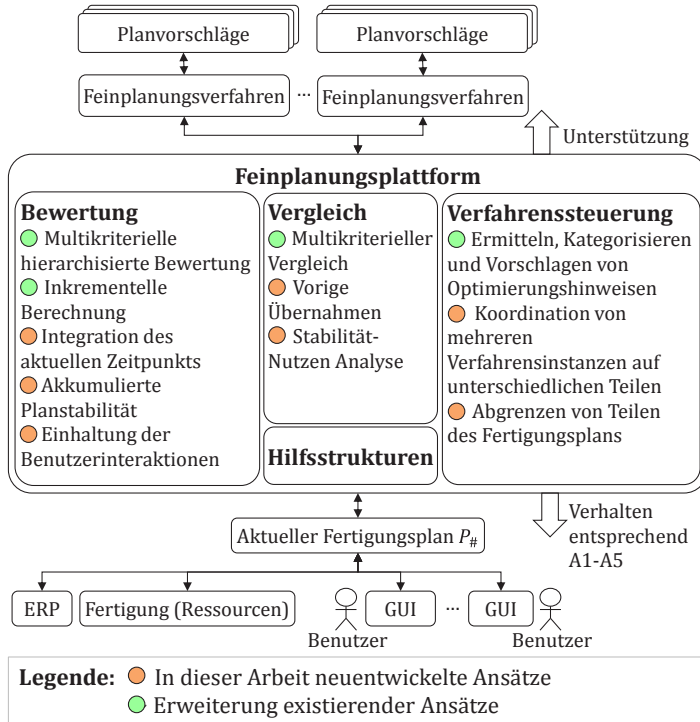


Abbildung 5.1: Einheitliches Konzept der Feinplanungsplattform und eigene Beiträge

Der Einsatz von optimierten inkrementell berechneten Datenstrukturen ist essenziell für die Fertigungssteuerung in Echtzeit, wie die nachfolgende einfache Rechnung zeigt. In einer mittleren Problemstellung mit 50 Maschinen, die ihren Fortschritt alle 5 Minuten melden, sind dies im Durchschnitt über 10.000 Meldungen täglich ohne die außerplanmäßigen Auftragsänderungen, Fertigungsstörungen oder Benutzerinteraktionen. Mit steigendem

Problemvolumen wächst die Dauer für die Verarbeitung von Ereignissen in $O(n)$. Konventionelle Planungssysteme koppeln sich deshalb von der Fertigung sowie vom ERP für gewisse Zeitintervalle ab, damit die externen Ereignisse sie nicht überfordern. Dies führt zu einer Alterung der Daten, mit denen die Planer dann nur eine mittel- und langfristige Planung durchführen können. Für die unmittelbare Fertigungssteuerung sind die Aktualisierungsintervalle zu lang.

Eine Lösung dieses Problems besteht in der inkrementellen Berechnung der verwendeten Datenstrukturen. In der Feinplanungsplattform sind neben der Bewertung weitere Hilfsstrukturen angesiedelt, die alle Feinplanungsverfahren in unterschiedlichem Grad aufbauen. Die meistverwendeten Hilfsstrukturen sind Intervall-Gebirge der Ressourcenverfügbarkeit und -belastung sowie Restriktionsgraphen (constraint graphs). Inkrementell berechnete Versionen [ChSm97] von Restriktionsgraphen wurden bereits für Verbesserungsverfahren konstruiert (vgl. 2.3.4). Vergleichsweise komplexer ist die inkrementelle Bewertung des Fertigungsplans im Zusammenhang mit der Fragmentierung. Diese wird im nächsten Unterkapitel beschrieben.

5.2 Inkrementell berechnete Bewertung des Fertigungsplans

Die inkrementelle Berechnung von Bewertungskriterien wird heute breit in betriebswirtschaftlichen und fertigungsspezifischen Analysesystemen (Business Intelligence und Manufacturing Intelligence) angewendet. Die an die Bewertung des Fertigungsplans gestellten Anforderungen liegen deutlich höher. Einerseits wird die inkrementelle Berechnung für Kriterien durch die Fragmentierung und die Integration des aktuellen Zeitpunkts deutlich komplexer. Andererseits ist eine sehr niedrige Reaktionszeit im

Vergleich zu den Analysesystemen notwendig. Nachfolgend werden diese Aspekte und die damit verbundenen Probleme analysiert und Lösungen entwickelt.

5.2.1 Inkrementell berechnete Kennzahlen

Die inkrementelle Berechnung einer Kennzahl bedeutet, dass nur die geänderten, eingefügten oder gelöschten Objekte neu evaluiert werden müssen, um die Kennzahl zu aktualisieren. Die Berechnung jeder Kennzahl besteht in der Iteration über die betroffenen Dispositionsobjekte, der daraus resultierenden Erzeugung von Zwischenwerten (intermediate data) und dem anschließenden Bilden des Kennzahlwerts.

Wie ein Beispiel in Abbildung 5.2 zeigt, wird bei der Aktualisierung von $\sum T_j$ nach einer Änderung von Auftrag 2 (z. B. C_2 oder d_2) seine vorige Verspätung subtrahiert und die neue Verspätung hinzugefügt. Bei hinzugefügten und gelöschten Aufträgen wird der erste bzw. zweite Schritt übersprungen.

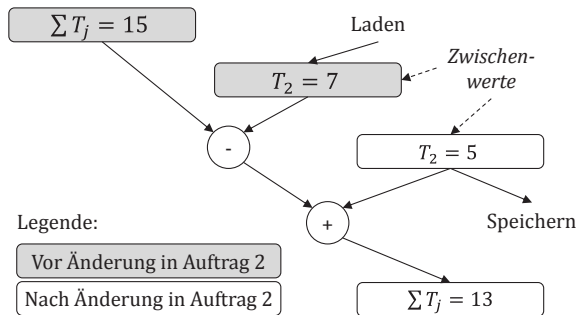


Abbildung 5.2: Inkrementell berechnete $\sum T_j$ bei Änderung von Auftrag 2

Die inkrementelle Berechnung benötigt die Zwischenwerte der betroffenen Dispositionsobjekte vor der Änderung. Die Speicherung der Zwischenwerte erspart ihre nochmalige Ermittlung. Die Zwischenwerte der neuen Version werden nach ihrer Berechnung

gespeichert und bei einer nachträglichen Änderung des Dispositionsobjekts ausgelesen. Die Reduzierung der Rechenzeit über die Speicherung der Zwischenwerte hat die Erhöhung des Speicherbedarfs zur Folge. Die Anzahl an Zwischenwerten wächst dabei linear mit der Anzahl der Dispositionsobjekte.

Etwas aufwändiger ist die Berechnung von komplexeren Kriterien wie σ_T^2 oder σ_T . Diese Art von Kennzahlen wird in drei untergeordnete Kennzahlen $\sum U_j$, $\sum T_j$ und $\sum T_j^2$ zerlegt. Diese Kennzahlen werden dann jeweils inkrementell berechnet. Anschließend werden sie zu σ_T^2 oder σ_T zusammengefasst. Neben dieser existieren auch andere Zerlegungen für optimierte rechnergestützte Berechnung [ChGL83, Finc09]. Analog sind $\emptyset DLZ$, $1/n \sum F_j$, TST , TSC und weitere (vgl. 2.2.3 und 4.2) ebenfalls inkrementell berechenbar. Einige Kennzahlen wie $\sum C_j$ können als untergeordnete Kriterien für weitere Kennzahlen wie $\emptyset DLZ$ dienen.

Die Berechnung der Zwischenwerte kann ebenfalls optimiert werden. Einfache Beispiele sind T_j , E_j und U_j , die alle auf L_j aufbauen. Somit kann L_j nur einmal für die Berechnung von T_j , E_j und U_j berechnet werden. L_j ist in der klassischen Ausführung eine einfache Differenz zweier Zeitpunkte. In der zeitbehafteten Ausführung wird sie über ein Integral berechnet.

Inkrementelle Berechnungen von Kennzahlen, die ein Minimum oder Maximum aus einer Menge darstellen, können nur in bestimmten Fällen aktualisiert werden. Solche Kennzahlen sind T_{max} und C_{max} . Falls für den gespeicherten Zwischenwert für einen Auftrag j $T_j = T_{max}$ bzw. $C_j = C_{max}$ gilt und der neue Zwischenwert kleiner ist, muss T_{max} bzw. C_{max} neu in $O(n)$ durch Iteration über alle Zwischenwerte berechnet werden. Eine Alternative bietet das Halten aller T_j bzw. C_j in einer sortierten Reihe, um bei einer Änderung das höchste T_j bzw. C_j sofort zu extrahieren. Hier

liegt der Aktualisierungsaufwand aufgrund des Einfügens bzw. Löschens in einer sortierten Reihe bei mindestens $O(\log n)$.

Mit wenigen Ausnahmefällen ist die Aktualisierung der Bewertung für alle Kennzahlen in $O(1)$ durchführbar. Der deutlich reduzierte Rechenaufwand öffnet Raum für eine umfassendere Evaluierung des Fertigungsplans. Der nächste Abschnitt zeigt die Fragmentierung und anschließende inkrementell berechnete Bewertung der Fragmente.

5.2.2 Performante Fragmentierungen

Für die Unterstützung der Feinplanungsverfahren werden Fragmente des Fertigungsplans mit Optimierungspotential aufgespürt. Theoretisch können dafür beliebige Fragmentierungen angewendet werden. Das Einheitsblatt VDMA 66412-1 [Vdma09] beschreibt sinnvolle (hierarchische) Fragmentierungen (sog. Drill Downs) für die unterschiedlichen Arten von Kennzahlen. Diese hierarchischen Fragmentierungen ermöglichen eine aus menschlicher Sicht bequeme Navigation durch die Kennzahlen der Fragmente. Dies ähnelt stark der Nutzung von elektronischen Landkarten mit unterschiedlichen Zoom-Stufen.

Die in Abbildung 4.14 dargestellte algorithmische Untersuchung auf Schwachstellen (vgl. 4.4.2) nutzt die entstandene hierarchische Struktur ähnlich wie ein Mensch. Falls sie die gleichen Fragmentierungen wie ein Planer einsetzt, ist ein menschenähnliches und dadurch nachvollziehbares Verhalten erzielbar. Eine sinnvolle nachvollziehbare Fragmentierung für die in dieser Arbeit verwendeten Kennzahlen ist der Zeitbezug (Woche, Tag, ...). Für die durchlauf- und rüstzeitbezogenen Kennzahlen sind weitere Fragmentierungen vorteilhaft (s. Tabelle 5.1).

Tabelle 5.1: Sinnvolle Fragmentierungen pro Kennzahlenart

<i>Kennzahlenart</i>	<i>Kennzahlen</i>	<i>Fragmentierung</i>
<i>Termintreue</i>	$\sum E_j, \sum T_j$	<i>Zeitbezug (Woche, Tag, ...)</i> <i>Auftrag</i>
<i>Durchlaufzeiten</i>	$1/n \sum F_j, \phi DLZ$	<i>Zeitbezug (Woche, Tag, ...)</i> <i>Maschinentyp</i> <i>Ausgangsmaterial</i> <i>Auftrag</i>
<i>Rüstzeiten</i>	<i>TST, TSC</i>	<i>Zeitbezug (Woche, Tag, ...)</i> <i>Maschinentyp</i> <i>Auftrag</i>

Für die Bewertung vieler Fragmente ist die Verwendung derselben Zwischenwerte möglich. So lassen sich die Kennzahlen der einzelnen Fragmente durch Iteration über die Zwischenwerte für den gesamten Fertigungsplan ermitteln. Wenn das zugeordnete Fragment für manche Kennzahlen eigene Zwischenwerte benötigt, werden nur diese für das Fragment gespeichert. Solche Zwischenwerte sind die Rüstzeiten von Operationen oder Durchlaufzeiten von Aufträgen, die in zwei Kalenderwochen liegen. Jeder Kalenderwoche wird nur die anteilige Rüst- bzw. Durchlaufzeit zugeteilt. Abbildung 5.3 zeigt wieder Auftrag 2 zusammen mit der anteiligen Durchlaufzeit in den unterschiedlichen Fragmenten.

Bevor überhaupt die Zwischenwerte bei einer Änderung aufgefrischt werden, ist sicherzustellen, dass ein Dispositionsobjekt nach einer Änderung noch in denselben Fragmenten liegt. Oft wechselt es bei einer Umplanung die Kalenderwoche oder den Maschinentyp. Ob das Dispositionsobjekt seine zugeordneten Fragmente wechselt oder nicht, hat keine signifikante Auswirkung auf die Performanz.

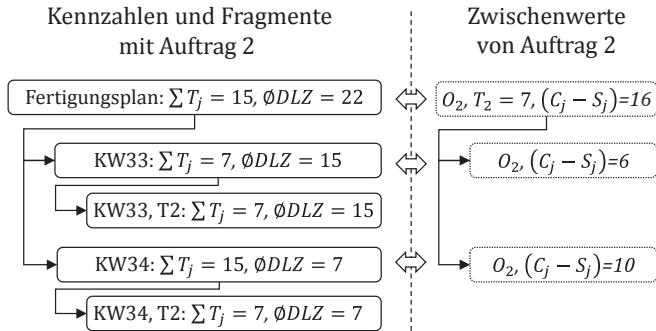


Abbildung 5.3: Zwischenwerte für Auftrag 2 entlang der Fragmenthierarchie

Die hierarchisierte Fragmentierung des Fertigungsplans stellt neue Herausforderungen an die inkrementelle Bewertung. Die vorgeschlagene Routine verringert die Rechenzeit auf Kosten eines erhöhten Speicherbedarfs. Dadurch können mehrere parallele hierarchische Fragmentierungen ohne merkbare Einbußen in der Reaktionszeit berücksichtigt werden.

5.2.3 Der aktuelle Zeitpunkt in der inkrementellen Bewertung

Wie in 4.2 gezeigt, sind die klassischen Kennzahlen um die Relevanz zum aktuellen Zeitpunkt erweiterbar. Derselbe Fertigungsplan kann bei Bewertung zu zwei unterschiedlichen Zeitpunkten unterschiedliche Kennzahlwerte bekommen. Dabei besteht einerseits die Gefahr, Fertigungspläne miteinander zu vergleichen, deren Bewertungen aufgrund des unterschiedlichen „aktuellen Zeitpunkts“ inkomparabel sind. Das würde möglicherweise zu Fehlentscheidungen in der Auswahl führen. Andererseits ist eine inkrementelle Bewertung nicht ohne weiteres möglich, da die Neuberechnung der Zwischenwerte auf einem neuen „aktuellen Zeitpunkt“ basiert.

Die Einführung eines Referenzzeitpunkts t^ρ anstelle des aktuellen Zeitpunkts t_0 ist notwendig, um eine Vergleichbarkeit zwischen Plänen, eine verlässliche inkrementelle Bewertung und realistische Aussagen über die Auswirkung von Umplanungsschritten zu gewährleisten. Der Referenzzeitpunkt t^ρ wird mit t_0 initialisiert und in bestimmten Intervallen mit Dauer d^ρ aktualisiert. Es soll $t^\rho \leq t_0$ gelten, sonst würden Operationen aus Sicht der Bewertung in der Vergangenheit liegen und deshalb eine ganz andere Bewertung bekommen. t^ρ bleibt für die Dauer von d^ρ konstant und die inkrementelle Bewertung kann Aktualisierungen der Zwischenwerte ohne negative Nebeneffekte durchführen. Nach der Neuinitialisierung des Referenzzeitpunkts sollen nur die zeitbezogenen Zwischenwerte und Kennzahlen neu bewertet werden. Die nichtzeitbezogenen Zwischenwerte bleiben unberührt.

Ein einheitlicher t^ρ ist für den Vergleich von Fertigungsplänen von besonderer Bedeutung. Besonders iterative Verfahren unterhalten eine hohe Anzahl von Fertigungsplänen. Ein Vergleich mit Fertigungsplänen aus früheren Iterationen ist auf Basis eines einheitlichen t^ρ möglich. Bei Aktualisierung von t^ρ müssen jedoch alle Planvorschläge neu bewertet werden.

d^ρ kann nicht pauschal für alle Fälle festgelegt werden. Ein sehr langes d^ρ würde sich auf den Einfluss der Relevanzkurven negativ auswirken und vor allem Fertigungspläne benachteiligen, die im unmittelbar bevorstehenden Planabschnitt besonders gut sind. Ein zu kurzes d^ρ würde dagegen die Vorteile der inkrementellen Bewertung so abmindern, dass zu wenig Zeit für die eigentliche Optimierung verbleibt, besonders bei umfangreichen Problemen.

5.2.4 Ableitung der Relevanzkurven

Das Unterkapitel 4.2 definiert eine Reihe an Kennzahlen für die Planbewertung auf Basis von Relevanzkurven. Da jedes Kriterium

eine oder mehrere Relevanzkurven benötigt und jede einzelne Relevanzkurve unzählige denkbare Ausprägungen besitzt, erhöht sich die Komplexität für die fallspezifische Definition der Kennzahlen. Ein Verfahren für die Ableitung von Relevanzkurven ist deshalb erforderlich.

Betriebe organisieren ihre Planung nach eigenen Bedürfnissen. Aufgrund der Komplexität wenden sie meistens die hierarchische Produktionsplanung (Hierarchical Production Planning, HPP) an [McSB95, McWi06, S.186ff, BaDM06, BaDi08]. Sie unterteilen die Planung oft in unmittelbare Fertigungssteuerung, kurzfristige Feinplanung und mittel- und langfristige Grobplanung. Oft wird eine Bedarfsprognose mitsimuliert. Dabei sind die Zeithorizonte für diese einzelnen Phasen auch unterschiedlich lang und hängen meistens von der Durchlaufzeit der Produkte, den dafür notwendigen Materialbeschaffungszeiten oder vom Bestellungsverlauf ab, wie zwei praktische Beispiele in Abbildung 5.4 zeigen.

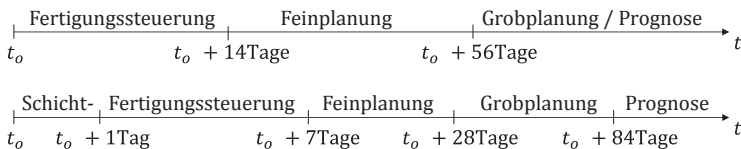


Abbildung 5.4: Zwei typische Zeithorizonte im hierarchischen Planungsprozess

Manche Betriebe erstellen eine langfristige Bedarfsprognose oft, um strategische Entscheidungen zu treffen. Hier ist ein sehr kleiner Personenkreis von Planern involviert. Die hier erstellten Aufträge werden entweder in der Grobplanung oder separat eingeplant. Im zweiten Fall spielt eine genaue Disposition dieser Aufträge keine Rolle und es zählen nur die aggregierten Bewertungskennzahlen.

In der Grobplanungsphase untersuchen die Planer unterschiedliche Fertigungsarten und Reihenfolgen. Die dort disponierten

Aufträge sind oft nur für alle Planer sichtbar, weshalb sie hier Umplanungen nur untereinander diskutieren.

In der Feinplanungsphase dagegen entstehen oft die Schichtpläne und externe Materialbelieferungen werden angeordnet. Der dabei involvierte Personenkreis ist hier größer und enthält oft Lagerverantwortliche, Verantwortliche für die Materialwirtschaft und Schichtmeister. Eine Planveränderung wirkt sich hier gelegentlich auf die Materialbestellungen oder auf die Personalplanung aus. In dieser Phase sichern oft Vertriebsmitarbeiter Kundentermine zu, weshalb Umplanungen hier eine gewisse Nervosität verursachen können.

Die Fertigungsverantwortlichen und die Planer versuchen den Fertigungsplan für die unmittelbare Fertigungssteuerung stabil zu halten, da ein großer Teil des Betriebs bei seiner Ausführung involviert ist. Jede Änderung der Ausführungszeiten oder der Ressourcen kann zu Materialumlagerung oder Maschinenumrüstung führen oder erfordert neue Personalanweisungen. Viele Veränderungen akkumulieren sich und führen zu Nervosität in der Fertigung.

Manchmal wird dem Schichtmeister ein genügend großer Spielraum für Änderungen überlassen. In diesem Fall kann der Schichtmeister Reihenfolgenänderungen innerhalb der Schicht durchführen, ohne sich dabei mit den Planern abzustimmen.

Diese Phasen haben in der Praxis keine festen Grenzen und es kann auch vorkommen, dass Planer in der Grobplanung bestimmte Aufträge feinplanen und Materialbestellungen auslösen. Der Phasenübergang ist fließend. In jeder Phase werden andere Ziele verfolgt. Aus der fallspezifischen Kenntnis, welches Kriterium wie wichtig in jeder Phase ist, können die einzelnen Gewichtungen abgeleitet werden.

Unter Kenntnis der Gewichtung der einzelnen Kriterien in den Planungsphasen kann die Relevanzfunktion auf unterschiedliche Weise abgeleitet werden. Aus der Menge von Zeithorizonten (hier in Zeitpunkte umgewandelt) samt Gewichtung $\{(t_i, w_i)\}$ kann die Relevanzfunktion mithilfe einer Treppenfunktion, einer stückweise linearen Funktion oder durch einen monotonen kubisch hermetischen Spline interpoliert werden. Somit sind alle Teilfunktionen entweder konstant oder monoton. Die Anforderung an die Gewichtungsfunktion $g(t)$ ist minimalistisch. Sie muss in $[t_o, +\infty[$ integrierbar sein. Für manche Kennzahlen wie die Termintreue ist allerdings eine Integrierbarkeit in $] -\infty, +\infty[$ erforderlich, da in einer laufenden Fertigung der späteste Fertigstellungszeitpunkt eines Auftrags j vor dem aktuellen Zeitpunkt liegen kann ($d_j < t_o$).

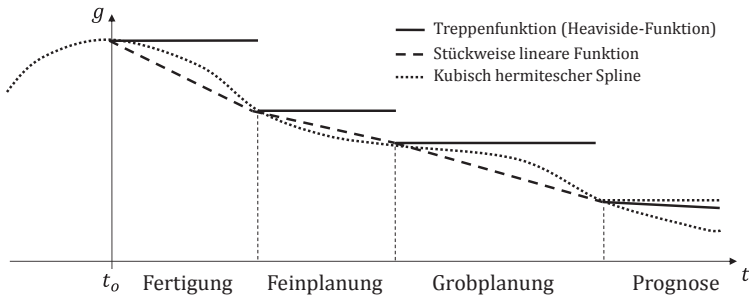


Abbildung 5.5: Arten von Relevanzfunktionen

Unabhängig von der Konstruktionsart entsteht eine Funktion, definiert zwischen dem ersten und letzten Zeitpunkt (t_1 und t_k). Für die Intervalle $] -\infty, t_1]$ und $[t_k, +\infty[$ kann je nach Bedarf eine beliebige Funktion angehängt werden, wie z. B. die konstante Funktion. Wenn die Konstruktionsart es zulässt, kann auch die Glockenkurve $e^{-\frac{1}{2}\left(\frac{t-t_o-\mu}{\alpha}\right)^2} / \alpha\sqrt{2\pi}$ an t_1 und t_k durch Anpassung von α und μ so angehängt werden, dass $g(t)$ differenzierbar in t_1 und t_k wird.

Die Funktion $g(t)$ soll so ausgewählt werden, dass in einer laufenden Fertigung die Bewertung der Dispositionsobjekte sich nicht sprunghaft ändert, wenn die Zeit voranschreitet und diese Dispositionsobjekte dann teilweise in eine frühere Phase wandern. Auch aus diesem Grund bewerten Unternehmen manche Kennzahlen wie die $\emptyset DLZ$ je nach Schichtplanung unter Ausschluss von Sonn- und Feiertagen oder des gesamten Wochenendes. Für diese Fertigungspause lässt sich dann $g(t)$ auf 0 setzen. Obwohl $g(t)$ dann nicht mehr stetig ist, ist die DLZ von diesen nichtproduktiven Zeitintervallen bereinigt und somit näher an der Netto- DLZ . Unternehmen mit kurzen Aufträgen haben einen genauen Wert für $\emptyset DLZ$.

Analog zu den Relevanzfunktionen für die Kennzahlen sind die in $]-\infty, t_0]$ definierten Relevanzfunktionen g^u und g^b aus Abschnitt 4.2.3 bzw. 4.2.4 abgeleitet. Die Menge $\{(t_i, w_i)\}$ ist weniger von der Fertigung abhängig, sondern eher von den Benutzerwünschen. So können die einzelnen Phasen aus Abbildung 4.5 und Abbildung 4.8 entnommen werden. Die Gewichtung dagegen muss benutzer-spezifisch erfolgen.

Mit der Definition der Relevanzkurven ist das Konzept der inkrementellen Bewertung des Fertigungsplans und seiner Fragmente vollständig. Das nachfolgende Unterkapitel beschäftigt sich mit dem Vergleich auf Basis der Kennzahlenwerte.

5.3 Vergleichsfunktion für Fertigungspläne und Übernahme von neuen Lösungen

Aufbauend auf den Schlussfolgerungen in Unterkapitel 4.3 erfolgt hier die Detaillierung der Vergleichsfunktion und der Übernahmelogik.

Die Planung als multikriterielles Problem stellt komplexe Anforderungen an die Vergleichsfunktion. Aufgrund der angestrebten Selbstständigkeit und Reaktionsfähigkeit in der Optimierung soll die Feinplanungsplattform eindeutig über die Übernahme eines Fragments oder eines ganzen Fertigungsplans in den aktuellen Fertigungsplan entscheiden können. Das ist notwendig, da die Übernahmeanfragen wegen der verschiedenen permanent laufenden Instanzen und den gewünschten inkrementellen Verbesserungen im Fertigungsplan hochfrequent sind und deshalb nicht den Benutzern überlassen werden können.

Die Verfolgung der Optimierungsziele soll aus Sicht des Anwenders unterschiedlich stattfinden. Manche Zielfunktionen sind eigentlich harte Restriktionen (hard constraints) und bestimmen über die Zulässigkeit des Fertigungsplans. Wenn die harten Restriktionen nicht erfüllt sind, wird der Fertigungsplan mit dem höheren Erfüllungsgrad dieser Restriktionen bevorzugt. Harte Restriktionen sind beispielsweise die Bedingungen $\sum T_j = 0$ oder die volle Einhaltung aller Reihenfolgenabhängigkeiten zwischen Operationen unter Berücksichtigung von maximal zulässigen Zeitspannen. Die Nichteinhaltung der Zeitspannen kann z. B. zu Materialverlust führen. Da nicht alle Verfahren diese Zeitspannen betrachten, findet die Prüfung für deren Einhaltung erst zum Bewertungszeitpunkt statt.

Falls zwei Fertigungspläne alle harte Restriktionen erfüllen, werden die weiteren Zielfunktionen in der Zielsetzung betrachtet. Die Möglichkeit, eine selbstständige Entscheidung im multikriteriellen Fall zu treffen, kann mithilfe von Gewichtung und Globalem Kriterium (GK) [Miet99, S.67ff, GuSi02, LoWH05], gebildet aus den unteren Schranken, geschaffen werden (vgl. 2.2.3 und 2.2.4). Diese Ansätze haben ihre Schwächen, sind aber für die Planer verständlich. Dadurch sind die Vergleichs- und Übernahmeentscheidungen nachvollziehbar. Diese Methoden werden auch für den Einhaltungsgrad bei mehreren harten Restriktionen verwendet.

Neben der verfolgten Optimierungsziele sind auch die zulässigen und unproblematischen Grenzwerte der beiden neuen Kennzahlen θ und ϑ^b bei einer Übernahme zu berücksichtigen (vgl. 4.3). Es kann durchaus vorkommen, dass keine Fertigungspläne mit Kennzahlenwerten innerhalb des jeweiligen zulässigen GW existieren. Dies tritt z. B. bei einer Anhäufung von Übernahmen oder Benutzerinteraktionen auf. Da diese zwei Kennzahlen zeitbezogen sind, würden sich nach gewisser Zeit (abhängig von den eingestellten Relevanzkurven) beide Kennzahlenwerte stark verringern (vgl. 5.2.4) und einen Raum für neue Übernahmen öffnen.

Bei der Bildung eines globalen Kriteriums aus θ und ϑ^b ist eine Anpassung von ϑ^b durch Multiplikation mit $|O|/|b_k|$ erforderlich, da sich ϑ^b nur auf einen sehr kleinen Anteil der Operationen in der Problemstellung bezieht. Je weniger Benutzerinteraktionen durchgeführt wurden, desto mehr wirken sich ihre Verletzungen auf den angepassten Wert von ϑ^b aus. Das ist durchaus ein gewünschter Nebeneffekt, weil der Planer bei weniger getätigten Interaktionen ihre Einhaltung höher einfordert.

Der zusammengefasste Ablauf des Vergleichs und der Übernahme von Fragmenten und Fertigungsplänen ist in Abbildung 5.6 dargestellt. Beim Vergleich von Fragmenten werden die Kennzahlenwerte des aktuellen Fertigungsplans verwendet, die sich nach einer hypothetischen Übernahme des jeweils zu vergleichenden Fragments ergeben. Die Ermittlung dieser Kennzahlenwerte ist durch die Veränderungen im Fragment mithilfe der inkrementell berechneten Bewertung einfach durchführbar. Eine Entscheidung über eine Übernahme wird erst getroffen, wenn der aktuelle Fertigungsplan nach den weiteren Zielkriterien verbessert wird.

Alle Zielfunktionen und ihre Gewichtungen sind für die klassische Ausführung der Bewertungskriterien geeignet. Im Falle einer multikriteriellen Zielsetzung mit zeitbezogenen Bewertungskriterien agieren die Relevanzkurven möglicherweise als versteckte

Gewichtung, weshalb die zeitbezogenen Ausführungen zusätzlich angepasst werden müssen. Es muss sichergestellt werden, dass für alle Relevanzkurven $g(t)$ die Gleichung $\int_{t_0}^{t_0+d_h} g(t)dt = d_h$ gilt, wobei d_h den typischen Zeithorizont in der Problemstellung darstellt. Andernfalls müssen die zeitbehafteten Ausführungen mit $d_h / \left(\int_{t_0}^{t_0+d_h} g(t)dt \right)$ multipliziert werden. Diese Art von Normierung dämpft die Auswirkungen aus den verschiedenen Relevanzkurven.

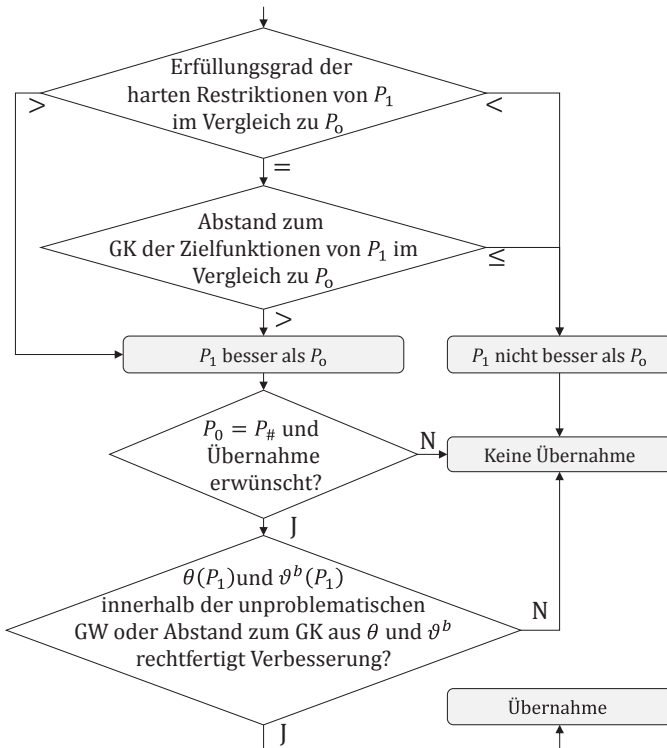


Abbildung 5.6: Vergleich und Übernahme von Fertigungsplänen

Im Falle von mehreren Verfahrensinstanzen (vgl. 4.4.3) werden Fragmente unabhängig voneinander übernommen. Falls die

einzelnen Verfahrensinstanzen mehrere Lösungen für die Fragmente halten und die Zielsetzung harte Restriktionen enthält, ist der Vergleich über die Zeit nicht eindeutig. Sobald alle harten Restriktionen erfüllt sind, verschiebt sich der Fokus auf die weiteren Optimierungsziele. Die Verfahrensinstanzen müssen in diesem Moment bei Bedarf die enthaltenen Lösungen nochmals untereinander vergleichen.

Die Konfiguration zur Durchführung von Vergleich und Übernahme hängt von vielen Parametern ab, die zum großen Teil aus der Problemstellung ableitbar sind. Ein subjektiver Aspekt bleibt der Einhaltunggrad von θ und ϑ^b , der nur benutzerspezifisch erfolgen kann.

5.4 Optimierungshinweise für die Unterstützung der Feinplanungsverfahren

Die Optimierungshinweise sind Fragmente im Fertigungsplan mit unterschiedlicher Größe, die ein Optimierungspotential besitzen (vgl. 4.2 und 4.4.1). Das Optimierungspotential leitet sich aus der Bewertung der einzelnen Fragmente ab. Nachfolgend wird gezeigt, wie die Zwischenwerte aus der inkrementellen Bewertung für die Extraktion von Optimierungshinweisen verwendet werden können und wie mit ihrer Hilfe im Restriktionsgraph des Fertigungsplans Fragmente für Verfahrensinstanzen (vgl. 4.4.2 und 4.4.3) markiert werden.

5.4.1 Extraktion von Optimierungshinweisen

Das Hauptprodukt der Bewertung ist eine Fragmenthierarchie, deren Knoten (Fragmente) mit verschiedenen Kennzahlenwerten versehen sind. Als Nebenprodukt entsteht eine Menge von Zwi-

schenwerten. Diese beiden Datenstrukturen sind stark miteinander vernetzt und werden inkrementell gepflegt (s. Abbildung 5.7a). Einige der berechneten Kennzahlen und Zwischenwerte spiegeln das theoretische Optimierungspotential der Fragmente sehr genau wider (vgl. 4.2.6).

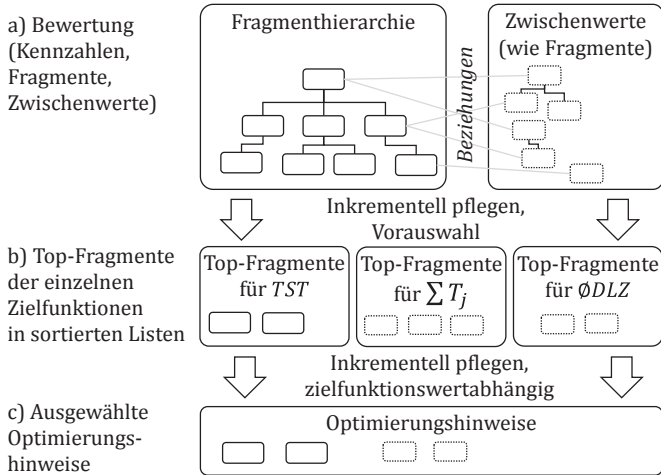


Abbildung 5.7: Extraktion von Optimierungshinweisen

Für jede einzelne Zielfunktion wird jeweils eine sortierte Folge für Top-Fragmente erstellt (s. Abbildung 5.7b). Die Top-Fragmente sind Fragmente oder Zwischenwerte mit einem überdurchschnittlich hohen Optimierungspotential. Die Zwischenwerte bilden zwar die Fragmentbewertung, können aber auch als atomare Fragmente angesehen werden. Die Bewertung frischt inkrementell diese sortierten Folgen zusammen mit den Kennzahlenwerten auf. Die sortierten Folgen enthalten nur eine beschränkte Anzahl an Fragmenten und Zwischenwerte, die ein überdurchschnittlich hohes Optimierungspotential haben.

Im letzten Schritt werden die Optimierungshinweise extrahiert (s. Abbildung 5.7c). Dies hängt stark vom Erhaltungsgrad der gesetz-

ten Zielfunktion ab. Falls die harten Restriktionen (vgl. 5.3) noch nicht erfüllt sind, werden an erster Stelle Top-Fragmente aus den entsprechenden Kennzahlen ausgewählt. Sobald die harten Restriktionen erfüllt sind, werden die Optimierungshinweise aus den weiteren Kennzahlen extrahiert. So verbessert sich die zielgerichtete Unterstützung für die Verfahren.

5.4.2 Markierung von Fragmenten

Während der Konzipierung der Feinplanungsplattform wurde in Abschnitt 4.4.2 auf die Möglichkeit eingegangen, Bereiche des Fertigungsplans zu markieren und eigenständigen Verfahrensinstanzen zur exklusiven Optimierung zuzuweisen. Nachfolgend wird dieses Vorgehen vertieft behandelt.

Die Dekomposition von Fertigungsplänen wird oft angewendet, um einen Zeitabschnitt oder eine einzelne Maschine des Fertigungsplans zu extrahieren und sie/ihn dann separat zu optimieren [AyKU03, Bran93, MSPF07, OvUz97]. Beispiel dafür sind Verfahren mit einem rollenden Horizont (rolling horizon), die über die gesamte Planlänge vom aktuellen Zeitpunkt bis zum Ende des Fertigungsplans schrittweise durchgehen und optimieren [FaXi97, OvUz95]. Dabei ist die Schrittweite beim Durchgehen des Fertigungsplans generell deutlich kürzer als die Planlänge selbst. Während die erwähnten Methoden die Planlänge unabhängig von seinem Optimierungspotential oder Optimierungsbedarf auswählen, nimmt die Feinplanungsplattform unter Verwendung der Ergebnisse aus der Bewertung eine intelligente Auswahl eines sowohl zeit- als auch maschinenbegrenzten Abschnitts des Fertigungsplans vor.

Als Einstieg für die Markierung dienen die Fragmenthierarchie, die Optimierungshinweise sowie die Zwischenwerte der inkrementellen Bewertung (vgl. auch 5.2). Die vorher in Abbildung 4.15 präsentierte Markierung erfolgt schrittweise. Ausgehend von den

Optimierungsstellen, deren Nachfolgern, Vorgängern und den alternativen Maschinen wird für den angestrebten Markierungsbereich eine zeitliche und maschinenabhängige Orientierung abgeleitet. Als Grundlage für die Markierung dient ein inkrementell berechneter Restriktionsgraph, der hier um die Kanten von Operationen auf zugeordneten Maschinen sowie auf alternativen Maschinen erweitert wird. Die Maschinen selbst sind dabei als Knoten dargestellt.

Die Markierungsprozedur wird mit der Markierungsorientierung und den Optimierungsstellen als den ersten markierten Operationen initialisiert. Um den Markierungsbereich auszuweiten, werden unterschiedliche Wege im Restriktionsgraph verfolgt. Es werden Nachfolger und Vorgänger von Operationen auf den zugeordneten Maschinen (s. Abbildung 5.8a) markiert, die zur Markierungsorientierung passen. Oder es werden Nachfolger und Vorgänger innerhalb des Auftrags nach den gleichen Kriterien markiert (b).

Etwas aufwändiger geschieht die Markierung von Operationen auf einer alternativen Maschine (s. Abbildung 5.8c). Hier wird über die zusätzlichen Kanten auf die alternative Maschine zugegriffen und die Operationen in der Markierungsorientierung werden markiert. Für die performante Selektion von Operationen auf einer Maschine und in einem bestimmten Zeitintervall werden pro Maschine inkrementell gepflegte Intervallbäume (interval tree) verwendet [BKOS00, S.212–217].

Den oberen Schritten folgend ist es nicht möglich, Maschinen ohne Operationen zu markieren. Um dies zu erreichen, werden Platzhalter eingeführt, die im Restriktionsgraph als Knoten eingefügt werden (s. Abbildung 5.8d) und bei Bedarf mit Operationen auf diesen Maschinen mittels Vorgänger-Nachfolger Kanten verbunden sind. Die Dauer der Platzhalter entspricht der längsten Rüstzeit auf der Maschine oder 0, falls die Maschine keine reihenfol-

genabhängigen Rüstzeiten erfordert. Die Platzhalter werden nicht markiert sondern als angrenzende Operationen gekennzeichnet.

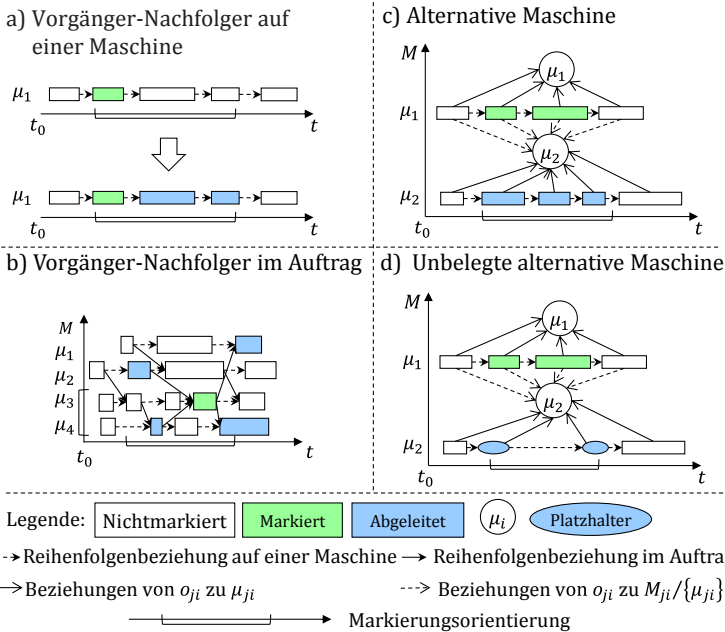


Abbildung 5.8: Möglichkeiten zur Markierungsausbreitung

Diese Schritte werden solange ausgeführt, bis in der Markierungsorientierung alle Operationen markiert sind oder angrenzende Operationen von anderen Markierungen erreicht sind. In einem weiteren Schritt werden Operationen markiert, die überwiegend Beziehungen zu schon markierten Operationen haben, damit keine Lücken im Markierungsbereich entstehen. Als nächstes wird der Markierungsbereich mit der Markierungsorientierung verglichen und die letztere kann sich noch ausbreiten oder schrumpfen. Im ersten Fall werden weitere Markierungsschritte ausgeführt, falls der Markierungsbereich nicht eine vorgegebene Größe erreicht hat.

Mit dem Kennzeichnen aller angrenzenden Operationen ist die Markierungsprozedur beendet. Ein Feinplanungsverfahren darf Operationen so lange umplanen, bis es Restriktionen zu angrenzenden Operationen verletzt. Ihm stehen die Zeitintervalle zwischen zwei angrenzenden Operationen auf den Maschinen zur exklusiven Nutzung zur Verfügung.

5.5 Anbindung eines von Optimierungshinweisen unterstützten Feinplanungsverfahrens

Das Konzept der Feinplanungsplattform sieht die Anwendung verschiedenartiger Feinplanungsalgorithmen vor. Für den vollwertigen Einsatz der Feinplanungsplattform soll neben der Bewertung und Gegenüberstellung von Fertigungsplänen auch die Optimierung mittels Feinplanungsverfahren erfolgen. Die eingesetzten Feinplanungsverfahren sollen von möglichst vielen Funktionsbausteinen der Feinplanungsplattform profitieren können. Während viele Verfahren auf externe Bewertung oder den Vergleich von Fertigungsplänen zugreifen können, sind solche mit integrierter Betrachtung von (externen) Optimierungshinweisen eine Minderheit. Als erstes soll aus der engeren Auswahl praxistauglicher Verfahren ein geeignetes selektiert werden. Bei Bedarf wird dieses Verfahren in einem folgenden Schritt angepasst. Für die prototypische Implementierung und Anbindung an der Feinplanungsplattform ist ein Verfahren erforderlich, das alle Funktionsbausteine nutzt, um sie auch evaluieren zu können.

5.5.1 Auswahl eines geeigneten Feinplanungsverfahrens

Für die Auswahl eines Feinplanungsverfahrens werden vorerst alle relevanten Kriterien aufgezählt. Das Verfahren muss mög-

lichst viele der von der Feinplanungsplattform angebotenen Funktionsbausteine abdecken und für dynamische Problemstellungen ausgelegt sein. Von den Funktionsbausteinen ist die Unterstützung von Optimierungshinweisen von besonderer Bedeutung. Weitere wichtige Kriterien sind die erwiesene Praxistauglichkeit, die Abbildung praktischer Problemstellungen und der dafür benötigte Konfigurationsaufwand.

Wie die Unterkapitel 2.3 und 2.4 zeigen, existiert eine breite Palette von Verfahren, von denen nur ein kleiner Teil seinen Weg in kommerzielle Anwendungen gefunden hat (vgl. 2.6). Diese praxistauglichen Verfahren sind der Giffler-Thompson Algorithmus samt unterschiedlicher Prioritätsregeln (GT), der genetische Algorithmus (GA) sowie mathematische Verfahren (MO) für ganzzahlige Optimierung. Diese drei Verfahrensarten wurden in Tabelle 5.2 anhand der oberen Kriterien bewertet. Da (metaheuristische) lokale Suchverfahren (LS) sich von Natur aus besonders gut für dynamische Problemstellungen eignen und einen für die Planer nachvollziehbaren Optimierungsablauf aufweisen, wurden auch sie in die Auswahl miteinbezogen.

Tabelle 5.2 bewertet jedes dieser vier Verfahren anhand der Erfüllung der Kriterien und zeigt, dass nur der GA alle fünf Kriterien ausreichend (oder besser) erfüllt. Die lokalen Suchverfahren scheiden aufgrund der unzureichenden Praxiserfahrung und der ungenügenden Abbildung realer Problemstellungen aus. Die Wahl fällt deshalb auf den GA.

Im Forschungsumfeld werden z. Z. vielversprechende Algorithmen wie die aktionsbasierten Verfahren oder die Ant-Colony-Optimization untersucht. Diese Verfahren bieten „im Labor“ für ausgewählte Problemstellungen sehr gute Ergebnisse, sind jedoch relativ jung im Vergleich zu den Genetischen Algorithmen und es mangelt an Erkenntnissen aus dem realen Einsatz, weshalb sie aus der Auswahl ausgeschlossen wurden.

Tabelle 5.2: Abdeckung gestellter Kriterien von oft eingesetzten Feinplanungsverfahren

<i>Kriterium</i>	<i>GT</i>	<i>GA</i>	<i>MO</i>	<i>LS</i>
<i>Nutzung von Plattformfunktionen</i>	-	+	--	+
<i>Dynamische Problemstellung</i>	--	o	o	++
<i>Praxiserfahrung</i>	++	+	+	-
<i>Abbildung realer Problemstellungen</i>	++	++	o	-
<i>Konfigurationsaufwand</i>	+	o	--	o
<i>Niedrigster Erfüllungsgrad aus den oberen fünf Kriterien</i>	--	o	--	-

Legende: GA – genetische Algorithmen, GT – Giffler-Thompson Algorithmus, LS – Lokale Suchverfahren (Iterative Repair, Simulated Annealing, Tabu Search), MO – Mathematische Optimierungsmethoden (Branch-and-Bound, Simplex);
 ++sehr gut, +gut, o ausreichend, -schlecht, --sehr schlecht

Es bleibt, die genaue Art von GA zu bestimmen. GA für die Feinplanung unterscheiden sich hauptsächlich in der Repräsentation des Fertigungsplans im genetischen Metamodell. Tabelle 5.3 listet die unterschiedlichen Kriterien der meistverwendeten Repräsentationen. Die Benotung erfolgt nach der Beschreibung in 2.3.5.

Der GA wird Optimierungshinweise erhalten, die aus Operationen und Kennzahlenwerten bestehen. Der GA soll diese Operationen in der Repräsentation identifizieren können, falls er die Optimierungshinweise einschlägig nutzen will. Auch soll der GA einen Fertigungsplan möglichst stabil halten. Dies ist ohne eine genaue Abbildung der Fertigungspläne in der Repräsentation nicht realisierbar. Deshalb sind die ersten beiden Kriterien in Tabelle 5.3 Ausschlusskriterien.

Tabelle 5.3: Unterschiedliche Repräsentationen eines GA

<i>Kriterium</i>	<i>K</i>	<i>B</i>	<i>PR</i>	<i>OR</i>	<i>SR</i>
<i>Operationen identifizierbar</i>	--	o	++	++	++
<i>Abbildungsgenauigkeit</i>	--	+	o	++	++
<i>Konvergenzzeit</i>	++	o	+	o	--

Legende: K-Parameterierungsraum von Konstruktionsverfahren;
 B – binäre, PR – Prioritätsregelkodierung,
 OR – Operationsreihenfolge samt Maschine,
 SR – Operationsstartzeitpunkt samt Maschine;
 ++sehr gut, +gut, o ausreichend, –schlecht, --sehr schlecht

Falls der GA den geplanten Startzeitpunkt mitkodiert (SR), ist der denkbare Lösungsraum sehr groß, da der GA auch unzulässige Pläne erstellt. Aufgrund der höheren Performanz ist die Kodierung der Reihenfolge der Operationen samt Maschine (OR) am besten für die prototypische Implementierung geeignet. Diese Art von GA erfüllt alle geforderten Kriterien und kann auch alle Funktionsbausteine der Feinplanungsplattform verwenden, wie das nachfolgende Unterkapitel zeigt.

5.5.2 Anbindung des ausgewählten genetischen Algorithmus an die Feinplanungsplattform

Die ausgewählte Art von GA wird auf die Nutzung der Funktionen der Feinplanungsplattform angepasst. So findet die Vergleichsfunktion Anwendung in der Selektion im genetischen Ablauf (s. Abbildung 2.18). Die Bewertungsfunktion dagegen berechnet die Fitness der einzelnen Individuen in der Evaluierungsphase. Es ist lediglich die Verwendung von Optimierungshinweisen in den GA zu integrieren.

Für die ausgewählte Repräsentationsart führt der GTA die Evaluierung der Individuen durch (vgl. 2.3.5). Dabei verwendet er die Reihenfolge der Gene als Prioritätsnummer für die Erstellung des Fertigungsplans (s. Abbildung 5.9, grüner Pfeil). Jedes Gen bildet eine Operation mit der zugeordneten Maschine ab und wird durch ($Operationsindex$
 $Maschinenindex$) kodiert. In Einzelfällen kodiert das Gen auch die zusätzlichen Ressourcen zur Operation, wenn sie für die Planstabilität relevant sind. Der Index der verwendeten Kombination aus der für die Operation zulässigen Kombinationen ($\subseteq \mathcal{P}(R)$) wird neben dem Maschinenindex eingefügt. Um die Auswahl dieser Kombinationen sowie der Maschine zu ermöglichen, werden zu jeder o_{ji} die zulässigen Maschinen- und Ressourcenkombinationssindizes gespeichert.

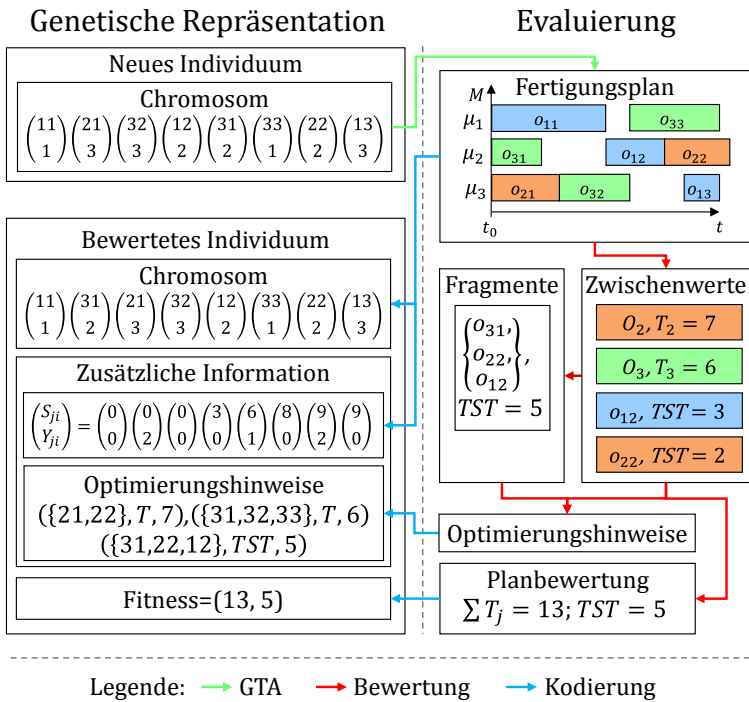


Abbildung 5.9: Evaluierung eines Individuums im GA [DiBa11]

Nach der vollständigen Konstruktion des Fertigungsplans durch den GTA ermittelt die Bewertung der Feinplanungsplattform die Zielfunktionswerte und die Optimierungshinweise (s. Abbildung 5.9, rote Pfeile). Diese Information soll in das Individuum einfließen.

Im nächsten Schritt wandern die Zielfunktionswerte in die Fitness des Individuums (s. Abbildung 5.9, blaue Pfeile). Die Repräsentation (das Chromosom) zusammen mit der Fitness ist ausreichend für die Reproduktion und den Vergleich mit anderen Fertigungsplänen. Abweichend vom klassischen GA wird hier das Individuum um einen informativen Teil erweitert. Er enthält die Optimierungshinweise, umgewandelt in einer vereinfachten Form der Art (*{Operationsindex, ...}, Hinweisart, Optimierungspotential*).

In diesem Schritt fließen auch der S_{ji} und der Rüstzustandsindex Y_{ji} für jede o_{ji} in das Individuum ein. Maschinen mit (reihenfolgenabhängigen) Rüstzeiten werden umgerüstet, falls die Vorgängeroperation die Maschine in einem anderen inkompatiblen Rüstzustand verlassen hat. Je nach Fertigungsart wird der Rüstzustand beispielsweise durch die Granulatfarbe, das NC-Programm, die eingebauten Werkzeuge oder weitere Einstellparameter beschrieben. Jedem Rüstzustand in der Problemstellung wird ein eindeutiger Rüstzustandsindex zugeordnet. Wenn die Einstellparameter für o_{ji} und o_{jk} übereinstimmen, gilt $Y_{ji} = Y_{jk}$. Auf Maschinen ohne reihenfolgenabhängige Rüstzeiten gilt immer $Y_{ji} = 0$. Der Einsatz von S_{ji} und Y_{ji} wird im nachfolgenden Abschnitt genauer erläutert.

Nach der Abbildung des erstellten Fertigungsplans auf das Chromosom kann das neue Chromosom sich vom Ausgangschromosom unterscheiden (s. Abbildung 5.9). Der GTA erzeugt nur aktive Fertigungspläne und korrigiert implizit Unstimmigkeiten wie eine unplausible Reihenfolge der Operationen desselben Auftrags im Chromosom. Auch wenn die beiden Chromosomen denselben

Fertigungsplan repräsentieren, ist die Version nach der Evaluierung zu bevorzugen, da sie bidirektional zum Fertigungsplan abbildbar ist (vgl. 2.3.5) und aus ihr eine genauere Schlussfolgerung über die tatsächliche Reihenfolge der Operationen gezogen werden kann.

Nach der Evaluation und Selektion enthält jedes Individuum eine akkurate Repräsentation des Fertigungsplans samt Bewertung und Optimierungshinweisen sowie viele Aspekte der Problemstellung. Die Kreuzung und die Mutation sind die Phasen in einem GA, die eine Erzeugung bzw. Veränderung in den Individuen durchführen. Neue Arten dieser Operatoren sollen gezielt auf Basis der Optimierungshinweise Individuen verändern bzw. erzeugen. Wie in der Gentechnik sollen mit wenigen Individuen gezielte Genmanipulationen durchgeführt werden, damit das neu kreierte Individuum besser als seine Elternteile ist. Sobald einmal das Individuum verändert wurde, sind die Optimierungshinweise nicht mehr aktuell und eine gezielte Veränderung kann aus diesem Grund scheitern. Deshalb ist der Einsatz der neuen Operatoren nur direkt nach der Selektionsphase sinnvoll. Nach der Ausführung der neuen Operatoren sind weiterhin die klassischen Operatoren anwendbar. Daraus ergibt sich der Ablauf in Abbildung 5.10. Dabei sollte nur ein Anteil der Individuen durch die neuen Operatoren gezielt manipuliert werden, damit der GA weiterhin nach seinem bewährten Schema abläuft, und nicht vorzeitig in einem lokalen Optimum hängen bleibt.

Als klassische Kreuzungsoperatoren werden der N-Point Crossover und der PPX-Crossover eingesetzt (vgl. 2.3.5). Die Mutation dagegen führt eine zufällige Verschiebung eines Anteils der Gene in der geordneten Reihe und/oder eine zufällige Veränderung oder Entfernung der Maschinenindizes durch. Im letzten Fall wählt der GTA eine Maschine während der Konstruktion aus. Diese Operatoren sind die am häufigsten eingesetzten Varianten in der Feinplanung [BiMa99, PeMC08, VaBQ08, Wall96].

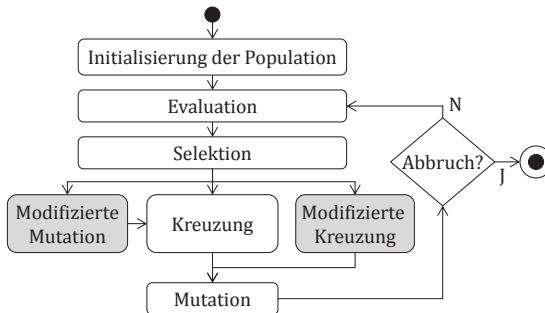


Abbildung 5.10: Ablauf des modifizierten GA

Die Selektion erfolgt nach der „Tournament“-Strategie, wobei das beste Individuum immer erhalten wird (Elitismus) [PeMC08, Wall96]. Bei der „Tournament“-Strategie werden besser bewertete Individuen mit höherer Wahrscheinlichkeit in der Population behalten.

Die Ergebnisse eines GA-Laufs sind stark von seiner initialen Population abhängig, die in der Feinplanung typischerweise 10 bis 100 Individuen enthält. Je besser und vielfältiger sie ist, desto schneller konvergiert der GA und desto bessere Endergebnisse liefert er auch. Deshalb wird die Population mit ca. 40 Prioritätsregeln (darunter EDD, SPT, Slack, EST, CoverT, RT, FIFO, WING, LING, RANDOM) und verschiedenen Werten des „look-ahead“ Parameters initialisiert (vgl. 2.3.3). Jeder Lauf liefert einen brauchbaren Fertigungsplan, der mindestens genauso gut ist wie die beste vom GTA angewendete Prioritätsregel.

Neben den aus der Forschung bekannten Prioritätsregeln [BIPH82, Haup89, Mose93] wurde speziell eine Rüstoptimierungsregel implementiert, die für die Einplanung der Operation mit der niedrigsten Rüstzeit sorgt. Obwohl die neue Regel den Fertigungsverlauf hinsichtlich Termintreue und Durchlauf sehr suboptimal plant, bilden die erstellten rüstoptimierten Sequenzen von Operationen gute Gene für spätere Kreuzungen.

Um die Beschreibung des GA zu vervollständigen, werden im nächsten Abschnitt die neuen Operatoren erörtert, die die Optimierungshinweise aus der Bewertung für gezielte Manipulationen nutzen.

5.5.3 Operatoren für „genetisches Engineering“

Die genetischen Algorithmen sollen die Evolution durch zufällige Kreuzung und Mutation nachbilden. Beobachtungen zufolge werden neuere Arten dieser Operatoren vorgeschlagen. Ein Großteil dieser Operatoren ist auf bestimmte Problemklassen und Repräsentationen fokussiert. Zudem ersetzt das sog. „genetische Engineering“ immer öfter den Zufall in den Operatoren, um eine schnelle Konvergenz in der Optimierung zu erreichen. In der Feinplanung werden z. B. gute Individuen auf Übereinstimmungen untersucht, die die Kreuzung in die nächste Generation übertragen soll [CCFC08, WalG05]. Auch werden gezielte Veränderungen in der Mutation durchgeführt, um verspätete Aufträge höher zu priorisieren, damit diese früher im Fertigungsplan starten [DoGi94]. Die hier entworfenen Operatoren sollen die Optimierungshinweise als Ausgangspunkt für das genetische Engineering verwenden.

Modifizierte Kreuzung

Ziel der modifizierten Kreuzung ist es, die Gene in den Optimierungshinweisen mit guter Bewertung möglichst in die nächste Generation zu vererben. Gene in den Optimierungshinweisen mit schlechter Bewertung soll die Kreuzung überspringen. Diese Kreuzung ähnelt dem N-Point-Crossover mit dem Unterschied, dass die N-Punkte nicht zufällig ausgewählt werden. Sie werden aus einem zufällig ausgewählten Teil der Optimierungshinweise und den Chromosomen beider Individuen ermittelt (s. Abbildung 5.11).

Die modifizierte Kreuzung ist meist für Optimierungshinweise geeignet, die einen abgeschlossenen Bereich des Fertigungsplans repräsentieren, wie z. B. die Belegung zweier gleichartiger Maschinen in der KW23. Optimierungshinweise wie z. B. die gut eingeplanten Aufträge enthalten oft Operationen, die überall im Fertigungsplan verteilt sein können. Sie können nicht einfach in das neue Individuum übertragen werden, da die Belegung einer Operation von der Reihenfolge des Gens und dadurch von allen vor ihr stehenden Operationen abhängt.

Genetische Repräsentation der Eltern mit beispielhaften Optimierungshinweisen

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix}$$

Optimierungshinweis
mit guter Bewertung
({32,13}, TST, 5)

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix}$$

Optimierungshinweis
mit schlechter Bewertung
({31,22,12}, TST, 5)

1. Umschreibung der Folgen abhängig von verwendeten Optimierungshinweisen

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix}$$

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix}$$

2. Punkte für N-Crossover aus den Optimierungsstellen ableiten

3. Den N-Crossover durchführen

$$\begin{pmatrix} 11 \\ 1 \end{pmatrix} \begin{pmatrix} 33 \\ 1 \end{pmatrix} \begin{pmatrix} 31 \\ 2 \end{pmatrix} \begin{pmatrix} 12 \\ 2 \end{pmatrix} \begin{pmatrix} 22 \\ 2 \end{pmatrix} \begin{pmatrix} 32 \\ 3 \end{pmatrix} \begin{pmatrix} 13 \\ 3 \end{pmatrix} \begin{pmatrix} 21 \\ 3 \end{pmatrix}$$
Genetische Repräsentation des Nachkommens

Abbildung 5.11: Gezielte Kreuzung im GA

Da ein Optimierungshinweis sich auch häufig auf ein Zeitintervall und gleichartige Maschinen bezieht, ist es oft vorher günstig, das Chromosom in einer äquivalenten Darstellungsform umzuschreiben (s. Abbildung 5.11). Im Chromosom sind die einzelnen Gene mit Nummer ji nach S_{ji} und anschließend nach μ_{ji} geordnet. Durch Sortierung erst nach (der Art von) μ_i und anschließend nach S_{ji} können besonders Optimierungshinweise für $\emptyset DLZ$, TST und TSC leichter im Chromosom erkannt werden. Diese Sortierungsart ist auch sehr bequem für die gezielte Mutation.

Modifizierte Mutation

Während die klassische Mutation Gene zufällig auswählt und zufällig verändert, werden in der modifizierten Mutation Gene aus beliebigen Optimierungshinweisen (gekennzeichnet durch „*“) ausgewählt und zufällig verändert (s. Abbildung 5.12a). Die modifizierte Mutation soll an erster Stelle Gene korrigieren, die in Optimierungshinweisen mit schlechter Bewertung vorkommen. Allerdings ist unter Kenntnis der Art des Optimierungshinweises auch eine gezielte Mutation der ausgewählten Gene durchführbar.

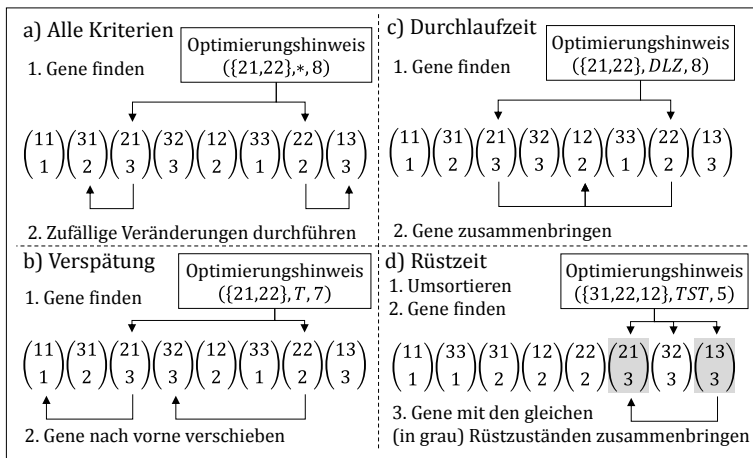


Abbildung 5.12: Gezielte Mutationsarten abhängig von der Art des Optimierungshinweises im GA

Wie Abbildung 5.12 zeigt, werden für manche Hinweisarten auch vielversprechendere Veränderungen durchgeführt. Bei einem verspäteten Auftrag können alle seine Gene jeweils um einige Positionen nach vorne verschoben werden, was zu einer Erhöhung ihrer Priorität und zu einer möglichen Einplanung zu einem früheren Zeitpunkt führt (b). Bei verfrühten Aufträgen ist auch eine gezielte Veränderung von Genen in Optimierungshinweisen mit guter Bewertung durch Verschiebung nach hinten möglich. Bei Optimierung der Durchlaufzeit können die zugehörigen Gene

so umsortiert werden, dass sie näher zusammen stehen, was zu einer kompakteren zeitlichen Einplanung der Operationen und einer Verringerung der Auftragsdurchlaufzeit führen soll (c).

Zur Optimierung der Rüstzeiten werden einfache Regeln angewendet. Zum Beispiel können zwei Gene ji und kl nebeneinander platziert werden, falls $Y_{ji} = Y_{kl} > 0$ gilt. Ggf. wird dabei auch der Maschinenindex angepasst (d).

Ein Vorteil der beiden Operatoren ist ihre Unabhängigkeit von der eigentlichen Art der Optimierungshinweise. So sind sie auch für andere Zielfunktionen einsetzbar. Falls die Hinweisart der gezielten Mutation bekannt ist, führt sie fortgeschrittene Veränderungen durch.

5.5.4 Anpassung des Genetischen Algorithmus für eine kontinuierliche Optimierung

Der GA ist ein iteratives Feinplanungsverfahren, das sich in seiner ursprünglichen Version darauf verlässt, dass sich sowohl die Problemstellung als auch die Zielsetzung nicht ändern. Die permanente Optimierung einer laufenden Fertigung erfordert die Aktualisierung der Problemstellung. Zwischen zwei solchen Aktualisierungen ist die Problemstellung veraltet und das Feinplanungsverfahren läuft für diese Zeit quasi entkoppelt (vgl. 2.4.3). Die beste gefundene Lösung ist möglicherweise nach der Aktualisierung sogar schlechter als der aktuelle Fertigungsplan. Der Ablauf des GA wird deswegen um eine Aktualisierungsphase ergänzt.

Die Aktualisierung der Problemstellung führt zur erneuten Evaluierung aller Individuen in der Population. Neue Gene werden möglicherweise hinzugefügt oder vorhandene entfernt, abhängig davon, ob neue Operationen hinzukamen oder vorhandene entfernt wurden. Bei der Neuevaluierung eines Individuums wird

außerdem dafür gesorgt, dass hinzugekommenen Operationen, wenn möglich, die gleichen Planzeiten und Ressourcen zugeteilt werden wie im aktuellen Fertigungsplan. Die komplette Evaluierung der Population ist ein aufwändiger Vorgang, dessen Rechenzeit mit der Größe der Population linear wächst.

Die Evaluierung der Individuen in einem für die Feinplanung eingesetzten GA ist die zeitintensivste Phase. In ihr wird ein Fertigungsplan konstruiert und vollständig bewertet. In dieser Phase kann die Evaluierung der Individuen aus der neuen Population auf mehreren Instanzen des GTA parallelisiert werden [AWWB09, S.17–22, Cant98]. Die Rechenzeit in den anderen Phasen ist vergleichsweise gering, dadurch vernachlässigbar und für die Parallelisierung unbedeutend. Um die Parallelisierung durchzuführen, ist in diesem Fall eine Kopie (des zur Optimierung markierten Fragments) des Fertigungsplans für jede Instanz erforderlich. Eine Kopie ist auch erforderlich, wenn die Markierung zu groß ist und eine Neuevaluierung der Population nach jedem Ereignis zu zeitaufwändig wäre. In diesem Fall ist es auch zulässig, wenn das Verfahren über ein längeres Zeitintervall entkoppelt läuft. Der zusammengefasste Ablauf ist in Abbildung 5.13 dargestellt.

Der modifizierte GA profitiert von allen Funktionen der Feinplanungsplattform wie den Bewertungskennzahlen, dem Vergleich, der Verfahrenssteuerung und den Optimierungshinweisen. Bei genauer Betrachtung trägt auch die inkrementelle Bewertung zur Gesamtperformanz bei. Zwar konstruiert eine Instanz des GTA immer einen neuen Fertigungsplan, aber die genetische Population enthält meistens verwandte Individuen, die oft gleiche Sequenzen enthalten. Die inkrementelle Bewertung wertet nur die Differenzen des neuen Fertigungsplans zum Vorgänger und spart somit auch Rechenzeit in der Evaluationsphase.

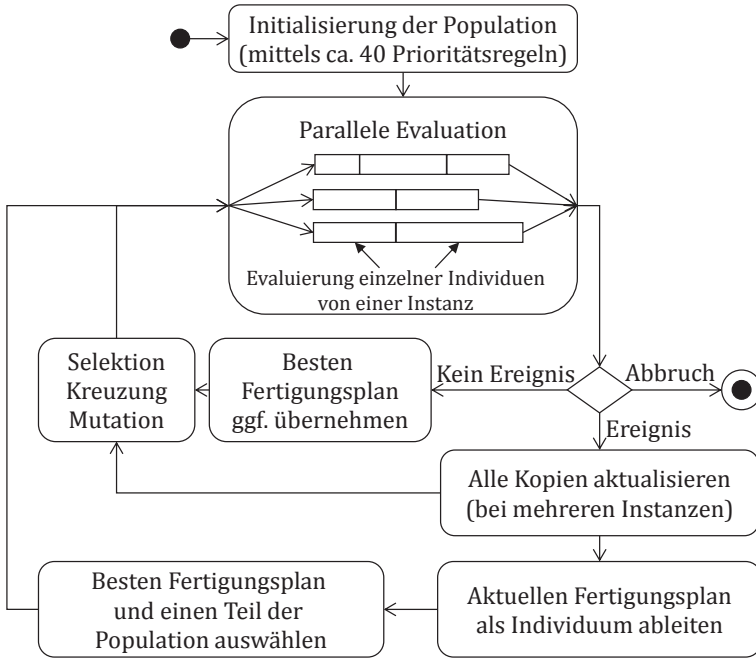


Abbildung 5.13: Zusammenspiel zwischen Feinplanungsplattform und GA

6 Anwendung und Ergebnisse

Das in dieser Arbeit entwickelte und prototypisch implementierte Konzept für die Feinplanungsplattform bringt eine Reihe von Verbesserungen und erlaubt einen ganz neuartigen Umgang mit Feinplanungsverfahren in der laufenden Fertigung. Es erlaubt die permanente Optimierung der Fertigung bei gleichzeitiger Kollaboration eines oder mehrerer Planer. Um realitätsnahe Auswertungen zu erzielen, wird eine Simulationsumgebung erstellt, die eine laufende Fertigung samt möglicher Abweichungen nachbildet. Für diese Evaluierung werden Fertigungsdaten von zwei Unternehmen verwendet.

Unterkapitel 6.1 beschreibt die hierfür erstellte dynamische Simulationsumgebung. In 6.2 finden sich implementierungsspezifische Details des realisierten Softwaresystems. Unterkapitel 6.3 erläutert die von zwei verschiedenen Unternehmen bereitgestellten Anwendungsfälle. Die Vorbereitung der Simulationsläufe und der Evaluierungsplan werden in 6.4 erläutert. Unterkapitel 6.5 und 6.6 fassen die Simulationsergebnisse zusammen. In 6.7 wird eine abschließende Bewertung der Feinplanungsplattform gegeben.

6.1 Simulationsumgebung

Feinplanungsverfahren sind für eine statische Problemstellung leicht zu evaluieren, da sie in diesem Fall abgeschlossen und ohne Kopplung zu anderen Systemen ablaufen. In der realen Welt sollen sie adäquat auf Abweichungen im Fertigungsablauf durch Störungen, Verspätungen etc. reagieren. Auch sollen die Feinplanungsalgorithmen Änderungen im Auftragsbestand und in den

Kapazitätsverfügbarkeiten schnell aufnehmen und möglichst berücksichtigen.

Um generell verlässliche Aussagen über den Einsatz und die Parametrierung neuer Feinplanungssysteme zu treffen, sind realitätsnahe Simulationsuntersuchungen notwendig. Derartige rechnergestützte Fertigungssimulationen werden zunehmend in der Industrie eingesetzt. Vor der physischen Durchführung einer Änderung in einer funktionierenden Fertigung wird im nachgebildeten genauen Modell der Fertigung (sog. Digitale Fabrik) deren Auswirkung analysiert [BrGW11, WeJR03]. Heute setzt z. B. die Halbleiter- und Automobilindustrie Methoden der digitalen Fabrik ein, um Entscheidungen über die Installation neuer Fertigungsanlagen zu treffen, die Layoutplanung neuer Fertigungshallen durchzuführen oder die Funktionsweise von speicherprogrammierbaren Steuerungen (SPS) zu testen. Für die Fertigungssteuerung werden Simulationen hauptsächlich für die Auswahl der besten Steuerungsstrategie oder zur Ermittlung des versteckten Potentials eingesetzt [BaDi11b, KáPM04, PKCM05, SFPF10].

Die am Markt verfügbaren Simulationswerkzeuge haben ihre Schwerpunkte in Hallenlayoutplanung, Materialfluss, Roboterkinematik und Arbeitsergonomie [Kühn06, S.203–280]. Sie konzentrieren sich meist auf Linienfertigung und bilden die Ereignisse in einer dynamischen Werkstattfertigung sehr unzureichend nach. Deshalb fällt die Entscheidung auf eine performante maßgeschneiderte Eigenentwicklung [DiBa09], die nur die für die Validierung der Feinplanungsplattform relevanten Aspekte der IT-Systemlandschaft nachbildet.

Kommunikationspartner der Feinplanungsplattform sind das ERP System und die einzelnen Maschinensteuerungen (Abbildung 3.6). Eine Simulationsumgebung für die Feinplanungsplattform entsteht durch Nachbildung der relevanten Teile der beiden Arten von Kommunikationspartnern (s. Abbildung 6.1). Die nachgebil-

deten Komponenten werden folglich ERP-Substitut und Fertigungssubstitut genannt.

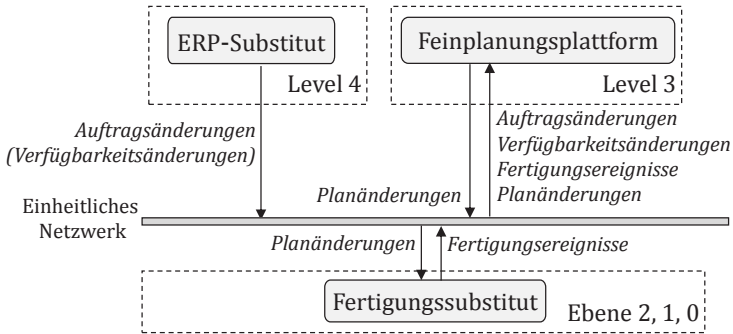


Abbildung 6.1: Simulationsumgebung

Das ERP-Substitut simuliert die Auftragsverwaltung des ERP Systems. Es erzeugt Meldungen über das Einfügen, Ändern oder Stornieren von Aufträgen, die Teil der Bewegungsdaten sind. Stammdaten wie Maschinenpark oder Personal ändern sich deutlich seltener und sind außerhalb des Tagesgeschäfts. Aus diesem Grund sind sie zwar für Demonstrationszwecke im ERP-Substitut nachbildbar, in den einzelnen Simulationsläufen werden sie jedoch nicht weiter betrachtet. Das ERP-Substitut soll wiederholbare Ereignismeldungen an die Feinplanungsplattform schicken, weshalb es in zwei unabhängige Komponenten unterteilt ist: Ereignisplaner und Meldungsgenerator. Vor den Simulationsläufen erstellt der Ereignisplaner einen initialen Auftragsbestand und eine Ereignisliste für das simulierte Zeitintervall, die der Meldungsgenerator sukzessiv beim Erreichen des simulierten Zeitpunkts in einzelne Meldungen umwandelt und an das Feinplanungssystem sendet.

Der Ereignisplaner pflegt eine Liste von Auftragschablonen. Jede Auftragschablone enthält:

- Gesamtgröße für die Schablone
- durchschnittliche Auftragsgröße
- Zeitspanne zwischen r_j und d_j des zu generierenden Auftrags j (Spielraum des Auftrags)
- Zeitspanne zwischen Erstellungszeitpunkt und r_j des zu generierenden Auftrags j (bei Eilaufträgen ist sie gleich 0)
- Anteil stornierter Aufträge
- Anteil geänderter Aufträge
- Ausgangsmaterial
- Meldungsschablone

Pro Schablone generiert das ERP-Substitut solange Aufträge, bis die Gesamtgröße aller Aufträge die voreingestellte Gesamtgröße der Schablone erreicht. Es erzeugt die Ereignisse für die generierten Aufträge sowie für eine mögliche spätere Stornierung und Änderung dieser Aufträge.

Zum Freigabezeitpunkt eines Auftrags fügt der Meldungsgenerator die Auftragsdaten in die Meldungsschablone ein und sendet diese an das Feinplanungssystem. Zum Änderungs- oder Stornierungszeitpunkt werden ebenfalls entsprechende Änderungen erstellt. Anders als in einer realen Fertigung sollen die Änderungs- und Stornomeldungen vor dem Freigabezeitpunkt auftreten. Das verhindert unerwünschte Nebeneffekte bei der Vergleichbarkeit, etwa die Benachteiligung von Simulationsläufen, bei denen ein stornierter Auftrag bereits gestartet wurde.

Das ERP-Substitut stellt unidirektionale Verbindung mit der Feinplanungsplattform her. In der Realität werden bei der Annahme eines Eilauftrags Kapazität und Auslastung berücksichtigt. In diese Entscheidung fließen weitere Faktoren, wie die Möglichkeit für zusätzliche Schichten, pünktliche Lieferung der notwendigen Eingangsmaterialien etc. ein. Die Simulationsläufe sollen jedoch vergleichbar und wiederholbar ablaufen. Eine schwankende Auftragsmenge würde die Vergleichbarkeit deutlich erschwe-

ren, weshalb der aktuelle Fertigungsplan oder der Fertigungsfortschritt in der Simulation sich nicht auf die vom ERP-Substitut generierten Meldungen auswirkt.

Das Fertigungssubstitut soll den Fertigungsplan von der Feinplanungsplattform erhalten, seinen Ablauf simulieren und bei Bedarf Abweichungen vorspiegeln. Anders als beim ERP-Substitut werden die Abweichungen nicht vor der Simulation vorbereitet. Es kann z. B. keine Störung auf eine Maschine eingeplant werden, ohne zu wissen, ob die Maschine zu diesem Zeitpunkt überhaupt eine Operation durchführt.

Das Fertigungssubstitut simuliert die Maschinen. Es erlaubt die Konfiguration des Verhaltens pro μ_i über die folgenden Variablen:

- Durchschnittliche Abweichung der Ist- p_{ji} von der Soll- p_{ji}
- Anteil der Störung an der Gesamtbearbeitungsdauer pro μ_i
- Durchschnittliche Störungsdauer pro μ_i
- Zeitintervall zwischen zwei Fortschrittmeldungen

Das Fertigungssubstitut verwaltet je Maschine für alle simulierten Störungen ein Konto. Die Wahrscheinlichkeit für eine Störung steigt, wenn die Gesamtdauer der Störungen kleiner als der eingestellte Anteil ist, und fällt, wenn für die Maschine zu viele Störungen simuliert wurden.

Für alle Abweichungen zur Soll- p_{ji} wird ebenfalls ein Konto gepflegt, wobei anders als beim Störungskonto die Abweichung auch negativ sein kann. Am Ende der Simulation sollte dieses Konto einen Wert nahe Null haben. Dies spiegelt gut gepflegte Stammdaten der Problemstellung wider, was für die durchgeführten Simulationsläufe auch angenommen wird. Die Ist- p_{ji} wird, wie die meisten simulierten Werte, mittels eines normalverteilten Zufallszahlgenerators erzeugt. Dabei werden die Parameter der

Normalverteilung abhängig vom Kontostand abgeleitet, um den Kontowert möglichst um die Null zu halten.

Das Fertigungssubstitut setzt eine ereignisdiskrete Simulation ein. Abbildung 6.2 zeigt die Simulation zweier Operationen des Fertigungsplans aus Abbildung 2.11. Aus deren Belegungen erzeugt das Fertigungssubstitut Ereignisse, welche letztlich zu Meldungen an die Feinplanungsplattform führen. Als Beispiel wird auf der Abbildung nach der ersten Fortschrittmeldung ($Progress_{12}^1$) eine Störung simuliert.

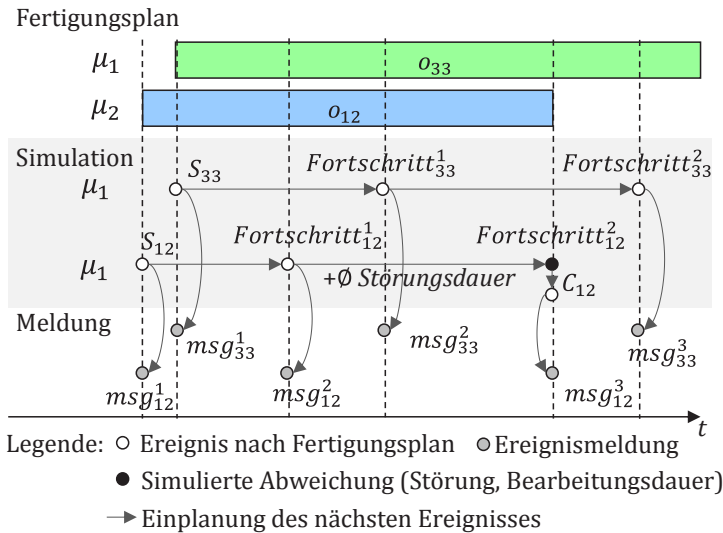


Abbildung 6.2: Simulationsablauf im Fertigungssubstitut

Das ERP-Substitut bildet zusammen mit dem Fertigungssubstitut eine dynamische Simulationsumgebung, die nur die notwendigen Teile von ERP und Fertigung nachbildet. Simulationen können damit gegenüber der realen Fertigungsgeschwindigkeit beschleunigt werden. Damit ist ein mehrmonatiger Fertigungsplan innerhalb eines Tages simulierbar.

6.2 Implementierung der Feinplanungsplattform

Die Feinplanungsplattform wurde auf große Problemstellungen, Plattformunabhängigkeit und Portierbarkeit auf Cloud-Plattformen wie Virtual Fort Knox¹ sowie kommerzielle Nutzung² ausgelegt. Typische Bedienfunktionen in der Feinplanungsplattform sollten eine maximale Reaktionszeit von einer Sekunde für Problemstellungen bestehend aus 100.000 Dispositionsobjekten (Aufträge + Operationen) auf handelsüblichen Rechnern erreichen.

Die Architektur folgt dem typischen Client-Server Konzept, in dem der Server die Verwaltung, Bewertung, Optimierung und Persistenz des aktuellen Fertigungsplans samt der Problemstellung übernimmt. Abbildung 6.3 zeigt die Referenzarchitektur und die für die Realisierung ausgewählten Fremdkomponenten. Der Client stellt den aktuellen (oder ausgewählten) Fertigungsplan mittels interaktiver Gantt-Diagramme, Intervall-Gebirge usw. grafisch dar. Diese Architekturart ist heute zwar selbstverständlich für viele neue Anwendungen mit Mehrbenutzerzugriff, zurzeit ist die Mehrheit der am Markt verfügbaren Feinplanungssysteme jedoch als Einbenutzer-Fat-Client realisiert [Bože12, Thie11, S.476–648].

Für den Serverteil wurde die Sprache Java aufgrund der Plattformunabhängigkeit, der Verbreitung und der hohen Verfügbarkeit von freien Software-Bibliotheken ausgewählt. Der Serverteil ist sowohl eigenständig als auch innerhalb des Apache Tomcat³ Servlet-Containers ausführbar. Auch andere verbreitete Servlet-Container wie SAP Netweaver und Jetty werden unterstützt.

¹ <https://www.virtualfortknox.de>, App: "JMS - Job Management System"

² <http://www.flexis.com/de/kunden/kawasaki-machine-industry-co.html>

³ <https://tomcat.apache.org>

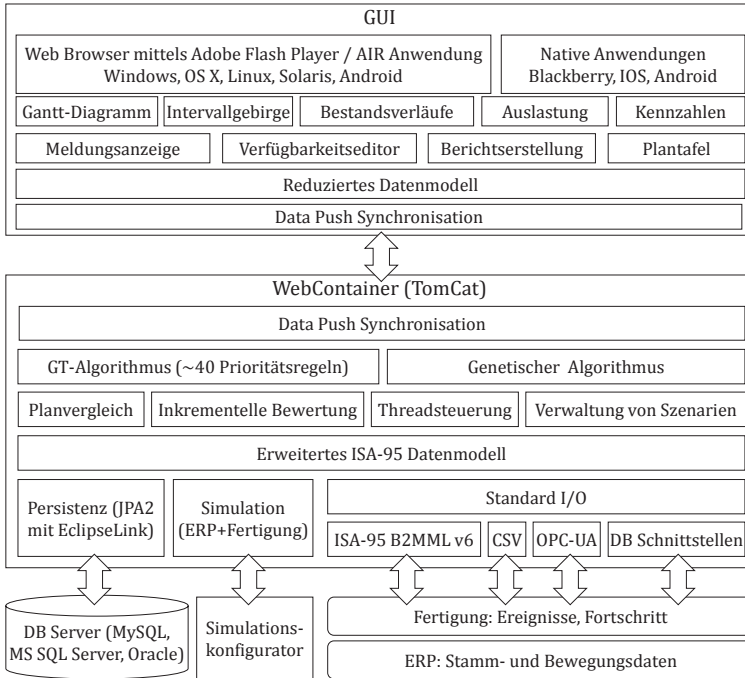


Abbildung 6.3: Architektur der realisierten Feinplanungsplattform

Der Serverteil integriert die Bewertungs- und Vergleichskomponenten und steuert die Verfahrensinstanzen. Dazu stellt er einen Threadpool zur Verfügung und weist den unterschiedlichen Verfahrensinstanzen und Komponenten eine bestimmte Anzahl von Threads zu. Die einzelnen Feinplanungsverfahren binden sich an den Server über eine festdefinierte Schnittstelle an.

Alle implementierten Kern-, Hilfsfunktionen und Feinplanungsverfahren im Server bauen auf einem erweiterten ISA-95 Datenmodell [Ansi13a, Ansi13b] der Feinplanungsplattform auf. Der ISA-95 Standard ist für unterschiedliche Fertigungsarten konzipiert und definiert alle möglichen Objekte in der Feinplanung wie Ausrüstung (Maschinen, Werkzeuge, Transporteinheiten), Personal, Materialien, Aufträge und Operationen sowie ihre Beziehun-

gen zueinander. Jedes Objekt lässt sich um eine uneingeschränkte Anzahl an Eigenschaften erweitern. Auch sind die Operationen und die Ausrüstung hierarchisch modellierbar, was eine Planung auf mehreren Detaillierungsstufen erlaubt.

Der ISA-95 Standard beschreibt ein generisches Modell mit einem breiten Interpretationsraum. Den kommunizierenden Systemen wird frei überlassen, zusätzliche Semantik in die frei definierbaren Eigenschaften einzubringen. Für die Planung relevante Informationen, wie die Berechnungsregel für die Dauer einer Operation oder für die Ermittlung der reihenfolgenabhängigen Rüstzeit, mussten dem Standard hinzugefügt werden. Auch wurde eine Modellierung von alternativen Maschinen integriert.

Jede Instanz des Datenmodells repräsentiert eine Problemstellung und den dazugehörigen Fertigungsplan und wird Szenario genannt. Jedes Szenario enthält zusätzlich die notwendigen Hilfsstrukturen für die Bewertung und Optimierung. Es ist außerdem unabhängig von den anderen und kann nur ein Fragment der Problemstellung repräsentieren. Das Szenario enthält weiterhin ein für die Darstellung auf dem Client optimiertes und komprimiertes Datenmodell des Fertigungsplans, das von der Data Push Synchronisation mit dem Datenmodell des Clients abgeglichen wird.

Der aktuell auszuführende Fertigungsplan ist im Hauptszenario und wird durch das Persistierungsframework EclipseLink auf einem Datenbank Server zurzeit MySQL⁴ abgelegt. Der Server kommuniziert mit den Partnersystemen (ERP oder anderen MES sowie Maschinensteuerungen) mittels des B2MML v6 Standards⁵. B2MML ist die einzige offizielle Implementierung des ISA-95 Standards. Wie jedes XML-basierte Format, beansprucht es viel Rechenzeit für die Verarbeitung größerer XML-Dateien. Deshalb

⁴ <https://www.mysql.com>

⁵ <http://www.mesa.org/en/B2MML.asp>

wurde eine Schnittstelle auf CSV-Basis nach RFC 4180⁶ entwickelt, die alle Informationen aus B2MML auch übertragen kann. Die CSV-Schnittstelle importiert laut Performanztest die gleiche Problemstellung über 80x schneller als die B2MML Schnittstelle.

Für den Clientteil hat sich das im Web-Browser laufende Plug-In Flash Player 11.1 von Adobe als sehr performant in der Zeichnung einer großen Anzahl von geometrischen Objekten erwiesen. Der Flash Player zeigte sich in einem Performanz-Test gleich schnell oder sogar schneller als eine native Implementierung. Das zugehörige Entwicklungswerkzeug Adobe Flex 4.6⁷ bietet zudem auch die Kompilierung von nativen Anwendungen für gängige mobile Plattformen wie Google Android, Apple iOS oder RIM Blackberry OS. Mit über 98% Verbreitung auf Desktop-Rechnern und Portierungsmöglichkeit auf über 95% der Smartphones stellt die Entwicklungsumgebung keine großen Einschränkungen für die Ausführung dar. Adobe Flex nutzt die Skriptsprache Actionscript 3, die als eine ECMA Draft 4 Skriptsprache mit JavaScript verwandt ist [WeYa08]. Dies vereinfacht eine Portierung auf das im Oktober 2014 verabschiedete HTML5, sobald eine ausreichende Unterstützung auf HTML5 in gängigen Web-Browsern erreicht wird.⁸ Aus den obigen Gründen fiel die Entscheidung auf Adobe Flex als Grundlage für die GUI.

Die GUI enthält moderne grafische Oberflächenkomponenten wie Gantt-Diagramme, Intervall-Gebirge oder Materialbestandsverläufe,⁹ die alle auf dieselbe Datenverwaltung zugreifen. Die Datenverwaltung synchronisiert sich mit dem Server und dadurch mit den anderen Client-Instanzen mittels Data-Push.¹⁰ Veränderungen in der lokalen Datenverwaltung lösen Ereignisse aus, die die

⁶ <http://tools.ietf.org/html/rfc4180>

⁷ <http://www.adobe.com/products/flash-builder-family.html>

⁸ <https://html5test.com>

⁹ <http://www.provis-aps.de>

¹⁰ <http://sourceforge.net/adobe/blazeds>

Auffrischung der graphischen Anzeigekomponenten der Oberfläche bewirken. Durch das Umplanen einer Operation auf der Oberfläche oder andere Benutzerinteraktionen wird die Datenverwaltung aktualisiert, was zu einem Ereignis führt. Infolge dieses Ereignisses aktualisieren sich alle Anzeigekomponenten, die die betroffene Operation anzeigen, und die Data-Push-Synchronisierung sendet die Veränderungen an alle zurzeit angeschlossenen Clients und den Server. Diese Realisierung erlaubt vielen Benutzern eine nahtlose Zusammenarbeit auf demselben Fertigungsplan.

Durch diese Mechanismen unterstützt die Feinplanungsplattform in ihrem Kern die Mixed Initiative Planung mit mehreren Benutzern und Algorithmen. Auch andere Funktionen der Feinplanungsplattform, wie die Bewertung, sind auf eine performante Ausführung ausgelegt, so dass dem Benutzer alle eigenen und fremden Interaktionen samt ihrer Auswirkung auf die Planbewertung zeitnah dargestellt werden.

6.3 Beschreibung der untersuchten Anwendungsfälle

Die realisierte Feinplanungsplattform wird in einer realitätsnahen Simulationsumgebung untersucht. Dafür stehen Fertigungsdaten von zwei europäischen Unternehmen, Batteriehersteller und Wafer-Fertiger, zur Verfügung. Sie werden nachfolgend als BAT bzw. WAF abgekürzt. Fertigungsdaten enthalten sensible Informationen über interne Fertigungsprozesse und -probleme, weshalb sie hier stark anonymisiert wurden. Die Fertigungsdaten umfassen jeweils einen abgelaufenen repräsentativen Zeitraum und enthalten alle notwendigen Informationen für die Durchführung der Simulationsläufe. Die Simulation mit echten Unternehmensdaten kann die Frage beantworten, wie sich der Fertigungsablauf

durch eine permanente Optimierung mittels der hier erarbeiteten Ansätze verbessern würde.

Tabelle 6.1 liefert alle Eckdaten zu den simulierten Anwendungsfällen. Alle Zeiteinheiten werden nachfolgend in Tagen angegeben.

Tabelle 6.1: Daten der simulierten Anwendungsfälle

<i>Anwendungsfälle</i>	<i>BAT</i>	<i>WAF</i>
<i>Zeithorizont d_h [Tage]</i>	<i>91 (13KW)</i>	<i>98 (14KW)</i>
<i>Anzahl Maschinen m</i>	<i>21</i>	<i>71</i>
<i>Anzahl Aufträge n</i>	<i>88</i>	<i>2.223</i>
<i>Anzahl Operationen O</i>	<i>12.291</i>	<i>13.184</i>
<i>Operationen pro Auftrag O_j</i>	<i>123-152</i>	<i>4-12</i>
<i>Zusätzliche Ressourcen \mathcal{R}</i>	<i>6 MA</i>	<i>-</i>
<i>Ist-Ablauf [Tage]</i>	$\sum T_j = 0$ $\emptyset DLZ \approx 18,39$	$\sum T_j \approx 1192,4$ $TST \approx 30,7$
<i>Rahmenbedingung</i>	$\sum T_j = 0$	<i>Nicht schlechter als der Ist-Ablauf</i>
<i>Optimierungsziele</i>	$\emptyset DLZ$	$\sum T_j$ TST
<i>Spezifika</i>	<i>-</i>	<i>Rüstzeiten</i>
<i>Untere Schranken [Tage]</i>	$LB_{\sum T_j} = 0$ $LB_{\emptyset DLZ} > 5$	$LB_{\sum T_j} = 0$ $LB_{TST} > 7$

Der erste Anwendungsfall stammt aus einer Fabrik für Autobatterien (BAT, s. Tabelle 6.1). Das Unternehmen strebt eine Minimie-

rung der Durchlaufzeit an, ohne dabei in Lieferverzug zu geraten. Die Herstellung einer Autobatterie ist ein aufwändiger Fertigungsprozess, je nach Produktart bestehend aus 25 bis 40 Fertigungsstufen [AxBK08]. Einige Maschinen werden in unterschiedlichen Stufen innerhalb eines Auftrags verwendet. Auf Wunsch des Kunden wurde jede Charge sowie die Maschinenvorbereitung (wenn vorhanden) innerhalb einer Stufe als einzelne Operation modelliert. Daher variiert die Anzahl der Operationen pro Auftrag in der Problemstellung zwischen 123-152. Aufgrund von Personalengpässen bei einer Fertigungsstufe wurden die dort zugeordneten 6 Mitarbeiter separat modelliert.

Der zweite Anwendungsfall stammt aus der Chip-Fertigung einer Halbleiterfabrik (WAF, s. Tabelle 6.1). Die hergestellten Mikrochips werden in eingebetteten Systemen im Auto und der Unterhaltungselektronik eingebaut. Aus diesem Grund ist die Produktvielfalt sehr hoch und gleichzeitig die Anfrage sehr schwankend. Die Herstellung der Chips enthält je nach Komplexität 4 bis 12 Fertigungsstufen. Für die Bonding- und Klebeprozesse existieren 30 bzw. 22 alternative Maschinen mit unterschiedlicher Bearbeitungsdauer. Dazu sind die Fertigungsparameter für die einzelnen Produktarten sehr unterschiedlich und für die notwendigen Umrüstungen fallen oft Rüstzeiten von bis zu drei Stunden an, die nicht selten die eigentliche Bearbeitungsdauer der Operation übersteigen. Das Unternehmen strebt Termintreue und eine Reduzierung der Gesamtrüstzeit an. Verspätungen werden in Kauf genommen, wenn die Rüstzeitersparnis dies rechtfertigt. Denn eine Umrüstung verursacht einerseits Kosten und andererseits steigt die Ausschussquote in der anschließenden Anlaufphase deutlich. Die große Anzahl von Maschinen mit reihenfolgenabhängigen Rüstzeiten zusammen mit dem Problemumfang stellt für jedes Planungsverfahren eine Herausforderung dar. Die Chip-Fertigung ist generell ein sehr anspruchsvolles Planungsproblem [MFDM11].

Die unteren Schranken für die beiden Anwendungsfälle wurden mit einem einfachen Lauf des GTA gegen unbeschränkte Kapazitäten ermittelt. Für LB_{TST} wurden die in 5.5.2 vorgestellten Regeln verwendet. Die Daten für den Ist-Ablauf basieren auf Angaben der Unternehmen über S_{ji} , C_{ji} und μ_{ji} , die zusätzlich plausibilisiert wurden.

Die beiden Anwendungsfälle stellen sehr heterogene Feinplanungsprobleme dar (s. auch Tabelle 6.1). Die Problemgrößen sind um das 15- bis 30-fache größer als die größten Benchmarkprobleme (s. 2.3.7) und stellen für jedes Feinplanungsverfahren eine Herausforderung dar.

6.4 Vorbereitung der Simulationsläufe

Vor der Durchführung der Simulationsläufe werden in diesem Unterkapitel wichtige Details über die Gestaltung der Simulationsläufe, angefangen bei der Simulationsumgebung, beschrieben. Danach wird ein Evaluierungsplan für die Anwendungsfälle erarbeitet. Abschließend wird anhand der bereitgestellten Fertigungsdaten gezeigt, wie dynamische Problemstellungen zu konstruieren sind.

6.4.1 Simulationsumgebung und Kalibrierung

Die Gestaltung der Simulationsumgebung fand auf einem Server mit Sechskernprozessor i7-970, 24 GB Arbeitsspeicher und SSD RAID statt. Dieser Simulationsrechner ist mit dem Betriebssystem Windows 2012 Server, Java Runtime Environment 1.7, Tomcat 7 und MySQL 5.6 ausgestattet. Die Rechnerarchitektur der Simulationsumgebung erlaubt die Ausführung von bis zu sechs komplett unabhängigen parallelen Prozessen ohne Laufzeitbeeinträchtigung.

Um die Simulationsläufe zu parametrieren, wurden vorerst Evaluierungen mit unterschiedlichen Routinen der Feinplanungsplattform durchgeführt. Die ermittelten Werte sind in Tabelle 6.2 in Millisekunden angegeben und sollen bei der Parametrierung der Evaluierung helfen. Der implementierte GA wurde ohne modifizierte Operatoren (GA0) mit der folgenden Parametrierung ausgeführt: Populationsgröße = 60, Kreuzungswahrscheinlichkeit = 0,4, N-Point-Crossover = 3, Mutationswahrscheinlichkeit = 0,4, Mutationsanteil = 0,005, Maschinenwechselwahrscheinlichkeit = 0,2, SteadyState = 0,18. Diese Einstellungen wurden auf die beiden Anwendungsfälle ohne Anwendung von modifizierten Operatoren iterativ abgeleitet. Des Weiteren wurde der GA mit unterschiedlichen Anteilen an modifizierten Operatoren (GA1-3) konfiguriert (s. Tabelle 6.3). GA1-3 nutzen zur besseren Vergleichbarkeit mit GA0 die Parametrierung von GA0.

Tabelle 6.2: Rechenzeit für typische Vorgänge in der Feinplanungsplattform auf den Anwendungsfällen in Millisekunden

<i>Vorgang</i>	<i>BAT</i>	<i>WAF</i>
<i>Bewertung</i>	<i>10</i>	<i>20</i>
<i>Bewertung bei 10% Änderung</i>	<i>3</i>	<i>4</i>
<i>Ein Lauf des GTA</i>	<i>960-3100</i>	<i>460-1500</i>
<i>Evaluierung eines Individuums im GA</i>	<i>790-1820</i>	<i>210-1502</i>
<i>∅Evaluierung eines Individuums im GA0</i>	<i>1155</i>	<i>235</i>
<i>∅Evaluierung eines Individuums im GA1-3</i>	<i>1011</i>	<i>219</i>

Tabelle 6.3: Unterschiede zwischen GA0, GA1, GA2 und GA3

<i>Parameter</i>	<i>GA0</i>	<i>GA1</i>	<i>GA2</i>	<i>GA3</i>
<i>Anteil an modifizierten Kreuzungen</i>	<i>0</i>	<i>0,25</i>	<i>0,5</i>	<i>0,75</i>
<i>Anteil an modifizierten Mutationen</i>	<i>0</i>	<i>0,25</i>	<i>0,5</i>	<i>0,75</i>

Um die statistische Signifikanz der Ergebnisse für den GA zu ermitteln, wurden zunächst 10 Läufe mit den Zielfunktionen $(\emptyset DLZ + 1/n \sum T_j)/2$ und $(TST + \sum T_j)/2$ für BAT bzw. WAF durchgeführt. Eine Ermittlung der σ^2 aus 5 Läufen zeigt die Streuung nach einer bestimmten Anzahl an evaluierten Individuen im GA0. Bei BAT gilt $\emptyset = 6,47$ und $\sigma^2 = 0,15$ sowohl nach 40.000 als auch nach 25.000 Evaluierungen (s. Abbildung 6.4). Bei WAF zeigt sich der GA0 mit $\emptyset = 162$ und $\sigma^2 = 40$ bereits nach 10.000 Evaluierungen trotz gleicher Problemgröße stabil (s. Abbildung 6.5). Grund dafür ist die deutlich geringere Anzahl von Beziehungen zwischen den Dispositionsobjekten. Bei der Betrachtung weiterer Läufe ändern sich die Durchschnitts- und Varianzwerte unwesentlich (<1%). Aus diesem Grund und im Hinblick auf zeitintensive dynamische Simulationsläufe werden die nachfolgenden Simulationsuntersuchungen mit je 5 Läufen durchgeführt und deren Mittelwert präsentiert.

GA1-3 zeigten in den Kalibrierungsläufen einerseits niedrigere Werte von σ^2 , andererseits schlugen sie i. A. bessere Individuen vor. Letzteres hat jedoch aufgrund der Natur des eingesetzten GTA eine signifikante Auswirkung auf die durchschnittliche Evaluierungsdauer (s. Tabelle 6.2). Der GTA frischt nach jeder Einlastung alle Warteschlangen auf (vgl. 2.3.3). Fertigungspläne aus besseren Individuen sind kompakter und die Warteschlangen innerhalb des GTA sind aufgrund der unterschiedlichen frühesten Startzeitpunkte (r_j) deutlich kürzer. Dies erklärt die ca. 10% kürzere

durchschnittliche Evaluierungszeit bei GA1-3. Aufgrund der etablierten Annahme in der Forschung, die Evaluierungszeit im GA als Konstante zu betrachten, wird der nachfolgende Vergleich zwischen GA0 und GA1-3 diesen Effekt in statischen Simulationsläufen ignorieren.

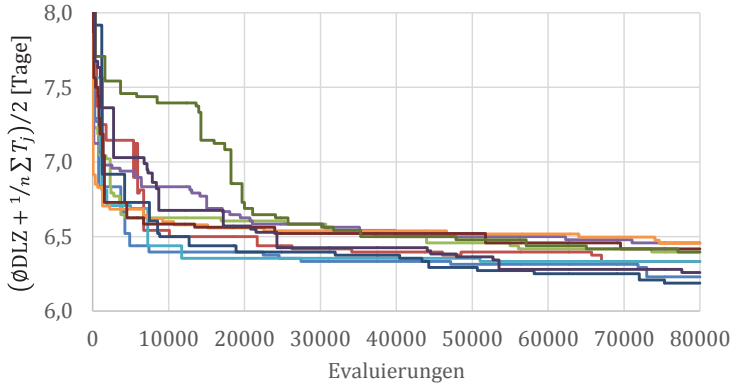


Abbildung 6.4: Zehn Simulationsläufe von GA0 für BAT

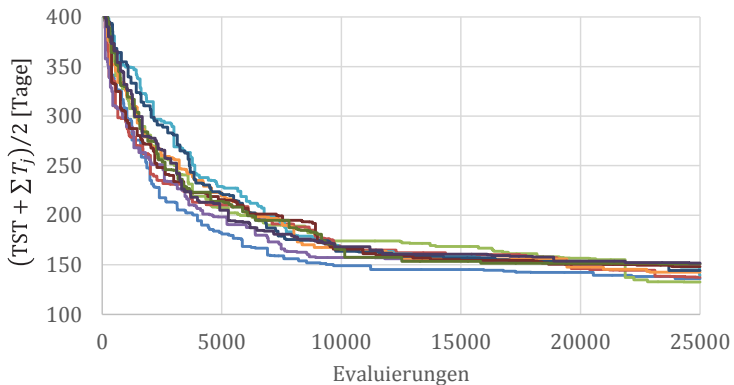


Abbildung 6.5: Zehn Simulationsläufe von GA0 für WAF

6.4.2 Evaluierungsplan

Das Ziel der nachfolgenden Simulationsläufe ist einerseits die in dieser Arbeit entwickelten Ansätze mit realen Daten zu evaluieren und andererseits die prototypisch implementierte Feinplanungsplattform für den Einsatz in einer laufenden Fertigung zu konfigurieren. Dies soll in zwei Schritten erfolgen.

Im ersten Schritt werden die Ansätze evaluiert, die auf Leistungssteigerung orientiert sind (vgl. Abbildung 6.6). Die Leistung wird hier sowohl hinsichtlich der Lösungsqualität als auch der Laufzeit betrachtet. Diesbezüglich werden statische und dynamische Problemstellungen aus den Fertigungsdaten konstruiert. Die Ansätze werden bis auf die Relevanzfunktionen in einer statischen Simulation (1a) kalibriert und evaluiert. Zu Beginn werden die Ergebnisse von GA0-3 gegenübergestellt.

Anschließend wird der GA mit der besten Konfiguration parallelisiert. Es werden unterschiedliche Größen für den Markierungsbereich durchprobiert. Die maximale Größe bei 6 parallel laufenden Verfahrensinstanzen beläuft sich auf ~12% des Fertigungsplans. Somit hat eine Verfahrensinstanz nach Freigabe ihrer Markierung einen ausreichenden Anteil von ~33% des Fertigungsplans für neue Markierungen. Auf der Suche nach einer optimalen Markierungsgröße wird diese Markierungsgröße abhängig vom erreichten Ergebnis reduziert oder erhöht. Als Abbruchkriterium für die innere Optimierungsschleife einer Verfahrensinstanz wird eine maximale Zeit von 50 Sekunden oder eine fehlende signifikante Verbesserung (<2%) innerhalb der letzten 20 Sekunden angesetzt. Da die parallele Ausführung von Verfahrensinstanzen im Hinblick auf große Planungsprobleme entworfen wurde, wird geprüft, ob sie in der nachfolgenden dynamischen Simulation auch geeignet ist.

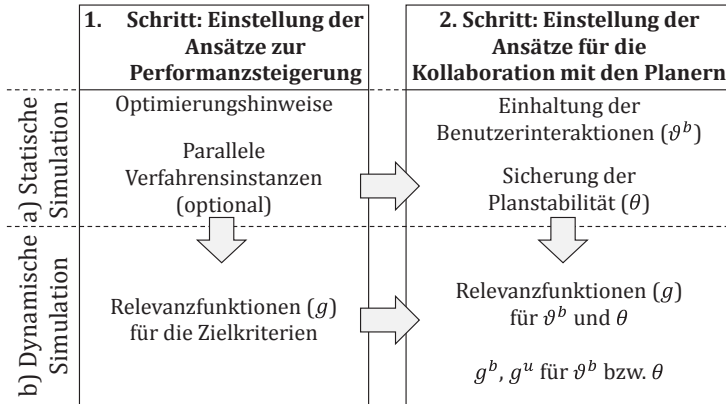


Abbildung 6.6: Evaluierungsplan für die einzelnen Ansätze

In einer dynamischen Simulation (1b) werden anschließend die zugehörigen Relevanzfunktionen angepasst. Zur Vereinfachung werden alle untersuchten Relevanzfunktionen als stückweise lineare Funktion aus einfachen Mengen $\{(t_i, w_i)\}$ abgeleitet. Als Referenz dient hier das Simulationsergebnis mit der einfachen Version des jeweiligen Bewertungskriteriums als Zielfunktion, die äquivalent zur zeitbehafteten Version unter $g_o \equiv 1$ ist.

Auch hier wird iterativ vorgegangen und es wird vorerst eine Variante ausprobiert. Abhängig vom erreichten Ergebnis werden bei Bedarf weitere Relevanzfunktionen evaluiert.

Die Verwendung von zeitbehafteten Bewertungskriterien kann theoretisch zur Verschiebung von Problemen in die Zukunft führen. Aus diesem Grund werden zwei weitere Läufe durchgeführt, die in einer ersten Stufe einen Vergleich von Fertigungsplänen über g_o durchführen und nur falls der Unterschied der Zielfunktionswerte kleiner als ein vorzugebender Schwellenwert ist, werden die zeitbehafteten Versionen der Bewertungskriterien unter g_x in der zweiten Stufe berücksichtigt. Sie werden als $g_o|g_x$ annotiert. Der Schwellenwert wird auf 3% festgelegt, damit der Unterschied aus Sicht eines Planers vernachlässigbar ist.

In Schritt 2 wird der Fokus auf die entwickelten Kollaborationsansätze gelegt. Die Feinplanungsplattform mit den Einstellungen aus 1a wird in einer statischen Simulation (2a) auf die Kollaboration mit dem Planer kalibriert. Die hier entworfenen Ansätze für die Sicherung der Planstabilität und die Einhaltung der Benutzerinteraktionen werden nach den Wünschen der Planer des jeweiligen Unternehmens eingestellt.

In den statischen Simulationsläufen werden die akzeptablen und zulässigen GW für θ und ϑ^b festgelegt. Im ersten Simulationslauf wird allein die Auswirkung der Verwendung von θ auf das Endergebnis gemessen. Der erreichte Zielfunktionswert wird mit dem klassischen Ansatz für Übernahmeentscheidungen basierend auf ϑ aus dem Stand der Technik verglichen, der sich nur auf die aktuelle Übernahme bezieht (vgl. 2.4.4). Aufgrund der festen Einschränkung von θ kann es passieren, dass nach einer gewissen Zeit keine neue Übernahmen zugelassen werden. Deshalb wird θ in einem weiteren Simulationslauf nur auf die letzten x Evaluierungen beschränkt ($\theta|x$).

In weiteren statischen Simulationsläufen wird ϑ^b eingeschaltet. Als Vergleichsbasis zum hier eingeführten Kriterium ϑ^b wird die starre Fixierung von Operationen verwendet. Dieser Ansatz aus dem Stand der Technik modelliert die vom Planer gesetzten Startzeiten der Operationen als harte Restriktionen in der Problemstellung. Dadurch wird der Lösungsraum zwar stark eingeschränkt, das Feinplanungsverfahren berücksichtigt jedoch (so weit zulässig) die Wünsche des Planers. Für die Simulationsläufe werden jeweils 2% bzw. 5% der Operationen als Ergebnis von Benutzerinteraktionen (BI) gekennzeichnet. Die gekennzeichneten Operationen sollen möglichst wenigen Aufträgen angehören, um noch Spielraum für die Optimierung zu bieten. Als gewünschte S_{ji}^k für die simulierten Benutzerinteraktionen b_k werden die Ist- S_{ji} verwendet. Bei der Kennzeichnung von Operationen als Ergebnis von Benutzerinteraktionen wird auch beachtet, dass der Anteil

an gekennzeichneten Operationen während der Dauer einer dynamischen Simulation im nachfolgenden Teilschritt nur minimal schwankt.

In Teilschritt 2b werden die Relevanzfunktionen für ϑ^b und θ definiert und evaluiert. Als Grundlage dafür sollen die Zeitpunkte der besten gefundenen Relevanzkurven aus Teilschritt 1b dienen. Auf Grundlage der Angaben des Unternehmens und der Wünsche des Planers werden jeweils zwei Relevanzkurven pro Bewertungskriterium definiert, die paarweise identische Zeitpunkte besitzen, sich jedoch von den Gewichtungen unterscheiden. Wie in Teilschritt 2a wird in dynamischen Simulationsläufen vorerst die Auswirkung von θ auf das Endergebnis evaluiert. Anschließend wird ϑ^b bei 2% bzw. 5% der gekennzeichneten Operationen eingeschaltet und das erreichte Endergebnis mit dem Einsatz von starren Fixierungen verglichen. Nach diesen Simulationsläufen ist die Feinplanungsplattform für den jeweiligen Anwendungsfall vollständig konfiguriert.

6.4.3 Konstruktion der dynamischen Problemstellungen

Die von den Unternehmen bereitgestellten Fertigungsdaten erfüllen alle Voraussetzungen für die Konstruktion von statischen Problemstellungen. Für realitätsnahe dynamische Simulationsuntersuchungen (Teilschritte 1b und 2b) sind weitere Angaben von den Unternehmen notwendig, um daraus die Konfiguration der Simulationsumgebung abzuleiten (vgl. 2.4 und 6.1). Die Angaben der Unternehmen über Ist- p_{ji} , Störungsanteile und -dauer auf den Maschinen bilden eine ausreichende Basis für die Konfiguration des Fertigungssubstituts. Die notwendigen Störgrößen werden auch aus dem Abgleich der Ist- p_{ji} mit den Soll- p_{ji} nach einer Analyse [DiBS10] abgeleitet.

Das ERP-Substitut soll realitätsnahe Auftragsmeldungen generieren. Änderungen des Auftragsbestands innerhalb der bereitgestellten Fertigungsabschnitte werden aufgrund des hohen Aufwands nicht protokolliert, weshalb sie aus den Angaben der Unternehmen über Planungshorizont, Vorlaufzeit und Anteil an Eilaufträgen rekonstruiert werden. Mithilfe der fortlaufenden Auftragsnummerierung bei den beiden Unternehmen können die Erstellungszeitpunkte der Aufträge ausreichend genau abgeleitet werden (s. Abbildung 6.7). Ausgehend von den Erstellungszeitpunkten wird ein Teil der bereitgestellten Aufträge als Initialbestand definiert. Für den Rest werden Änderungsmeldungen generiert.

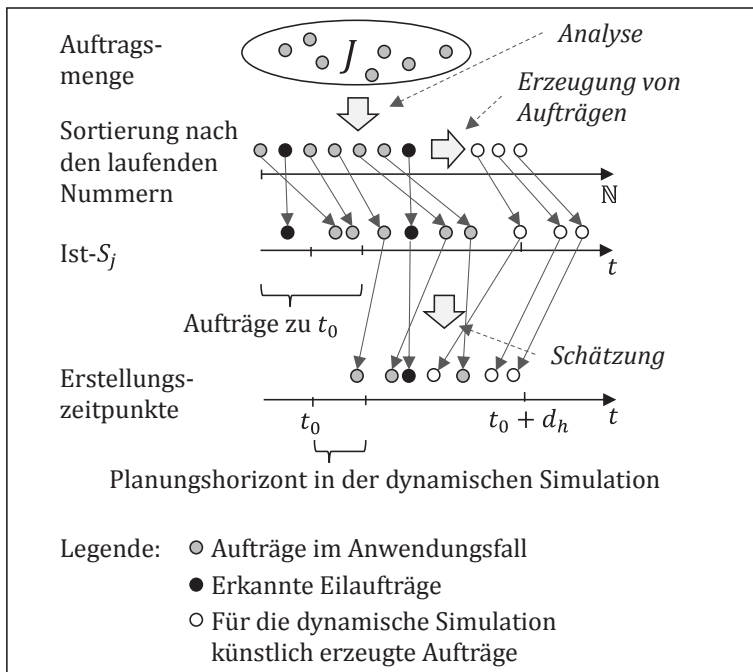


Abbildung 6.7: Einstellung der Auftragsdaten für die dynamische Simulation

Um eine fortlaufende Fertigung genauer abzubilden, werden weitere Aufträge für die Zeit nach dem bereitgestellten Fertigungsabschnitt erstellt, die die Fertigung weiterhin ähnlich hoch auslasten würden. Sie sollen das Auslaufen der Simulation verhindern, indem sie für eine kontinuierlich hohe Beanspruchung der Feinplanungsplattform sorgen. Diese Aufträge werden für die Dauer des Planungshorizonts nach dem simulierten Fertigungsabschnitt eingeplant. Die dynamische Simulation wird am Ende des bereitgestellten Fertigungsabschnitts beendet. Die Ergebniswerte für den Vergleich der Ansätze basieren nur auf den von den Unternehmen bereitgestellten Aufträgen. Die künstlich erzeugten Aufträge werden ignoriert.

Mit der dynamischen Simulationsumgebung werden in den Teilschritten 1b und 2b zuerst Untersuchungen mit Beschleunigung und ohne simulierte Abweichungen (störungsfreie Untersuchungen) durchgeführt. Das Fertigungssubstitut ist dadurch vorhersehbar und die Vergleichbarkeit zwischen den Ansätzen wird nicht durch weitere simulierte Zufälle belastet. In weiteren Simulationsläufen werden Störungen simuliert, um vorrangig die Reaktionsfähigkeit der Ansätze bei Abweichungen im Fertigungsablauf zu bewerten.

6.5 Evaluierung des Anwendungsfalls Batteriehersteller

Vor den Simulationsläufen mit der Feinplanungsplattform untersuchte das Unternehmen seine Fertigungsdaten mithilfe frei verfügbarer Bibliotheken von Feinplanungsverfahren und Feinplanungswerkzeugen. Das beste erreichte Ergebnis für diese Fertigungsdaten war die durchschnittliche Durlaufzeit (vgl. 4.2) $\emptyset DLZ = 14,83$ Tage unter Einhaltung von $\sum T_j = 0$. Dies entspricht einer Verbesserung von 19,34% gegenüber dem Ist-

Ablauf. Jedoch wurde durch Analyse des Ergebnisses weiteres Optimierungspotential aufgespürt, das die untersuchten Standardverfahren nicht komplett ausschöpfen können. Dieses Potential wird nach dem Evaluierungsplan aus Abschnitt 6.4.2 nachfolgend vertieft untersucht und die mögliche Kollaboration mit den Planern simuliert.

6.5.1 Ansätze zur Performanzsteigerung

In diesem Abschnitt wird Schritt 1 des Evaluierungsplans durchgeführt. Zunächst werden die Performanzansätze bei statischer Problemstellung und anschließend bei simulierter dynamischer Problemstellung untersucht.

Performanzsteigerung bei statischer Simulation (BAT-1a)

Im ersten statischen Simulationslauf (Teilschritt 1a) mit der Feinplanungsplattform wurden GA0-3 gegenübergestellt. Die erreichte $\emptyset DLZ$ ist in Tabelle 6.4 aufgeführt. Der GA wird mithilfe von 40 Prioritätsregeln initialisiert, die teilweise die Restriktion $\sum T_j = 0$ einhalten konnten. Somit konzentriert sich der GA alleine auf die Minimierung der $\emptyset DLZ$, weshalb alle Evaluierungskurven fallend sind (s. Abbildung 6.8). GA0 konnte eine signifikante Verbesserung der $\emptyset DLZ$ erreichen, jedoch nicht in allen Simulationsläufen die Untergrenze von 12,75 Tagen. In einem 48-stündigen Simulationslauf konnte GA3 die $\emptyset DLZ$ nach ca. 330.000 Evaluierungen sogar auf 11,34 Tage reduzieren.

Tabelle 6.4: Ergebnis für BAT-1a- $\emptyset DLZ$ in Tagen

<i>Zielsetzung</i>	<i>GA0</i>	<i>GA1</i>	<i>GA2</i>	<i>GA3</i>
$\emptyset DLZ$	12,92	12,17	12,01	11,70

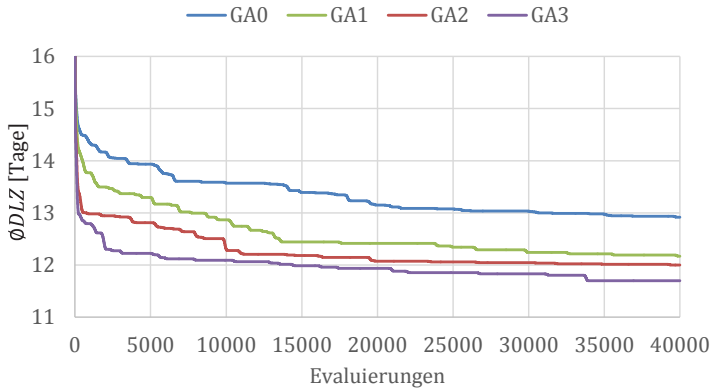


Abbildung 6.8: Ergebnis für BAT-1a-ØDLZ

Demnach ist unter Verwendung von Optimierungsstellen eine deutliche Verbesserung erzielbar. Der Grund dafür liegt in der Natur der GA und der Anzahl der Operationen pro Auftrag in diesem Anwendungsfall. Je höher diese Anzahl ist, desto unwahrscheinlicher wird es für GA0 alle Operationen eines Auftrags richtig zu positionieren. Sobald eine einzige Operation von den anderen Operationen desselben Auftrags zeitlich weit entfernt platziert wird, steigt die *DLZ* des Auftrags und dadurch die *ØDLZ* des Individuums und somit die Wahrscheinlichkeit in der Selektionsphase nicht ausgewählt zu werden. GA1-3 dagegen koordinieren alle ihre Veränderungen im Individuum und produzieren deshalb meist bessere Fertigungspläne.

Als nächstes werden mehrere Verfahrensinstanzen (VI) für ein paralleles Optimieren konfiguriert. Dieser Ansatz wurde mit Blick auf WAF entworfen, wird hier aber auf BAT angewendet. Die Ergebnisse von jeweils 2, 4 und 6 parallelen Verfahrensinstanzen bei einer Größe des Markierungsbereichs von ~12% (12% MB) des Fertigungsplans (~1500 Operationen) werden mit Läufen von GA3 bei Parallelisierung der Evaluierung auf 2 und 4 Threads verglichen (s. Abbildung 6.9). Da die von den parallelen Verfahrensinstanzen erreichten Zielfunktionswerte nicht zufriedenstel-

lend waren, wurde die angestrebte Größe für eine Markierung auf 28% hochgesetzt. Bei dieser Größe können zwei Verfahrensinstanzen ohne gegenseitige Behinderungen parallel ausgeführt werden. Das Ergebnis dieser Simulationsläufe ist ebenfalls in Abbildung 6.9 ersichtlich.

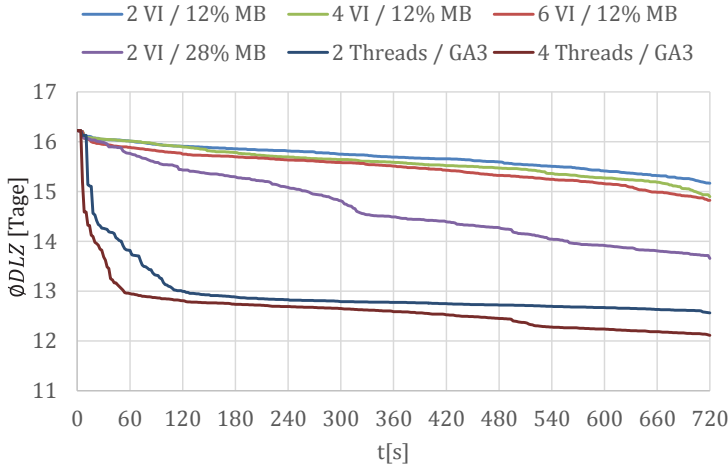


Abbildung 6.9: Ergebnis für BAT-1a-VI

Die Ergebnisse zeigen, dass die Optimierung der Verfahrensinstanzen auf 12% des Fertigungsplans aufgrund der hohen $\emptyset|O_j|$ nicht so effizient ist wie GA3 auf dem gesamten Fertigungsplan. Auf 28% des Fertigungsplans (28% MB) dagegen treten bei den Verfahrensinstanzen stetig Verbesserungen ein, jedoch deutlich langsamer als bei GA3 auf dem gesamten Fertigungsplan. Die Problemgröße erlaubt in diesem Fall nur bis zu zwei parallele Verfahrensinstanzen. Die Effizienz der Verfahrensinstanzen hängt am verwendeten GA, der einen ausreichend großen Optimierungsraum für diese spezifische Fertigungsart benötigt. Aus diesem Grund wird in diesem Anwendungsfall in den nachfolgenden Simulationsläufen auf diesen Ansatz verzichtet.

Performanzsteigerung bei dynamischer Simulation (BAT-1b)

Für Teilschritt 1b wird aus BAT eine dynamische Problemstellung nach der Vorgehensweise aus dem vorigen Unterkapitel konstruiert. Üblicherweise treten bei BAT Aufträge mit ein- bis zweiwöchiger Vorlaufzeit auf. Aufgrund der langen, teilweise über 20-tägigen Ist-Durchlaufzeiten beträgt der Planungshorizont in der Fertigungssteuerung 6 Wochen. Das Unternehmen bekommt gelegentlich auch Eilaufträge ohne Vorlaufzeit. Aus der Analyse des Ist-Ablaufs und der laufenden Nummer der Aufträge konnten 8 Eilaufträge (~9%) ermittelt werden.

Aus den obigen Angaben wurde die dynamische Problemstellung konstruiert. Die Abbildung 6.10 zeigt die untersuchten Relevanzfunktionen. Die dynamische Simulation wurde mit einem verspätungsfreien Fertigungsplan für die Problemstellung zu t_0 initialisiert. Dieser Fertigungsplan wurde mittels GA0 nach 5000 Evaluierungen erstellt.

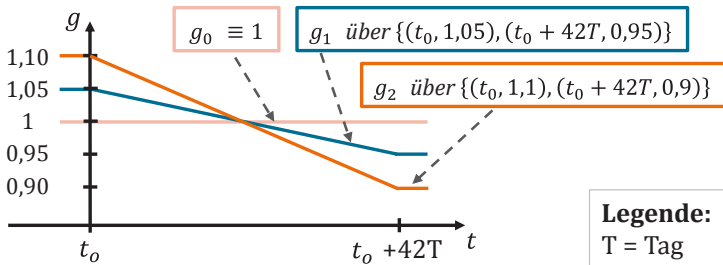


Abbildung 6.10: Untersuchte Relevanzfunktionen in BAT-1b

Es wurden g_0 , g_1 und g_2 in der dynamischen Simulationsumgebung evaluiert und die Ergebnisse analysiert (s. Tabelle 6.5). Eine deutliche Reduzierung der $\emptyset DLZ$ konnte beim Einsatz der zeitbehafteten Version für den Planvergleich erzielt werden. Die stärkere Gewichtung der naheliegenden Fragmente konnte ihren Vorteil bei simulierten Fertigungsstörungen hervorheben. Eine zu starke Gewichtung wie bei g_2 konnte diesen Vorteil jedoch wieder ab-

mindern. Interessant hier ist das Verhalten von g_2 , das in einer störungsfreien Simulation ein schlechteres Ergebnis lieferte, aber seine Stärke unter Störungen zeigen konnte.

Die erbrachte Verbesserung mithilfe der Kombinationen $g_o|g_1$ bzw. $g_o|g_2$ war minimal. Dies bedeutet, dass die Läufe mit den Relevanzfunktionen nach der klassischen $\emptyset DLZ$ deutlich schlechtere Fertigungspläne im Simulationsablauf erzielen, aber dennoch ein besseres Gesamtergebnis erreichen.

Tabelle 6.5: Simulationsergebnisse für BAT-1b in Tagen

g	g_o	g_1	g_2	$g_o g_1$	$g_o g_2$
<i>störungsfrei</i>	<i>13,10</i>	<i>12,74</i>	<i>13,37</i>	<i>13,03</i>	<i>12,99</i>
<i>Störungen</i>	<i>15,21</i>	<i>13,11</i>	<i>13,52</i>	<i>15,02</i>	<i>14,91</i>

Um diesen Effekt zu erklären, werden zwei einzelne Simulationsläufe mit g_o und g_1 in Simulationen mit Störungen in Abbildung 6.11 verglichen. Dabei wird für jeden Tag der mittlere Planwert der Zielfunktion und der bis zu diesem Tag akkumulierte Istwert aus der dynamischen Simulation errechnet. Aus Sicht eines Planners hat die Feinplanungsplattform mit g_o überwiegend einen besseren Fertigungsplan. Obwohl bei g_1 der auszuführende Fertigungsplan aus theoretischer Sicht häufig schlechter war, konnten längere Aufträge früher beendet werden. Dies wirkt sich zunächst auf den Istwert negativ aus, während der Planwert reduziert wurde. Da die Aufträge im naheliegenden Fragment des Fertigungsplans sehr kompakt disponiert sind, war g_1 deutlich störungsanfälliger, wie der Verlauf des Planwerts zeigt. Jedoch gelang es mit g_1 wieder einen besseren Gesamtfertigungsplan herzustellen. Am Ende konnte ein deutlich besserer Gesamt-Istwert für das Evaluierungsintervall erzielt werden. Dieser Effekt war nur in einer dynamischen Simulation erkennbar.

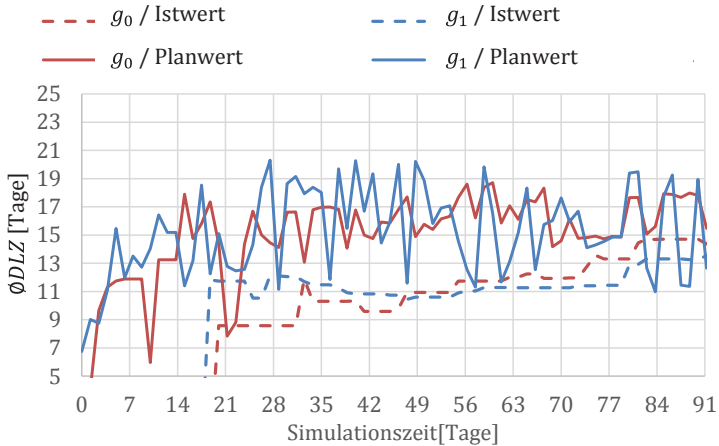


Abbildung 6.11: Zwei repräsentative Simulationsläufe aus BAT-1b in dynamischen Simulationsläufen mit Störungen

6.5.2 Kollaborationsansätze

In Schritt 2 steht die Kollaboration mit dem Planer im Fokus. Analog zu 6.5.1 werden zuerst statische und anschließend dynamische Simulationsläufe durchgeführt.

Kollaborationsansätze bei statischer Simulation (BAT-2a)

Zuerst werden die Planübernahmen in den Simulationsläufen von GA3 aus BAT-1a analysiert. Jede Planübernahme bringt eine gewisse Perturbation im Fertigungsplan mit sich. In den Simulationsläufen in Teilschritt 1a wanderte jede Operation durch Umlanungen im Durchschnitt 29,41 Tage innerhalb des Fertigungsplans, wobei zwei Drittel davon (19,32) innerhalb der ersten 10.000 Evaluierungen stattfanden. Eine Analyse der Übernahmen zeigt, dass häufig Verbesserungen von $\sim 0,3\%$ die Planstabilität mit $\vartheta/|O| \approx 1$ belastet hatten. Andere Übernahmen konnten bei einer ähnlichen Perturbation $\sim 3\%$ Verbesserung erreichen. Im

Schnitt trug jede Übernahme ca. 0,98 zu $\vartheta/|O|$ bei. $\vartheta/|O|$ zwischen dem Ist-Ablauf und dem bestgefundenen Fertigungsplan in BAT-1a beträgt 5,90.

Um die Feinplanungsplattform auf die Kollaboration mit den Planern einzurichten, wurde GA3 als die beste Konfiguration aus Schritt 1a zusätzlich um die Einhaltung der Planstabilität und der Benutzerinteraktionen erweitert. Als akzeptable GW für die Planer des Unternehmens galten die folgenden Werte: für θ ein zulässiger GW von 2 und ein unproblematischer GW von 0,2, für ϑ^b 0,2 bzw. 0,01. Für die beiden zulässigen GW wurde eine Verbesserung von mindestens 2% festgelegt. Dabei wird eine Verschiebung auf eine andere Maschine innerhalb der ersten Woche des Fertigungsplans oder auf einen anderen Mitarbeiter in den ersten drei Tagen als eine Schicht (= 0,33 Tage) quantifiziert.

Im ersten Simulationslauf wird nur die Einhaltung der Planstabilität evaluiert. Der Vergleich in Abbildung 6.12 zeigt, wie die $\varnothing DLZ$ bei der Anwendung verschiedener Ansätze für Planstabilität konvergiert. Als Basis dienen die Simulationsläufe von GA3 aus Teilschritt 1a.

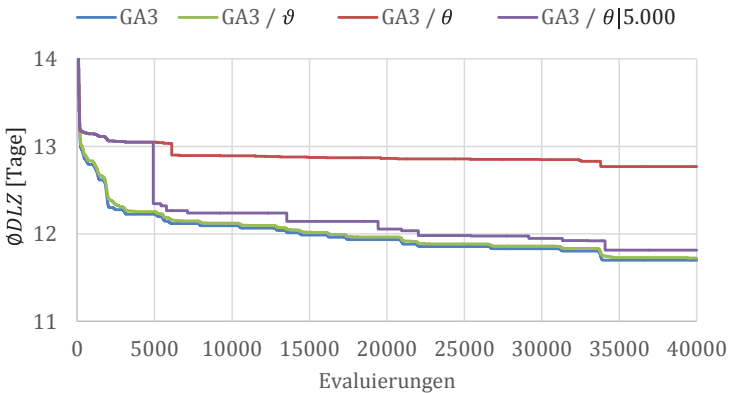


Abbildung 6.12: Einhaltung der Planstabilität in statischen Simulationen (BAT-2a- θ)

Die Verwendung von ϑ verhindert große Perturbationen bei Übernahmen. Anstatt einen Fertigungsplan mit großer Veränderung zu übernehmen, wurden mit diesem klassischen Ansatz mehrere kleine Übernahmen durchgeführt. Deshalb verhielt sich dieser Ansatz oft kontraproduktiv.

Durch die Einführung von θ konnte eine deutliche Verbesserung der Planstabilität erreicht werden. Aufgrund der festen Einschränkung von θ wurden nach einer gewissen Zeit keine neuen Übernahmen zugelassen. Deshalb wurde θ in einem vierten Simulationslauf nur aus den letzten 5.000 Evaluierungen berechnet ($\theta|5.000$). Hier konnte die in Teilschritt 1a erzielte $\varnothing DLZ$ wiederholt werden, während $\vartheta/|O|$ auf unter $1/6$ gesenkt wurde. Damit wurde das Bereitstellen von Zwischenergebnissen aus dem statischen Simulationslauf für einen Planer deutlich komfortabler.

Nachfolgend wird die Einhaltung der Benutzerinteraktionen (BI) evaluiert. Das Ergebnis aus den Simulationsläufen mit starrer Fixierung und ϑ^b ist in Tabelle 6.6 und Abbildung 6.13 dargestellt. Bei 2% BI konnte der GA auch bei starrer Fixierung noch ungehindert optimieren. Bei 5% BI ist der Lösungsraum so stark eingeschränkt, dass nur minimale Verbesserungen der $\varnothing DLZ$ erreichbar sind. Der Einsatz von ϑ^b konnte den Lösungsraum erweitern und so dem Feinplanungsverfahren eine deutliche Verbesserung der $\varnothing DLZ$ ermöglichen. Dies war trotz der Verwendung von $\theta|5.000$ möglich.

Tabelle 6.6: Ergebnis für BAT-2a- ϑ^b in Tagen

<i>BI</i>	<i>Fixiert</i>	<i>ϑ^b</i>
<i>2%</i>	<i>13,87</i>	<i>13,44</i>
<i>5%</i>	<i>17,33</i>	<i>14,57</i>

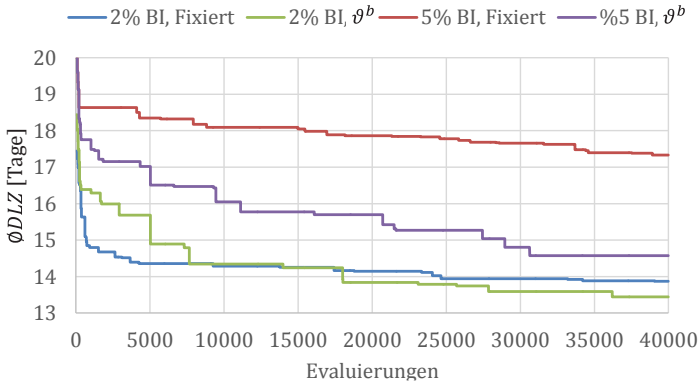


Abbildung 6.13: Berücksichtigung der Benutzerinteraktionen in statischen Simulationen (BAT-2a- ϑ^b)

Kollaborationsansätze bei dynamischer Simulation (BAT-2b)

In Teilschritt 2b werden die Relevanzfunktionen für θ und ϑ^b gemäß den Angaben des Planers geschätzt. Für die Gewichtung der vorigen Übernahmen in $\theta(g^\theta)$ und des Alters der Benutzerinteraktionen in $\vartheta^b(g^b)$ werden jeweils zwei Funktionen untersucht. Sie sind in Abbildung 6.14 dargestellt. Interaktionen bzw. Übernahmen des letzten Tages werden dadurch hoch gewichtet.

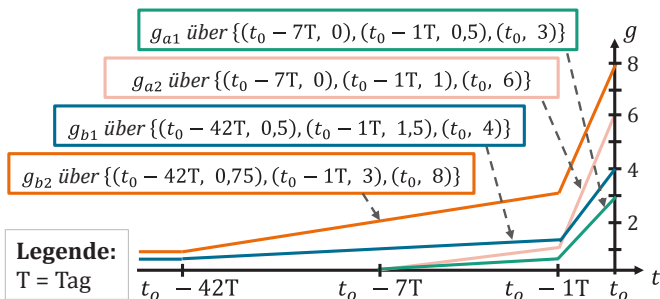


Abbildung 6.14: Untersuchte Relevanzfunktionen in BAT-2b

Zunächst wird die Einhaltung der Planstabilität in der dynamischen Simulation untersucht (BAT-2b- θ). Als Grundlage dient hier

GA3 mit $g^{\theta DLZ} := g_1$. In Tabelle 6.7 sind die Ergebnisse aus den dynamischen Simulationsläufen mit den unterschiedlichen Ansätzen für die Einhaltung der Planstabilität abgebildet. Die beiden simulierten Relevanzfunktionen zeigen ihre Unterschiede bei simulierten Störungen. Die Kombination aus den steileren Relevanzfunktionen beschränkt die Reaktionsfähigkeit des GAs im naheliegenden Bereich, was sich auf das Endergebnis negativ auswirkt. Außerdem kann die Planstabilität nur minimal verbessert werden.

Tabelle 6.7: Einhaltung der Planstabilität in dynamischen Simulationen (BAT-2b- θ) in Tagen

g	<i>ohne</i>	$g^\theta = g_{a1}$ $g^\vartheta = g_1$	$g^\theta = g_{a2}$ $g^\vartheta = g_2$
<i>störungsfrei</i>	12,74 ($\vartheta/ O =1249$)	13,38 ($\vartheta/ O =127$)	13,77 ($\vartheta/ O =110$)
<i>Störungen</i>	13,11 ($\vartheta/ O =1762$)	14,23 ($\vartheta/ O =188$)	15,41 ($\vartheta/ O =167$)

θ in der ersten Kombination ($g^\theta = g_{a1}$, $g^\vartheta = g_1$) hat nur geringe Auswirkungen auf das Gesamtergebnis, der Planwert während des Simulationsablaufs ist aber aufgrund der reduzierten Übernahmehäufigkeit deutlich höher. Abbildung 6.15 zeigt einen Vergleich eines einzelnen Simulationslaufs mit dem blauen Simulationslauf ($g^{\theta DLZ} = g_1$) aus Abbildung 6.11 (BAT-1b), hier rot dargestellt. Die schlechtere Reaktion auf Störungen ist am Verlauf des Planwerts bei θ deutlich zu sehen.

Als nächstes wird die Einhaltung von Benutzerinteraktionen in einer dynamischen Simulation evaluiert (BAT-2b- ϑ^b). Hier werden zwei Kombinationen ($g^b = g_{b1}$, $g^{\vartheta^b} = g_1$) und ($g^b = g_{b2}$, $g^{\vartheta^b} = g_2$) betrachtet. Wie auch in den statischen Simulationsläufen hat die Fixierung die Benutzerwünsche zwar exakt eingehal-

ten, den Lösungsraum aber besonders bei 5% BI spürbar eingeschränkt. Auch bei der starren Fixierung wurden Abweichungen zu den Benutzerwünschen festgestellt, die durch Störungen verursacht werden.

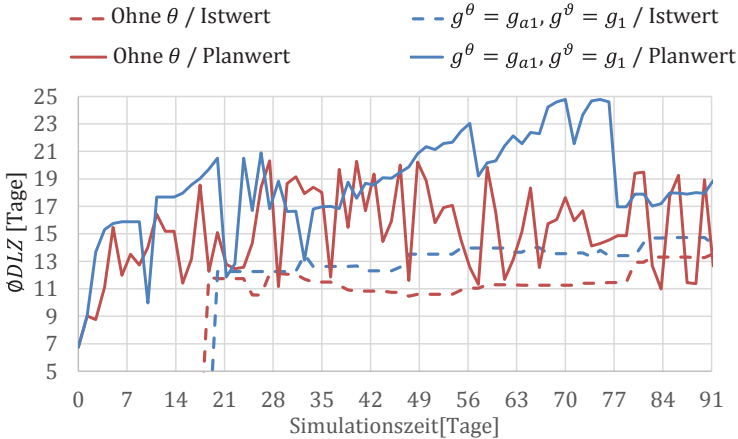


Abbildung 6.15: Zwei repräsentative Simulationsläufe aus BAT-2b- θ in dynamischen Simulationsläufen mit Störungen

Tabelle 6.8: Berücksichtigung der Benutzerinteraktionen in dynamischen Simulationen (BAT-2b- ϑ^b) in Tagen

g	fixiert	$g^b = g_{b1}$ $g^{\vartheta^b} = g_1$	$g^b = g_{b2}$ $g^{\vartheta^b} = g_2$
störungsfrei, 2% BI	13,42 ($\vartheta^b / O = 0,04$)	12,91 ($\vartheta^b / O = 0,62$)	13,40 ($\vartheta^b / O = 0,32$)
Störungen, 2%, BI	14,67 ($\vartheta^b / O = 0,21$)	13,46 ($\vartheta^b / O = 0,72$)	13,95 ($\vartheta^b / O = 0,38$)
störungsfrei, 5%, BI	20,71 ($\vartheta^b / O = 0,11$)	13,76 ($\vartheta^b / O = 0,75$)	14,89 ($\vartheta^b / O = 0,47$)
Störungen, 5% BI	22,55 ($\vartheta^b / O = 0,23$)	14,88 ($\vartheta^b / O = 0,82$)	16,31 ($\vartheta^b / O = 0,56$)

Wie aus der Tabelle ersichtlich ist, bringen die hier entworfenen Konzepte für Planstabilität auch für die Einhaltung der Benutzerinteraktionen deutliche Vorteile mit sich. Die Benutzerinteraktionen werden bis auf kleine Abweichungen eingehalten und das Feinplanungsverfahren hat sowohl einen größeren Optimierungsraum als auch eine höhere Reaktionsgeschwindigkeit, wie die Optimierungsverläufe von ausgewählten Konfigurationen in Abbildung 6.16 zeigen. Harte Fixierungen dagegen schränken den Optimierungsraum des Feinplanungsverfahrens stark ein. Dies ist besonders bei der Simulation mit Störungen ersichtlich.

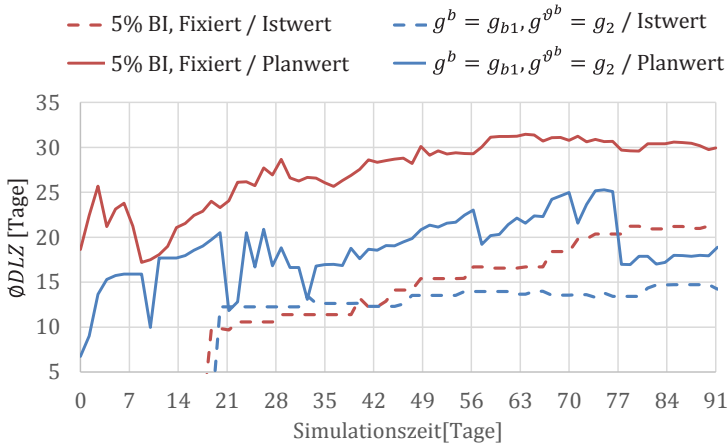


Abbildung 6.16: Zwei repräsentative Simulationsläufe aus BAT-2b- ϑ^b -5%BI in dynamischen Simulationsläufen mit Störungen

Wie aus den Ergebnissen zu erkennen ist, erreicht die Feinplanungsplattform mit den Einstellungen ($g^{\vartheta^{DLZ}} = g_1$, $g^{\vartheta} = g_{a1}$, $g^b = g_{b1}$, $g^{\vartheta} = g^{\vartheta^b} = g_1$) sowohl eine Verbesserung der Planungsgüte als auch der Bedienbarkeit durch die Planer. Bestehende Konzepte aus dem Stand der Technik hatten sich hier als ungeeignet für eine kontinuierliche Feinplanung in einer dynamischen Umgebung erwiesen.

6.6 Evaluierung des Anwendungsfalls Wafer-Fertiger

Die von WAF bereitgestellten Fertigungsdaten sind ebenso umfangreich wie die von BAT und bilden aufgrund der Rüstzeiten, der hohen Anzahl an alternativen Maschinen sowie der multikriteriellen Zielsetzung eine sehr anspruchsvolle Problemstellung. Das Unternehmen hat mit einfachen Maßnahmen versucht, die hohen Rüstzeiten in der Fertigung zu reduzieren. Bei der Analyse vergangener Fertigungsperioden hat auch dieses Unternehmen Optimierungspotential erkannt.

6.6.1 Ansätze zur Performanzsteigerung

Analog zum vorigen Anwendungsfall werden statische und dynamische Simulationsläufe nach Schritt 1 des Evaluierungsplans durchgeführt.

Performanzsteigerung bei statischer Simulation (WAF-1a)

Bevor die multikriterielle Problemstellung in Teilschritt 1a evaluiert wird, wird für die Untersuchung des Lösungsraums jeweils ein Simulationslauf mit einem Kriterium als Optimierungsziel durchgeführt, während der Wert des anderen Kriteriums vom Ist-Ablauf als harte Restriktion dient.

In Teilschritt 1a ($1a-\sum T_j$) wurde die Minimierung von $\sum T_j$ bei $TST \leq 30,7$ angestrebt. Bei Gleichheit von $\sum T_j$ bestimmt der Wert von TST das bessere Individuum. Sowohl GA0 als auch GA1-3 wurden ausgeführt (s. Tabelle 6.9 und Abbildung 6.17).

Die Ergebnisse zeigen eine deutlich schnellere Konvergenz unter Nutzung von Optimierungshinweisen, auch wenn die modifizierten Operatoren nur mit 25% Wahrscheinlichkeit ausgeführt

wurden. Beim ausschließlichen Einsatz der modifizierten Operatoren konvergiert der Zielfunktionswert signifikant schneller.

Tabelle 6.9: Ergebnis für WAF-1a in Tagen

Zielsetzung (Randbedingung)	GA0	GA1	GA2	GA3
$\sum T_j$ ($TST \leq 30,7$)	155,3	80,3	61,0	41,5
TST ($\sum T_j \leq 1192,4$)	23,87	19,23	18,73	17,82

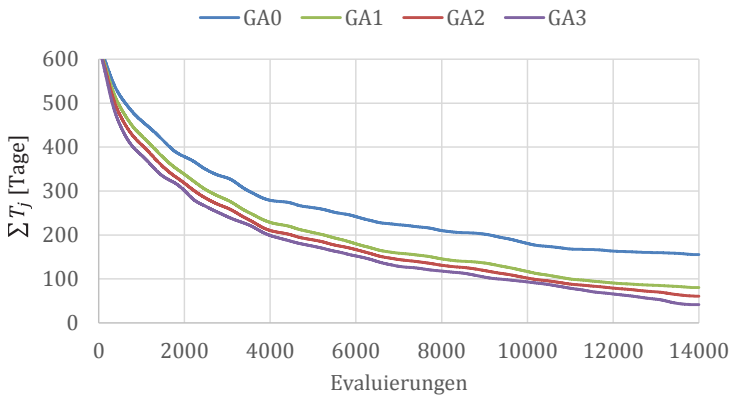


Abbildung 6.17: Ergebnis für WAF-1a- $\sum T_j$

Im zweiten Simulationslauf (1a- TST) wurde der GA auf Minimierung von TST bei $\sum T_j \leq 1192,4$ eingestellt. Bei Gleichheit von TST bestimmt der Wert von $\sum T_j$ das bessere Individuum. Tabelle 6.9 und Abbildung 6.18 zeigen die Ergebnisse der Simulationsläufe. Mit einem GA in klassischer Ausführung ist es nicht möglich Rüstzeiten zu optimieren. GA0 erzielte mit einer TST von 23,9 Tagen nur eine geringe Verbesserung gegenüber den Prioritätsregeln. Mit GA1-3 konnten die Rüstzeiten dagegen deutlich reduziert werden.

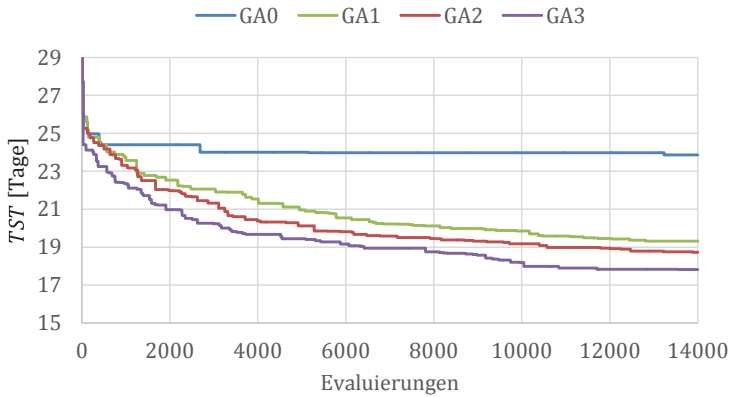


Abbildung 6.18: Ergebnis für WAF-1a-TST

Zusammengefasst zeigten diese beiden Simulationsläufe die deutlichen Vorteile bei der Verwendung von Optimierungshinweisen. Auf diese Weise wird mittels GA die Optimierung der TST möglich. Dieser Eindruck wurde in zwei Langläufen bestätigt. In einem 24-stündigen Simulationslauf konnte GA3 nach Evaluierung von 140.000 Individuen $\sum T_j = 0$ bei $TST \leq 30,7 \cdot 150\%$ erreichen. Der dabei erreichte Wert stimmt mit $LB_{\sum T_j}$ überein. In einem weiteren 24-stündigen Simulationslauf konnte GA3 nach der Evaluierung von 165.000 Individuen $TST = 14,86$ Tage bei $\sum T_j \leq 1192,4 \cdot 150\%$ erreichen. Der erreichte Wert entspricht ungefähr $2 \cdot LB_{TST}$.

Im nächsten Schritt wurde die multikriterielle Zielsetzung ZF auf Basis der Ergebnisse, des abgetasteten Lösungsraums und der approximierten Pareto-Front (s. Abbildung 6.19) definiert als

$$ZF := \sqrt{(\sum T_j / 24)^2 + (TST - 7)^2}$$
 Diese Formel bildet den euklidischen Abstand zum globalen Kriterium definiert durch $(LB_{\sum T_j}, LB_{TST})$. Dabei sollen die beiden harten Restriktionen $\sum T_j \leq 1192,4$ und $TST \leq 30,7$ eingehalten werden. Aufgrund der hohen Stundensätze für die Einrichter war das Unternehmen

bereit für jede gesparte Stunde Rüstzeit einen Tag zusätzliche Verspätung in Kauf zu nehmen, weshalb die quadratische Summe entsprechend gewichtet ist. Alternative Express-Versandoptionen bei kleinen Produktdimensionen erlauben hier anders als bei BAT gewisse Terminüberschreitungen.

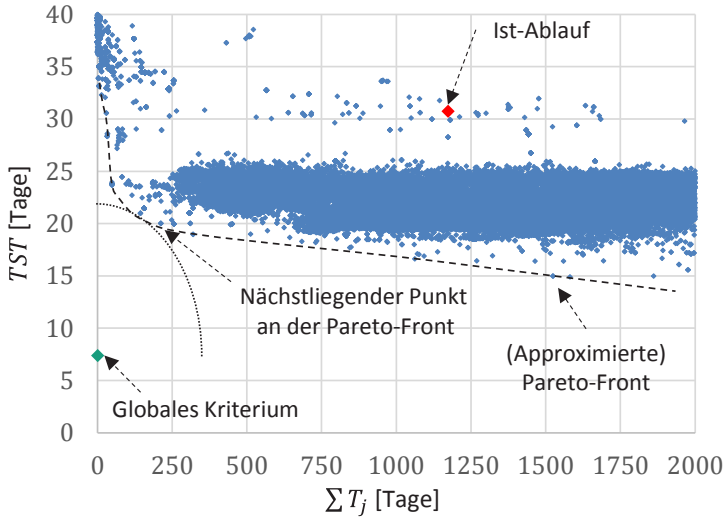


Abbildung 6.19: Untersuchte Fertigungspläne in Teilschritt WAF-1a

Die Ergebnisse für die Zielsetzung ZF sind in Tabelle 6.10 und Abbildung 6.20 ersichtlich. Da es mit GA0 nicht möglich ist, TST zu optimieren, war das erzielte Ergebnis von 28,83 Tagen nur durch Reduzierung von $\sum T_j$ möglich. GA1-3 dagegen konnten rüstoptimale Operationsfolgen bilden und sich dadurch schneller an das globale Kriterium annähern.

Tabelle 6.10: Ergebnis für WAF-1a-ZF in Tagen

Zielsetzung (Randbedingung)	GA0	GA1	GA2	GA3
ZF ($TST \leq 30,7; \sum T_j \leq 1192,4$)	28,83	19,98	18,94	17,15

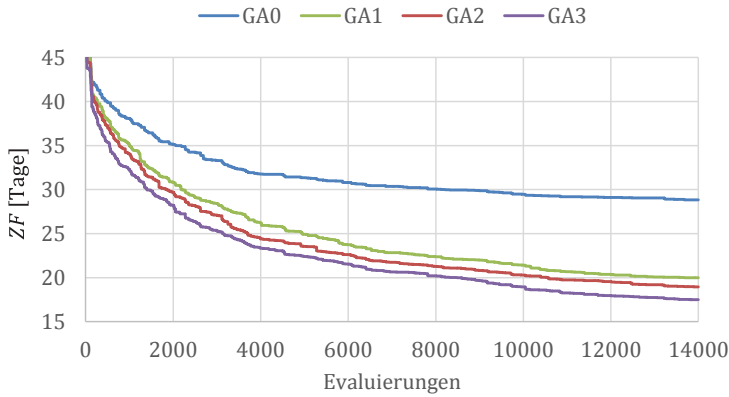


Abbildung 6.20: Untersuchte Fertigungspläne in Teilschritt WAF-1a-ZF

Im nächsten Evaluierungslauf werden mehrere Verfahrensinstanzen für ein paralleles Optimieren konfiguriert. Wie bei BAT wurde zuerst eine Markierungsgröße von $\sim 12\%$ (~ 1600 Operationen) untersucht (12% MB). Die Ergebnisse in Abbildung 6.21 zeigen, dass jede zusätzliche Verfahrensinstantz gleichmäßig zur Optimierungsgeschwindigkeit beiträgt. Die gesamte Evaluierungsgeschwindigkeit stieg im Vergleich zu GA3 bei gleicher Anzahl von Threads deutlich.

Die Simulationsumgebung kann 6 Verfahrensinstanzen (VI) unabhängig voneinander ausführen. Die Intels Hyper-Threading Technologie¹¹ erlaubt jedoch theoretisch auch 12 parallele Verfahrensinstanzen mit kleineren Laufzeiteinbußen. Deshalb wurde die Feinplanungsplattform mit kleineren Markierungsgrößen von ca. $\sim 4\%$ (~ 530 Operationen) untersucht (4% MB). Denn mit steigender Anzahl an Verfahrensinstanzen erhöht sich der Anteil des Fertigungsplans, der für neue Markierungen ungeeignet ist. Die Simulationsergebnisse in Abbildung 6.21 zeigen hier eine genauso hohe Optimierungsgeschwindigkeit wie bei GA3. Das Hinzufügen von weiteren Verfahrensinstanzen bei dieser kleinen

¹¹ <http://www.intel.com/technology/platform-technology/hyper-threading>

Markierungsgröße hat keine wesentliche Auswirkung auf das Simulationsergebnis, da eine kleinere Markierungsgröße dem GA keinen ausreichenden Optimierungsraum bereitstellen kann.

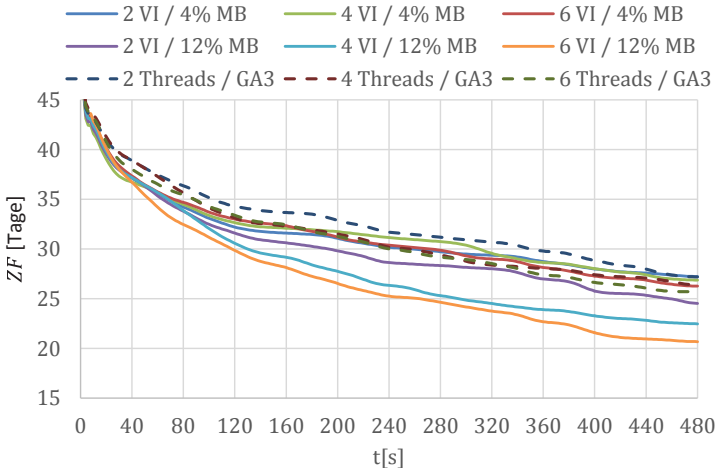


Abbildung 6.21: Ergebnis für WAF-1a-VI

Die laufende Problemstellung reduziert sich in den nachfolgenden dynamischen Simulationsläufen auf ca. 33% gegenüber der statischen Problemstellung. Grund dafür ist die Fertigungsorganisation bei WAF mit kurzen Planungshorizonten. Bei Markierungsgrößen von ca. 800 Operationen sind deshalb maximal zwei Verfahrensinstanzen parallel ausführbar. Sie würden in einer dynamischen Simulation zu einer nicht signifikanten Verbesserung gegenüber GA3 führen, weshalb auf diesen Ansatz in den nachfolgenden Simulationsläufen verzichtet wird.

Performanzsteigerung bei dynamischer Simulation (WAF-1b)

Für Teilschritt 1b wird aus WAF eine dynamische Problemstellung nach der Vorgehensweise in Abschnitt 6.4.3 konstruiert. Üblicherweise werden bei WAF Aufträge mit bis zu zwei Wochen Vorlaufzeit eingefügt. Der Planungshorizont in der Fertigungs-

steuerung beträgt 4 Wochen. Bei diesem Unternehmen treten häufiger Eilaufträge ohne Vorlaufzeit auf. Es konnten hier 263 Eilaufträge ($\sim 12\%$) identifiziert werden.

Mithilfe dieser Werte wurde die dynamische Problemstellung konstruiert, mit der unterschiedliche Relevanzfunktionen in Teilschritt 1b untersucht werden. Um die passende Relevanzkurve je Kriterium zu finden, werden wie in Teilschritt 1a einzelne Simulationsläufe für je $\sum T_j$ und TST durchgeführt. Anschließend werden die besten Relevanzfunktionen je Kriterium für die Optimierung von ZF angewendet. Die Abbildung 6.22 zeigt die untersuchten Relevanzfunktionen.

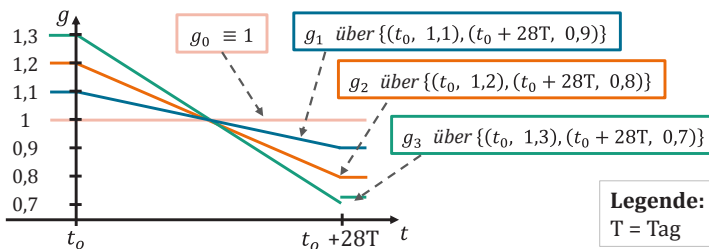


Abbildung 6.22: Untersuchte Relevanzfunktionen in WAF-1b

Die dynamische Simulation wurde mit einem Fertigungsplan für die Problemstellung zu t_0 initialisiert, der mittels GA0 nach 600 Evaluierungen erstellt wurde. Es wurden g_0, g_1, g_2 für $\sum T_j$ und g_0, g_2, g_3 für TST evaluiert. Die Ergebnisse sind in Tabelle 6.11 bzw. Tabelle 6.12 zu finden. Bei der Evaluierung von $\sum T_j$ wurde deutlich, dass eine noch steilere Relevanzfunktion wie g_3 keine Verbesserung erreichen wird.

Tabelle 6.11: Simulationsergebnisse für WAF-1b- $\sum T_j$ in Tagen

g	g_0	g_1	g_2	$g_0 g_1$	$g_0 g_2$
<i>störungsfrei</i>	48,7	41,0	45,1	47,4	47,8
<i>Störungen</i>	72,1	43,7	49,9	65,3	66,8

Tabelle 6.12: Simulationsergebnisse für WAF-1b-TST in Tagen

g	g_0	g_1	g_2	g_3	$g_0 g_2$	$g_0 g_3$
<i>störungsfrei</i>	22,9	20,5	18,6	19,1	22,4	22,6
<i>Störungen</i>	27,7	22,8	19,8	22,3	26,6	27,0

Eine deutliche Reduzierung der $\sum T_j$ konnte beim Einsatz der zeitbehafteten Version für den Planvergleich erzielt werden. Die stärkere Gewichtung der naheliegenden Fragmente konnte ihren Vorteil bei simulierten Fertigungsstörungen herausstellen. Durch zu starke Gewichtung wie bei g_2 wurde dieser Vorteil jedoch wieder abgeschwächt. Die Simulationsläufe mit $g_0|g_1$ und $g_0|g_2$ konnten nur kleine Verbesserungen gegenüber g_0 bringen.

Steilere Relevanzfunktionen konnten zur Reduzierung der TST deutlich mehr beitragen als bei $\sum T_j$. Während $\sum T_j$ eher durch zeitliche Verschiebung von Operationen zu minimieren ist, kann TST alleine durch Maschinenwechsel minimiert werden. Somit konnten längere Rüstzeiten erst kurz vor deren Fälligkeit optimiert werden, ohne das zweite Bewertungskriterien zu verschlechtern.

Zur Erklärung dieses Effekts werden analog zu BAT-1b die Verläufe von jeweils zwei typischen Simulationsläufen pro Zielfunktion nebeneinander gestellt. Für $\sum T_j$ wird je ein Lauf mit g_0 und g_1 in Abbildung 6.23 dargestellt. Für TST zeigt Abbildung 6.24 je einen Lauf mit g_0 und g_2 .

Mit g_1 bzw. g_2 wurde ein besserer simulierter Ist-Ablauf erzielt. Eine große Rolle spielen hier die Rüstzeiten. Jede Änderung im naheliegenden Fragment des Fertigungsplans ändert die Abarbeitungsreihenfolgen auf den Maschinen. Dies führt oft zu kleinen Veränderungen in der Rüstzeit und möglicherweise zu einer Reihe

von Verschiebungen abhängiger Operationen. Dieser Effekt ist besonders bei WAF-1b- TST anhand des Planwerts zu beobachten. Da die Relevanzfunktionen den GA zwingen, vorrangig naheliegende Fragmente zu optimieren, tritt dies dort seltener ein.

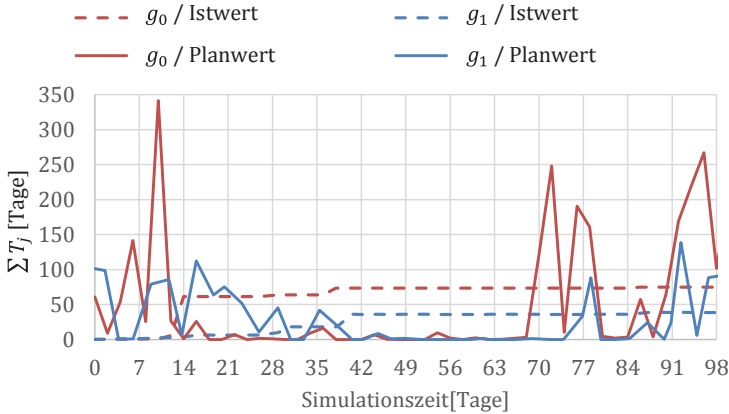


Abbildung 6.23: Zwei repräsentative Simulationsläufe aus WAF-1b- ΣT_j in dynamischen Simulationsläufen mit Störungen

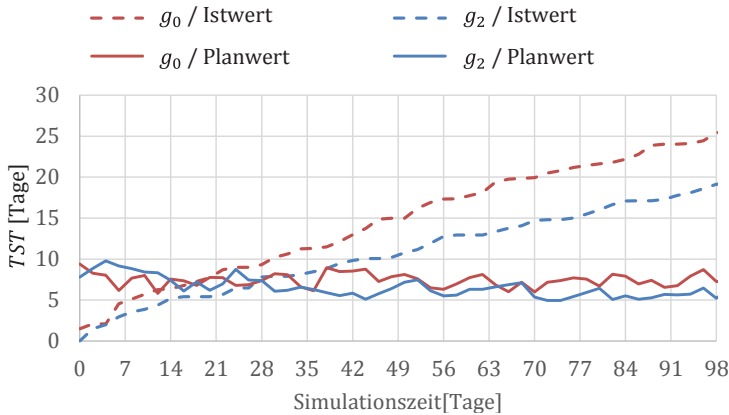


Abbildung 6.24: Zwei repräsentative Simulationsläufe aus WAF-1b- TST in dynamischen Simulationsläufen mit Störungen

Im Anschluss werden die besten und zweitbesten Relevanzfunktionen in die multikriterielle Zielfunktion ZF überführt. Zunächst wird $ZF := \sqrt{(\sum T_j / 24)^2 + (\max\{0, TST - 7 \cdot |O| / 13184\})^2}$ für die dynamische Problemstellung festgelegt. Der Unterschied zur Formulierung für die statischen Simulationsläufe liegt in der Korrektur der LB_{TST} abhängig von $|O|$. Entsprechend umgeschrieben werden die harten Restriktionen als $\sum T_j \leq 1192,4 \cdot |J| / 2223$ und $TST \leq 30,7 \cdot |O| / 13184$. Da jede Relevanzfunktion g so definiert wurde, dass $\int_{t_0}^{t_0+d_h} g(t)dt = 1$ gilt, wird keine weitere Normierung (vgl. 5.3) erforderlich.

Mit allen Kombinationen der Relevanzfunktionen für ZF sowie mit der klassischen Version von ZF wurden dynamische Simulationsläufe durchgeführt (s. Tabelle 6.13). Auch hier wird der Vorteil der zeitbehafteten Bewertungskriterien bei Störungen deutlich. Der Durchschnitt der Simulationsläufe mit g_0 ist nicht nur schlechter, sondern ist g_0 mit $\sigma^2 = 35,7$ bei Störungen instabil. Bei den zeitbehafteten Bewertungskriterien dagegen ist $\sigma^2 < 3$.

Tabelle 6.13: Simulationsergebnisse für WAF-1b-ZF mit zwei Kombinationen aus Relevanzfunktionen in Tagen

g	$g^{\sum T_j} = g_0$ $g^{TST} = g_0$	$g^{\sum T_j} = g_1$ $g^{TST} = g_2$	$g^{\sum T_j} = g_2$ $g^{TST} = g_3$
<i>störungsfrei</i>	28,1	19,1	20,3
<i>Störungen</i>	37,7	21,6	26,7

Auch hier werden zwei repräsentative Simulationsläufe (s. Abbildung 6.25) gegenübergestellt. Die deutlich höhere Reaktionszeit mit den Relevanzfunktionen ist auch hier wiederum am geglätteten Kurvenverlauf des Planwerts erkennbar.

Die Evaluierungen in diesem Abschnitt zeigten, dass bei der Verwendung von Optimierungshinweisen in Verbindung mit einer zeitbehafteten Bewertung deutlich bessere Ergebnisse erzielt werden. Außerdem ist das Ergebnis der statischen Problemstellung (WAF-1a) auch in der dynamischen Simulation unter Störungen (WAF-1b) reproduzierbar. Das Verhalten des Feinplanungsverfahrens wird stabiler und vorhersagbarer.

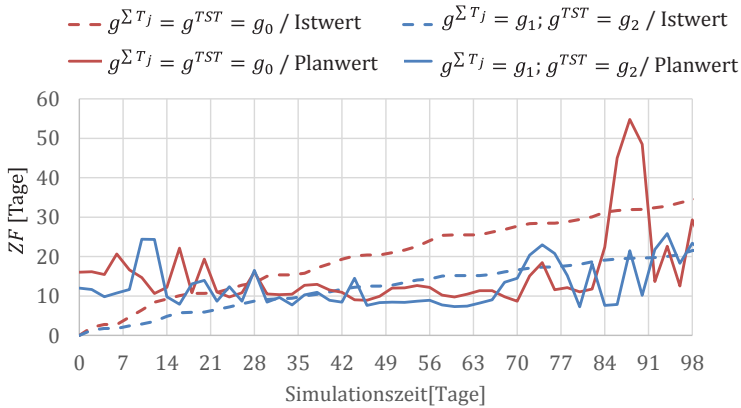


Abbildung 6.25: Zwei repräsentative Simulationsläufe aus WAF-1b-ZF in dynamischen Simulationsläufen mit Störungen

6.6.2 Kollaborationsansätze

In Schritt 2 wird die Kollaboration mit dem Planer untersucht. Grundlage für die nachfolgenden statischen und dynamischen Simulationsläufe ist die in Schritt 1 definierte mutlikriterielle Zielsetzung.

Kollaborationsansätze bei statischer Simulation (WAF-2a)

Zuerst werden die Planübernahmen in den Simulationsläufen von WAF-1a-ZF unter die Lupe genommen. In den Simulationsläufen von GA3 wanderte jede Operation durch Umplanungen im Durch-

schnitt 9,6 Tage innerhalb des Fertigungsplans, wobei fast die Hälfte davon (4,7 Tage) innerhalb der ersten 3.500 Evaluierungen stattfand. Während sich die Anzahl der Übernahmen pro Simulationslauf bei BAT zwischen 30 und 85 belief, lag die Anzahl der Übernahmen trotz einer wesentlich kleineren Evaluierungsanzahl zwischen 180 und 250. Dementsprechend sind die Evaluierungskurven bei WAF-1a-ZF geglättet.

Aufgrund der Auftragsstruktur (kürzere Operationszeiten, weniger Operationen pro Auftrag) werden Verbesserungen von 3-4% ohne große Perturbationen erreicht. Oft trug jede Übernahme ca. 0,05 zu $\vartheta/|O|$ bei. Da die Differenz zwischen dem Ist-Ablauf und dem besten gefundenen Fertigungsplan aus WAF-1a nur $\vartheta/|O| = 1,46$ beträgt, ist eine Reduzierung der Übernahmehäufigkeit ohne Gefährdung der Plangüte zu erwarten.

Analog zu BAT-2 werden die GW für θ und ϑ^b festgelegt. Als akzeptable GW für die Planer des Unternehmens galten die folgenden Werte: für θ ein zulässiger GW von 1 und ein unproblematischer GW von 0,1, für ϑ^b 0,2 bzw. 0,01. Für die beiden zulässigen GW wurde eine Verbesserung von mindestens 2% gesetzt. Dabei wurde eine Verschiebung auf eine andere Maschine innerhalb der ersten Woche des Fertigungsplans als eine Schicht (=0,33 Tage) quantifiziert.

In der ersten Simulation wurde nur die Einhaltung der Planstabilität evaluiert. Dabei werden dieselben Simulationsläufe aus WAF-1a-ZF verwendet. Ein Vergleich auf Abbildung 6.26 zeigt, wie sich der Zielfunktionswert bei der Anwendung der verschiedenen Ansätze auf diesen Simulationsläufen ändert.

Entsprechend der Simulationen bei BAT verhielt sich die Verwendung von ϑ kontraproduktiv, da dadurch mehrere kleine Übernahmen durchgeführt wurden statt einer großen. Aufgrund der festen Einschränkung von θ wurden nach einer gewissen Zeit fast

keine neuen Übernahmen zugelassen. $\theta|2.000$ konnte das Ergebnis der Simulationen bei GA3 aus WAF-1a-ZF wiederholen und dabei $\vartheta/|O|$ auf unter 1/4 reduzieren. Damit wird das Bereitstellen von Zwischenergebnissen aus dem statischen Simulationslauf für einen Planer deutlich komfortabler.

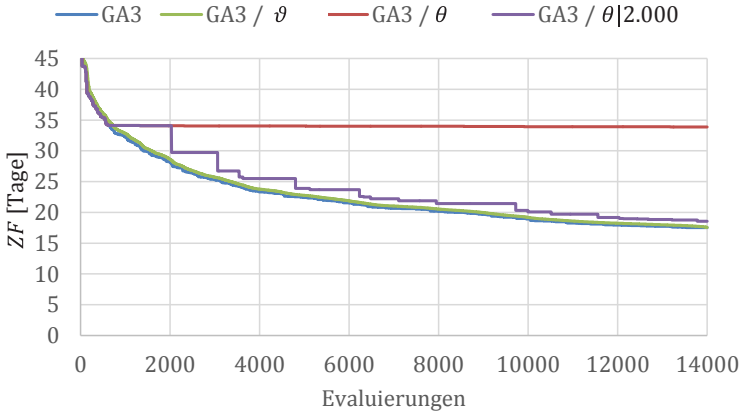


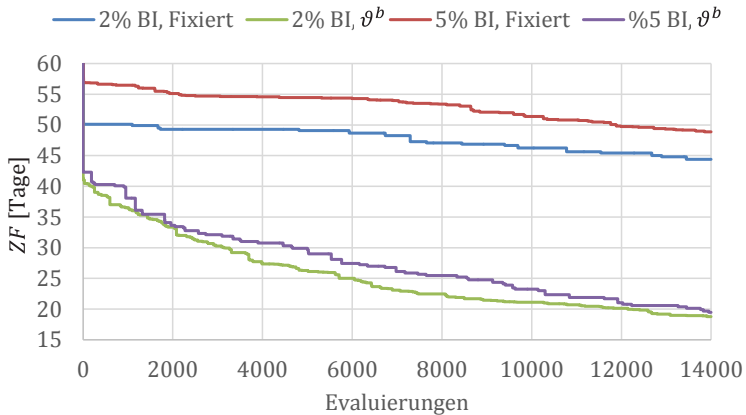
Abbildung 6.26: Einhaltung der Planstabilität in statischen Simulationen (WAF-2a- θ)

Nachfolgend wird die Einhaltung der Benutzerinteraktionen evaluiert. Das Ergebnis aus den Simulationsläufen mit starrer Fixierung und ϑ^b ist in Tabelle 6.14 und Abbildung 6.27 dargestellt. Die starre Fixierung der Operationen bringt sehr viele harte Restriktionen mit sich und schränkt den Lösungsraum so stark ein, dass nur minimale Verbesserungen erreichbar sind. Die fixierten Operationen erschweren die Bildung von rüstopimalen Operationsfolgen, was sich auch negativ auf die Termintreue auswirkt.

Die Verwendung von ϑ^b erlaubt dagegen vor allem die einfache Bildung von rüstopimalen Operationsfolgen. Durchschnittlich war das Endergebnis nur 10% höher als GA3 aus WAF-1a-ZF. Auch hier war die Restriktion $\theta|2.000$ zusammen mit ϑ^b aktiv.

Tabelle 6.14: Ergebnis für WAF-2a- ϑ^b in Tagen

BI	Fixiert	ϑ^b
2%	44,60	18,77
5%	49,34	19,47


Abbildung 6.27: Berücksichtigung der Benutzerinteraktionen in statischen Simulationen (WAF-2a- ϑ^b)

Kollaborationsansätze bei dynamischer Simulation (WAF-2b)

In Teilschritt 2b werden analog zu BAT die Relevanzfunktionen für θ und ϑ^b gemäß der Angaben des Planers geschätzt. Für die Gewichtung der vorigen Übernahmen in $\theta(g^\theta)$ und des Alters der Benutzerinteraktionen in $\vartheta^b(g^b)$ werden jeweils zwei Funktionen untersucht. Sie sind in Abbildung 6.28 dargestellt. Interaktionen bzw. Übernahmen innerhalb des letzten Tages werden dadurch sehr hoch gewichtet.

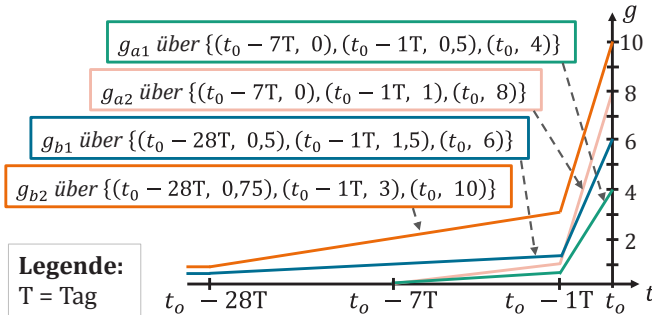


Abbildung 6.28: Untersuchte Relevanzfunktionen in WAF-2b

Für die Einhaltung der Planstabilität (WAF-2b- θ) werden die zwei Kombinationen $(g^\theta = g_{a1}, g^\vartheta = g_2)$ und $(g^\theta = g_{a2}, g^\vartheta = g_3)$ untersucht. Als Grundlage dient hier GA3 mit $g^{\sum T_j} := g_1$ und $g^{TST} := g_2$. Tabelle 6.15 fasst die Ergebnisse zusammen und stellt einen Vergleich mit dem Lauf aus WAF-1b-ZF dar. Beide Kombinationen reduzieren die Perturbation im Fertigungsplan deutlich, zeigen jedoch ihre Unterschiede bei Störungen. Die zweite Kombination von Relevanzfunktion $(g^\theta = g_{a2}, g^\vartheta = g_3)$ ist steiler und schränkt den Optimierungsraum des GAs im naheliegenden Bereich stärker ein. Deshalb wirkt sie sich negativ auf die Reaktionsfähigkeit und das Endergebnis aus.

Tabelle 6.15: Einhaltung der Planstabilität in dynamischen Simulationen (WAF-2b- θ) in Tagen

g	ohne	$g^\theta = g_{a1}$ $g^\vartheta = g_2$	$g^\theta = g_{a2}$ $g^\vartheta = g_3$
störungsfrei	19,25 ($\vartheta/ 0 =562$)	19,80 ($\vartheta/ 0 =111$)	22,29 ($\vartheta/ 0 =81$)
Störungen	21,67 ($\vartheta/ 0 =762$)	22,83 ($\vartheta/ 0 =138$)	26,68 ($\vartheta/ 0 =97$)

Die Verwendung von θ in der ersten Kombination ($g^\theta = g_{a1}$, $g^\vartheta = g_2$) hat zwar wenig Auswirkung auf das Endergebnis, der Planwert ist aber aufgrund der reduzierten Übernahmehäufigkeit wenig stabil und zum Teil deutlich höher. Abbildung 6.29 zeigt den Vergleich eines Simulationslaufs mit dem blauen Simulationslauf aus Abbildung 6.25, hier rot dargestellt. Der überwiegend schlechtere Planwert bei Verwendung von θ nach auftretenden Störungen ist deutlich zu erkennen. Der Istwert am Ende ist jedoch nur geringfügig schlechter.

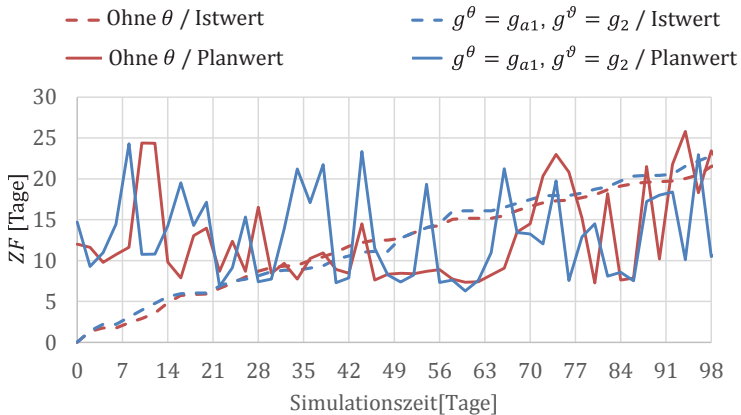


Abbildung 6.29: Zwei repräsentative Simulationsläufe aus WAF-2b- θ in dynamischen Simulationsläufen mit Störungen

Im nächsten Schritt wird die Einhaltung von Benutzerinteraktionen in einer dynamischen Simulation evaluiert (WAF-2b- ϑ^b). Hierzu wurden auch zwei Kombinationen von Relevanzfunktionen untersucht (s. Tabelle 6.16). Wie auch in den statischen Simulationsläufen hielt die Fixierung der Operationen die Benutzerwünsche zwar sehr genau ein, schränkte den Lösungsraum aber spürbar ein. Besonders bei simulierten Störungen reduziert sich die Reaktionsfähigkeit des GAS deutlich. Die Einhaltung der Benutzerinteraktionen mittels ϑ^b mit der ersten Kombination ($g^b = g_{b1}$, $g^{\vartheta^b} = g_2$) erzielte durchschnittliche Abweichungen

von ca. 0,9 Tagen. Die zweite Kombination ($g^b = g_{b2}$, $g^{\vartheta^b} = g_3$) konnte diesen Wert in allen Simulationsläufen auf unter 0,6 bei spürbarer Verschlechterung des Endergebnisses senken.

Die Benutzerinteraktionen werden bis auf akzeptable Abweichungen eingehalten und das Feinplanungsverfahren hat sowohl einen größeren Optimierungsraum als auch eine höhere Reaktionsgeschwindigkeit, wie ein Vergleich von zwei Simulationsläufen in Abbildung 6.30 zeigt. Harte Fixierungen dagegen schränken den Optimierungsraum des Feinplanungsverfahrens ein. Dies ist besonders bei der Simulation von Störungen ersichtlich, bei der eine kurzzeitige Nichteinhaltung der Benutzerinteraktionen von Vorteil wäre.

Tabelle 6.16: Berücksichtigung der Benutzerinteraktionen in dynamischen Simulationen (WAF-2b- ϑ^b) in Tagen

g	<i>fixiert</i>	$g^b = g_{b1}$ $g^{\vartheta^b} = g_2$	$g^b = g_{b2}$ $g^{\vartheta^b} = g_3$
<i>störungsfrei,</i> <i>2% BI</i>	54,6 ($\vartheta^b / 0 = 0,05$)	21,9 ($\vartheta^b / 0 = 0,66$)	28,2 ($\vartheta^b / 0 = 0,42$)
<i>Störungen,</i> <i>2%, BI</i>	66,5 ($\vartheta^b / 0 = 0,22$)	24,2 ($\vartheta^b / 0 = 0,85$)	36,6 ($\vartheta^b / 0 = 0,53$)
<i>störungsfrei,</i> <i>5%, BI</i>	62,7 ($\vartheta^b / 0 = 0,07$)	23,9 ($\vartheta^b / 0 = 0,78$)	33,9 ($\vartheta^b / 0 = 0,44$)
<i>Störungen,</i> <i>5% BI</i>	71,3 ($\vartheta^b / 0 = 0,29$)	25,6 ($\vartheta^b / 0 = 0,92$)	41,1 ($\vartheta^b / 0 = 0,58$)

Wie aus den Ergebnissen ersichtlich ist, bringt die Feinplanungplattform mit den Einstellungen ($g^{\Sigma T_j} = g_1$, $g^{TST} = g_2$, $g^\theta = g_{a1}$, $g^b = g_{b1}$, $g^\vartheta = g^{\vartheta^b} = g_2$) sowohl eine Verbesserung der Planungsgüte als auch der Planstabilität und der Einhaltung der Benutzerinteraktionen mit sich. Bestehende Konzepte aus

dem Stand der Technik hatten sich hier als ungeeignet für eine kontinuierliche Feinplanung in einer dynamischen Umgebung erwiesen.

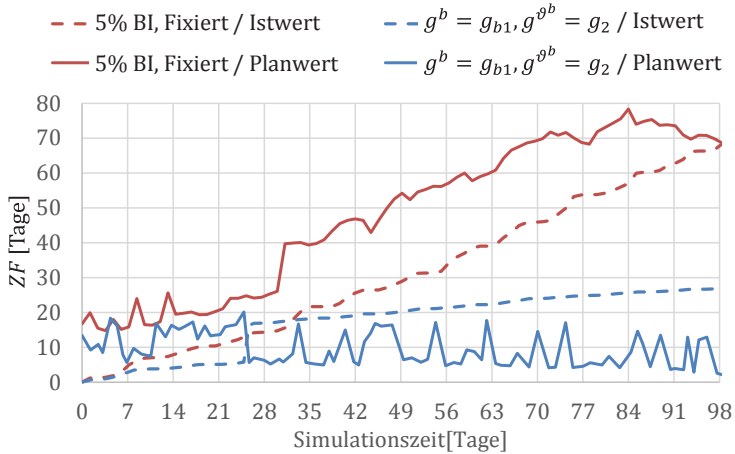


Abbildung 6.30: Zwei repräsentative Simulationsläufe aus WAF-2b- g^b -5%BI in dynamischen Simulationsläufen mit Störungen

6.7 Abschließende Bewertung

Die präsentierten Simulationsergebnisse sind das Resultat aus ca. 900 Stunden Rechenzeit, die sich zu gleichen Anteilen aus den Anwendungsfällen BAT und WAF zusammensetzt. Mehr als Dreiviertel davon sind zeitintensive dynamische Simulationsläufe. Bei WAF wurden aufgrund der multikriteriellen Zielsetzung erheblich mehr Simulationsläufe zur Einstellung der Zielfunktion durchgeführt.

Die Zeitabschnitte der beiden Anwendungsfälle umfassen jeweils ca. ein Quartal, liegen in verifizierter und plausibilisierter Form vor und sind für realitätsnahe Simulationsuntersuchungen repräsentativ. Trotz unterschiedlicher Technologien, Organisationen

und Optimierungsziele bei den Anwendungsfällen BAT und WAF konnten die konzipierten Ansätze der Feinplanungsplattform in der Simulation eine signifikante Verbesserung im zweistelligen Prozentbereich erzielen. Manche der Ansätze, wie die Verfahrensunterstützung durch Optimierungsstellen, konnten ohne fallspezifische Konfiguration eingesetzt werden und erzielten eine Verbesserung von ca. 10% bei BAT und bis zu 40% bei WAF. Die parallele Ausführung von Verfahrensinstanzen auf disjunkte Teile des Fertigungsplans konnte bei WAF ihre Stärke zeigen. Dort wurde eine 2-fache Beschleunigung gegenüber dem klassischen Ansatz gemessen. Entscheidend für das Endergebnis war die Größe des Markierungsbereichs.

Die Verwendung der Relevanzkurven erfordert eine Einstellung mithilfe von dynamischen Simulationen. Die Feinplanungsplattform konnte beim simulierten Fertigungsablauf mit Störungen durch die Relevanzfunktionen mit 10-40% ein deutlich besseres und stabileres Endergebnis erreichen. Zudem erhöhte sich die Reaktionsfähigkeit bei Störungen deutlich und wiederholte das erreichte Ergebnis aus den statischen Simulationsläufen fast.

Die Ansätze für Kollaboration mit dem Planer wurden nach der Interpretation ihrer Wünsche konfiguriert. Sie zeigten eine um 90% erhöhte Planstabilität bei akzeptabler Verschlechterung des Endergebnisses im einstelligen Prozentbereich. Die entworfenen Ansätze konnten ihre Stärke bei der Einhaltung der Benutzerinteraktionen zeigen. Dort wurde deutlich, dass gegenüber der klassischen starren Fixierung von Operationen je nach Einhaltungsgrad eine Verbesserung von 30-70% erzielbar ist.

Da die Auftragslage bei Unternehmen aus der Automobil- und Elektronikindustrie sehr schwankend ist, ist eine Überprüfung der Einstellungen der Feinplanungsplattform mit weiteren Fertigungszeiträumen vor einer Inbetriebnahme ratsam. Bei WAF bestand die Möglichkeit, die abgeleitete Einstellung der Feinpla-

nungsplattform auf weitere Daten aus zwei früheren Fertigungszeiträumen zu überprüfen. Dort hat sich die Feinplanungsplattform als sehr vorteilhaft erwiesen [DiBa11, DiBS11, DiBa12]. Die Verbesserung der TST (bei gleichbleibender $\sum T_j$) gegenüber dem Ist-Ablauf lag bei niedrigem Auftragsvolumen bei ca. 28%, bei hohem Auftragsvolumen sogar bei ca. 79%. Die $\sum T_j$ konnte in diesen beiden Fällen auf unter einen Tag reduziert werden. Von den Ansätzen zur Performanzsteigerung ist deshalb zu erwarten, dass die simulierten Verbesserungen bei ähnlicher Auftragslage auch in der realen Fertigung erzielbar werden.

In den drei von WAF bereitgestellten Zeiträumen wurde neben der Verbesserung gegenüber dem Ist-Ablauf eine Korrelation zwischen Simulationsergebnis und unterer Schranke festgestellt. Die $\sum T_j$ konnte bis auf $LB_{\sum T_j} + 2$ (Tage) reduziert werden. Bei gleichbleibender $\sum T_j$ belief sich die TST auf $2 * LB_{TST}$. Bei BAT konnte die gleiche Korrelation zwischen $\emptyset DLZ$ und $LB_{\emptyset DLZ}$ festgestellt werden wie bei der TST des zweiten Unternehmens. Im Endeffekt zeigt diese Korrelation, dass bei ähnlicher Auftragslast die Erreichung der Optimierungsziele mithilfe der Feinplanungsplattform vorhersagbar ist.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung der Beiträge und Ergebnisse

Diese Arbeit präsentiert innovative Ansätze für permanent laufende Feinplanungsverfahren. Sie verschaffen dem Verhalten der Feinplanungsverfahren mehr Intelligenz und Kollaborationsfähigkeit bei gleichzeitig steigender Optimierungsqualität. Die Ansätze wurden in einer Feinplanungsplattform zusammengefasst und mit realen Daten zweier Unternehmen untersucht. Die durchgeführten statischen und dynamischen Simulationsläufe belegen die Stärken der entwickelten Ansätze: Sehr hohe Performanz, Reaktionsfähigkeit, Planstabilität und gute Einhaltung von Benutzerinteraktionen. Diese Ergebnisse konnten mittels durchdringender Analyse des von den Planern im Industrie 4.0 Kontext gewünschten Verhaltens und konsequenter Umsetzung in den entwickelten Ansätzen erzielt werden.

Die Feinplanungsplattform integriert stark aufgewertete Varianten der klassischen Bewertungskriterien und Vergleichsfunktionen für Fertigungspläne. Der Fertigungsplan wird sowohl als Ganzes als auch in einer mehrstufigen Fragmentierung betrachtet. Die sehr wirksamen neuen Bewertungskriterien geben Auskunft über Optimierungspotential und -bedarf im gesamten Fertigungsplan und in seinen einzelnen Fragmenten. Dabei berücksichtigen sie den verfügbaren Optimierungsraum und die Relevanz zum aktuellen Zeitpunkt.

Die Feinplanungsplattform ist durch einen neuartigen Ansatz in der Lage Feinplanungsverfahren von außen zu steuern, in dem sie auf Fragmente mit Optimierungsbedarf und -potential hinweist. Diese Fragmente können selbst mehreren unabhängigen parallel laufenden Verfahrensinstanzen zur exklusiven Optimierung zugeteilt werden, um die Optimierungsgeschwindigkeit zu steigern. In einer laufenden Fertigung wird das Feinplanungsverfahren gezielt auf Fragmente gelenkt, deren Optimierung zum aktuellen Zeitpunkt dringender ist oder die ein sehr hohes Optimierungspotential aufweisen.

Um Änderungsvorschläge aus einem Feinplanungsverfahren in den aktuell auszuführenden Fertigungsplan zu übernehmen, wird die voraussichtliche Akzeptanz der Planer durch zwei neuartige Kennzahlen quantifiziert. Die erste Kennzahl spiegelt die Einhaltung der Benutzerinteraktionen wieder. Die zweite Kennzahl misst die kumulierte Planstabilität durch Summierung der Perturbationen im auszuführenden Fertigungsplan. So kann die Feinplanungsplattform abhängig von den Werten der beiden Kennzahlen und der potentiellen Verbesserung im Fertigungsplan qualifiziert über eine Übernahme der Änderungsvorschläge entscheiden.

In einer dynamischen Simulationsumgebung wurden die entwickelten Ansätze mit zwei Anwendungsfällen aus der Industrie und beim Einsatz eines modifizierten genetischen Algorithmus untersucht. Die auf Performanzsteigerung ausgelegten Ansätze konnten in einer statischen Simulation Verbesserungen von 10-40% erreichen. In einer dynamischen Simulation konnten sie dieses Ergebnis sogar unter Störungen mit geringen Schwankungen wiederholen, während klassische Ansätze aus dem Stand der Technik ein um 10-40% schlechteres und stark schwankendes Ergebnis produzierten.

Die auf Kollaboration mit den Planern ausgelegten Ansätze konnten die Planstabilität sowohl in statischen als auch in dynamischen Simulationsläufen deutlich erhöhen und dabei die Lösungsgüte nur minimal beeinträchtigen. Ihre Stärke zeigen diese Ansätze bei getätigten Benutzerinteraktionen, wo eine Verbesserung von 30-70% gegenüber der klassischen Operationsfixierung erzielt werden konnte.

7.2 Ausblick

Die Feinplanungsplattform bringt signifikante Verbesserungen für Feinplanungsverfahren in der Fertigungssteuerung und öffnet neue Horizonte. Die erarbeiteten Ansätze sind zum Teil auf eine Konfigurierung durch Simulationsläufe angewiesen. Um den breiteren Einsatz der entwickelten Ansätze zu erleichtern, sollte die Konfigurierbarkeit der Ansätze selbst zum Forschungsgegenstand werden. Anstatt die Feinplanungsplattform in Simulationsläufen abgekoppelt von der Fertigung zu konfigurieren, kann die Konfiguration im laufenden Betrieb optimal und automatisiert an die Wünsche und Verhaltensweisen der Benutzer angepasst werden. Die Feinplanungsplattform würde dadurch konfigurationsfrei und sofort einsatzbereit.

Die entwickelten Ansätze zeigen in den Simulationsläufen sehr gute Ergebnisse mit dem eingesetzten genetischen Algorithmus. Weitere iterative Feinplanungsverfahren können an die Feinplanungsplattform angedockt werden. Damit wäre es möglich, abhängig vom konkreten Optimierungsproblem und der Zielfunktion das am besten geeignete Feinplanungsverfahren auszuwählen. Für punktuelle Verbesserungen kann ohne explizite Konfiguration automatisch beispielsweise ein lokales Suchverfahren ausgewählt werden, für Termintreue ein genetischer Algorithmus und für die Optimierung von Rüstzeiten etwa ein Ameisenalgorithmus. So

kann die Feinplanungsplattform aus den verschiedenen Verfahren das Beste herauschlagen.

Die Ansätze wurden gezielt für die Fertigungsfeinplanung entwickelt, können aber auf verwandte Planungsprobleme mit hohen Reaktionsanforderungen und ausgeprägter Mensch-Maschine-Interaktion übertragen werden. So können beispielsweise die Transport- oder Projektplanung von den Fragmentierungstechniken und den Kollaborationsansätzen bedeutend profitieren.

Literaturverzeichnis

- [AdBZ88] Adams, Joseph ; Balas, Egon ; Zawack, Daniel: The shifting bottleneck procedure for job shop scheduling. In: *Manage. Sci.* Bd. 34 (1988), Nr. 3, S. 391–401
- [AdKa91] Adelsberger, Heimo H ; Kanet, John J: The Leitstand-a new tool for computer integrated manufacturing. In: *Production and Inventory Management Journal* Bd. 32 (1991), Nr. 1, S. 43–48
- [AdZA10] Adibi, M. A. ; Zandieh, M. ; Amiri, M.: Multi-objective scheduling of dynamic job shop using variable neighborhood search. In: *Expert Systems with Applications* Bd. 37 (2010), Nr. 1, S. 282–287
- [AiBR03] Aiex, Renata M ; Binato, Silvio ; Resende, Mauricio GC: Parallel GRASP with path-relinking for job shop scheduling. In: *Parallel Computing* Bd. 29 (2003), Nr. 4, S. 393–430
- [AlAz03] Alagöz, Oğuzhan ; Azizoğlu, Meral: Rescheduling of identical parallel machines under machine eligibility constraints. In: *European Journal of Operational Research* Bd. 149 (2003), Nr. 3, S. 523–532
- [Amic91] AMICE Consortium, AMICE: Open System Architecture, CIMOSA, AD 1.0, Architecture Description. In: *ESPRIT Consortium AMICE, Brussels, Belgium* (1991)
- [ANCK08] Allahverdi, Ali ; Ng, C.T. ; Cheng, T.C.E. ; Kovalyov, Mikhail Y.: A survey of scheduling problems with setup times or costs. In: *European Journal of Operational Research* Bd. 187 (2008), Nr. 3, S. 985–1032
- [Ansi10] ANSI/ISA-95.00.01, ANSI: *Enterprise-Control System Integration - Part 1: Models and Terminology (IEC 62264-1 Mod)* : Technical report, ISA, The Instrumentation, Systems, and Automation Society, 2010

- [Ansi13a] ANSI/ISA-95.00.03: *Enterprise-Control System Integration - Part 3: Activity Models of Manufacturing Operations Management (IEC 62264-3 Mod)* : Technical report, ISA, The Instrumentation, Systems, and Automation Society, 2013
- [Ansi13b] ANSI/ISA-95.00.05: *Enterprise-Control System Integration - Part 5: Business-to-Manufacturing Transactions* : Technical report, ISA, The Instrumentation, Systems, and Automation Society, 2013
- [Awf85] AwF: *Einsatz in der Produktion. CIM-Computer Integrated Manufacturing. Begriffe, Definitionen, Funktionszuordnung*. Eschborn : Ausschuss für wirtschaftliche Fertigung, 1985
- [AWWB09] Affenzeller, Michael ; Winkler, Stephan ; Wagner, Stefan ; Beham, Andreas: *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Auflage: 1. Aufl. Boca Raton : Chapman & Hall, 2009 — ISBN 9781584886297
- [AxBK08] Axsen, Jonn ; Burke, Andy ; Kurani, Kenneth S: Batteries for plug-in hybrid electric Vehicles (PHEVs): goals and the state of technology circa 2008. In: *Institute of Transportation Studies* (2008)
- [AyKU03] Aytug, H. ; Kempf, K. ; Uzsoy, R.: Measures of subproblem criticality in decomposition algorithms for shop scheduling. In: *International Journal of Production Research* Bd. 41 (2003), Nr. 5, S. 865–882
- [BaDi08] Baumann, Michael ; Dimitrov, Todor: Hierarchisierte Modelle in Advanced Planning and Scheduling (APS). In: *Automatisierungstechnik : AT* Bd. 56 (2008), Nr. Nr.2, S. 90–97
- [BaDi11a] Baumann, Michael ; Dimitrov, Todor: Fertigungsmanagementsysteme werden zunehmend intelligenter. In: *Maschinenmarkt. MM, das Industriemagazin* (2011), Nr. Nr.12

- [BaDi11b] Baumann, Michael ; Dimitrov, Todor: Simulation zur Auswahl und Parametrierung einer Produktionssteuerung: Risikolos verstecktes Optimierungspotenzial erschließen. In: *Productivity management* Bd. 16 (2011), Nr. Nr.1, S. 23-25
- [BaDM06] Baumann, Michael ; Dimitrov, Todor ; Martinek, Robert: Komplexität in der Produktion beherrschen - Adaptivität und Interoperabilität in Produktionsplanung und -steuerung mit dem ISA-95 Standard. In: *PPS Management* Bd. 11 (2006), Nr. Nr.3, S. 29-32
- [Baek99] Baek, Dong H.: A visualized human-computer interactive approach to job shop scheduling. In: *International Journal of Computer Integrated Manufacturing* Bd. 12 (1999), Nr. 1, S. 75-83
- [Bake77] Baker, Kenneth R: An experimental study of the effectiveness of rolling schedules in production planning. In: *Decision Sciences* Bd. 8 (1977), Nr. 1, S. 19-27
- [Bake98] Baker, Albert D: A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: dispatching, scheduling, and pull. In: *Journal of Manufacturing Systems* Bd. 17 (1998), Nr. 4, S. 297-320
- [Bala69] Balas, Egon: Machine sequencing via disjunctive graphs: an implicit enumeration algorithm. In: *Operations research* Bd. 17 (1969), Nr. 6, S. 941-957
- [BaSR10] Barták, Roman ; Salido, Miguel A ; Rossi, Francesca: Constraint satisfaction techniques in planning and scheduling. In: *Journal of Intelligent Manufacturing* Bd. 21 (2010), Nr. 1, S. 5-15
- [BDMN99] Brucker, Peter ; Drexl, Andreas ; Möhring, Rolf ; Neumann, Klaus ; Pesch, Erwin: Resource-constrained project scheduling: Notation, classification, models, and methods. In: *European Journal of Operational Research* Bd. 112 (1999), Nr. 1, S. 3-41

- [BeGe12] Behnke, Dennis ; Geiger, Martin Josef: Test instances for the flexible job shop scheduling problem with work centers (2012)
- [BEPT04] Blażewicz, Jacek ; Ecker, Klaus ; Pesch, Erwin ; Trystram, Denis: Scheduling in Computer and Manufacturing Systems. In: Blażewicz, J. ; Ecker, K. ; Pesch, E. ; Trystram, D. (Hrsg.): *Scheduling in Computer and Manufacturing Systems, Dagstuhl Seminar Proceedings*. Dagstuhl, Germany : Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2004
- [BHLR02] Binato, S ; Hery, WJ ; Loewenstern, DM ; Resende, MGC: A GRASP for job shop scheduling. In: *Essays and surveys in metaheuristics* : Springer, 2002, S. 59–79
- [BiMa99] Bierwirth, Christian ; Mattfeld, Dirk C.: Production scheduling and rescheduling with genetic algorithms. In: *Evol. Comput.* Bd. 7 (1999), Nr. 1, S. 1–17
- [BKOS00] De Berg, Mark ; Van Kreveld, Marc ; Overmars, Mark ; Schwarzkopf, Otfried Cheong: *Computational geometry* : Springer, 2000
- [BIPH82] Blackstone, John H ; Phillips, Don T ; Hogg, Gary L: A state-of-the-art survey of dispatching rules for manufacturing job shop operations. In: *The International Journal of Production Research* Bd. 20 (1982), Nr. 1, S. 27–45
- [BIPS00] Blażewicz, Jacek ; Pesch, Erwin ; Sterna, Malgorzata: The disjunctive graph machine representation of the job shop scheduling problem. In: *European Journal of Operational Research* Bd. 127 (2000), Nr. 2, S. 317–331
- [BIRo03] Blum, Christian ; Roli, Andrea: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. In: *ACM Computing Surveys (CSUR)* Bd. 35 (2003), Nr. 3, S. 268–308

- [BoJV06] Bowling, Michael ; Jensen, Rune ; Veloso, Manuela: Multi-Agent Planning in the Presence of Multiple Goals. In: Wezel, W. van ; Jorna, R. ; Meystel, A. (Hrsg.): *Planning in Intelligent Systems* : John Wiley & Sons, Inc., 2006 — ISBN 9780471781264, S. 301–325
- [Bože12] Božek, Andrzej: Custom scheduling algorithms in real-world applications. In: *Research in Logistics & Production* Bd. 2 (2012), S. 233–245
- [Bran67] Brankamp, Klaus: *Ein Terminplanungssystem für Unternehmen der Einzel- und Serienfertigung*, 1967
- [Bran93] Brandimarte, Paolo: Routing and scheduling in a flexible job shop by tabu search. In: *Annals of Operations Research* Bd. 41 (1993), Nr. 3, S. 157–183
- [BrGW11] Bracht, Uwe ; Geckler, Dieter ; Wenzel, Sigrid: Digitale Fabrik. In: *Methoden und Praxisbeispiele, Berlin, Heidelberg* (2011)
- [BrKn03] Brucker, Peter ; Knust, Sigrid: Lower bounds for resource-constrained project scheduling problems. In: *European Journal of Operational Research, Sequencing and Scheduling*. Bd. 149 (2003), Nr. 2, S. 302–313
- [BrKn09] Brucker, Peter ; Knust, Sigrid: *Complexity results of scheduling problems*. URL <http://www.informatik.uni-ornsnabrueck.de/knust/class/>. - abgerufen am 2015-03-15
- [Bruc07] Brucker, Peter: *Scheduling Algorithms*. Auflage: 5th ed. 2007. Aufl. Berlin ; New York : Springer, 2007 — ISBN 9783540695158
- [BrWA05] Braune, Roland ; Wagner, Stefan ; Affenzeller, Michael: On the analysis of crossover schemes for genetic algorithms applied to the job shop scheduling problem. In: *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*, 2005, S. 236–241

- [BuKe13] Burke, E. K. ; Kendall, G. (Hrsg.): *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Auflage: 2nd ed. 2014. Aufl. New York : Springer, 2013 — ISBN 9781461469391
- [BYVB09] Buyya, Rajkumar ; Yeo, Chee Shin ; Venugopal, Srikumar ; Broberg, James ; Brandic, Ivona: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. In: *Future Generation Computer Systems* Bd. 25 (2009), Nr. 6, S. 599–616
- [Cant98] Cantú-Paz, Erick: A survey of parallel genetic algorithms. In: *Calculateurs paralleles, reseaux et systems repartis* Bd. 10 (1998), Nr. 2, S. 141–171
- [CaPi89] Carlier, Jacques ; Pinson, Eric: An algorithm for solving the job-shop problem. In: *Management science* Bd. 35 (1989), Nr. 2, S. 164–176
- [Casa01] Casanova, Henri: Simgrid: A toolkit for the simulation of application scheduling. In: *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on* : IEEE, 2001, S. 430–437
- [CBCC04] Ai-Chang, Mitchell ; Bresina, John ; Charest, Len ; Chase, Adam ; Hsu, JC-J ; Jonsson, Ari ; Kanefsky, Bob ; Morris, Paul ; u. a.: Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission. In: *Intelligent Systems, IEEE* Bd. 19 (2004), Nr. 1, S. 8–12
- [CCFC08] Chang, Pei-Chann ; Chen, Shih-Hsin ; Fan, Chin-Yuan ; Chan, Chien-Lung: Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems. In: *Applied Mathematics and Computation, Special Issue on Advanced Intelligent Computing Theory and Methodology in Applied Mathematics and Computation*. Bd. 205 (2008), Nr. 2, S. 550–561
- [CDMT94] Colorni, Alberto ; Dorigo, Marco ; Maniezzo, Vittorio ; Trubian, Marco: Ant system for job-shop scheduling. In: *Belgian Journal of Operations Research, Statistics and Computer Science* Bd. 34 (1994), Nr. 1, S. 39–53

- [ChGL83] Chan, Tony F ; Golub, Gene H ; LeVeque, Randall J: Algorithms for computing the sample variance: Analysis and recommendations. In: *The American Statistician* Bd. 37 (1983), Nr. 3, S. 242–247
- [ChGT96] Cheng, Runwei ; Gen, Mitsuo ; Tsujimura, Yasuhiro: A tutorial survey of job-shop scheduling problems using genetic algorithms—I. Representation. In: *Computers & Industrial Engineering* Bd. 30 (1996), Nr. 4, S. 983–997
- [ChGT99] Cheng, Runwei ; Gen, Mitsuo ; Tsujimura, Yasuhiro: A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies. In: *Computers & Industrial Engineering* Bd. 36 (1999), Nr. 2, S. 343–364
- [ChHT09] Choudhary, Alok Kumar ; Harding, Jenny A ; Tiwari, Manoj Kumar: Data mining in manufacturing: a review based on the kind of knowledge. In: *Journal of Intelligent Manufacturing* Bd. 20 (2009), Nr. 5, S. 501–521
- [ChJi07] Chen, K.J. ; Ji, P.: A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with a frozen interval. In: *Expert Systems with Applications* Bd. 33 (2007), Nr. 4, S. 1004–1010
- [ChSm97] Cheng, Cheng-Chung ; Smith, Stephen F: Applying constraint satisfaction techniques to job shop scheduling. In: *Annals of Operations Research* Bd. 70 (1997), S. 327–357
- [ChSu01] Chrysolouris, George ; Subramaniam, Velusamy: Dynamic scheduling of manufacturing job shops using genetic algorithms. In: *Journal of Intelligent Manufacturing* Bd. 12 (2001), Nr. 3, S. 281–293
- [ChUz92] Chruch, Laura K. ; Uzsoy, R.M.: Analysis of periodic and event-driven rescheduling policies in dynamic shops. In: *International Journal of Computer Integrated Manufacturing* Bd. 5 (1992), Nr. 3, S. 153–163

- [CoRC03] Coello, Carlos A Coello ; Rivera, Daniel Cortes ; Cortes, Nareli Cruz: Use of an artificial immune system for job shop scheduling. In: *Artificial Immune Systems* : Springer, 2003, S. 1–10
- [CPPR04] Carlier, Jacques ; Péridy, Laurent ; Pinson, Éric ; Rivreau, David: Elimination rules for job-shop scheduling problem: overview and extensions. In: Leung, J. Y.-T. (Hrsg.): *Handbook of scheduling: algorithms, models and performance analysis*. Boca Raton : Chapman & Hall/CRC, 2004 — ISBN 1584883979 9781584883975
- [CuFe09] Curry, Guy L ; Feldman, Richard M: *Manufacturing systems modeling and analysis*. Berlin : Springer, 2009 — ISBN 9783540887638 3540887636
- [DaDe97] Das, I. ; Dennis, J. E.: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. In: *Structural optimization* Bd. 14 (1997), Nr. 1, S. 63–69
- [DeHe92] Demeulemeester, Erik ; Herroelen, Willy: A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem. In: *Management Science* Bd. 38 (1992), Nr. 12, S. 1803–1818
- [DeKü13] Demir, Yunus ; Kürşat İşleyen, S.: Evaluation of mathematical models for flexible job-shop scheduling problems. In: *Applied Mathematical Modelling* Bd. 37 (2013), Nr. 3, S. 977–988
- [DiBa09] Dimitrov, Todor ; Baumann, Michael: Simulationsumgebung zur Analyse von Algorithmen für dynamische Scheduling-Probleme im Produktionsumfeld. In: *Informatik 2009. Im Focus das Leben*, 2009 — ISBN 9783885792482, S. 471, 15
- [DiBa11] Dimitrov, Todor ; Baumann, Michael: Genetic algorithm with genetic engineering technology for multi-objective dynamic job shop scheduling problems. In: *GECCO 2011, 13th Annual Conference Companion on Genetic and Evolutionary Computation. Proceedings*, 2011 — ISBN 9781450306904, S. 833–834

- [DiBa12] Dimitrov, Todor ; Baumann, Michael: Criteria evaluation considering the current time used by a genetic scheduling algorithm for changeover and tardiness reduction. In: *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 2012, S. 425–430
- [DiBS10] Dimitrov, Todor ; Baumann, Michael ; Schenk, Manfred: Master data extraction and adaptation based on collected production data in manufacturing execution systems. In: *CIRP ICME 2010, 7th CIRP International Conference on Intelligent Computation in Manufacturing Engineering. Innovative and Cognitive Production Technology and Systems*, 2010
- [DiBS11] Dimitrov, Todor ; Baumann, Michael ; Schenk, Manfred: Scheduling of large dynamic job shop problems by concurrent non-blocking threads. In: *5th Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA 2011*, 2011
- [Dine08] DIN/EN 62264-1: *Integration von Unternehmensführungs- und Leitsystemen. Teil 1: Modelle und Terminologie* : Deutsches Institut für Normung (DIN), 2008
- [DoCG99] Dorigo, Marco ; Caro, Gianni ; Gambardella, Luca: Ant algorithms for discrete optimization. In: *Artificial life* Bd. 5 (1999), Nr. 2, S. 137–172
- [DoGi94] Dorn, Jürgen ; Girsch, Mario: Genetic operators based on constraint repair. In: *Proceedings of the ECAI*. Bd. 94 : Citeseer, 1994
- [DoGV98] Dorn, Jürgen ; Girsch, Mario ; Vidakis, Nikos: DÉ]À VU—A reusable framework for the construction of intelligent interactive schedulers. In: *Advances in Production Management Systems* : Springer, 1998, S. 467–478
- [DoKa01] Do, Minh Binh ; Kambhampati, Subbarao: Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP. In: *Artificial Intelligence* Bd. 132 (2001), Nr. 2, S. 151–182

- [DoKT95] Dorn, Jürgen ; Kerr, Roger ; Thalhammer, Gabi: Reactive scheduling: Improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning. In: *International journal of human-computer studies* Bd. 42 (1995), Nr. 6, S. 687–704
- [DoPe95] Dorndorf, Ulrich ; Pesch, Erwin: Evolution based learning in a job shop scheduling environment. In: *Computers & Operations Research* Bd. 22 (1995), Nr. 1, S. 25–40
- [DrEK09] Drótos, Márton ; Erdős, Gábor ; Kis, Tamás: Computing lower and upper bounds for a large-scale industrial job shop scheduling problem. In: *European Journal of Operational Research* Bd. 197 (2009), Nr. 1, S. 296–306
- [DVZC92] Doumeingts, G ; Vallespir, B ; Zanettin, M ; Chen, D: GRAI-GIM Integrated Methodology, A Methodology for Designing CIM Systems, Version 1.0. In: *LAP/GRAI, University Bordeaux I, France* (1992)
- [Enge68] Engelbart, Douglas C: The mother of all demos (1968)
- [EvAn12] Evans, Peter C ; Annunziata, Marco: Industrial internet: Pushing the boundaries of minds and machines. In: *White Paper available at <http://www.ge.com>* (2012)
- [EvKe00] Eversheim, Walter ; Kees, Alexandra: *Wörterbuch der PPS: Deutsch - Englisch/Englisch - Deutsch = Dictionary of PPC*. Berlin [u.a.] : Springer, 2000 — ISBN 3540661395 9783540661399
- [FaFG94] Fandel, Günter ; Francois, Peter ; Gublitz, Klaus-Martin: *PPS-Systeme: Grundlagen, Methoden, Software, Marktanalyse*. Berlin u.a. : Springer, 1994 — ISBN 9783540583936
- [FaMJ07] Fattahi, Parviz ; Mehrabad, Mohammad Saidi ; Jolai, Fariborz: Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. In: *Journal of Intelligent Manufacturing* Bd. 18 (2007), Nr. 3, S. 331–342

- [FaXi97] Fang, Jian ; Xi, Yugeng: A rolling horizon job shop rescheduling strategy in the dynamic environment. In: *The International Journal of Advanced Manufacturing Technology* Bd. 13 (1997), Nr. 3, S. 227–232
- [Finc09] Finch, Tony: Incremental calculation of weighted mean and variance. In: *University of Cambridge* (2009)
- [FoSm84] Fox, Mark S ; Smith, Stephen F: ISIS—a knowledge-based system for factory scheduling. In: *Expert systems* Bd. 1 (1984), Nr. 1, S. 25–49
- [Fren82] French, Simon: *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-shop* : Ellis Horwood Ltd, Publisher, 1982 — ISBN 0853122997
- [FrNS01] Franck, Birger ; Neumann, Klaus ; Schwindt, Christoph: Truncated branch-and-bound, schedule-construction, and schedule-improvement procedures for resource-constrained project scheduling. In: *OR-Spektrum* Bd. 23 (2001), Nr. 3, S. 297–324
- [Fröh06] Fröhling, Magnus: *Zur taktisch-operativen Planung stoffstrombasierter Produktionssysteme: Dargestellt an Beispielen aus der stoffumwandelnden Industrie*. Auflage: 2006. Aufl. Wiesbaden : Deutscher Universitätsverlag, 2006 — ISBN 9783835004498
- [FrRu10] Framinan, Jose M. ; Ruiz, Rubén: Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. In: *European Journal of Operational Research* Bd. 205 (2010), Nr. 2, S. 237–246
- [GaFW10] Gasser, Roland ; Fischer, Katrin ; Wäfler, Toni: Decision Making in Planning and Scheduling: A Field Study of Planning Behaviour in Manufacturing. In: Fransoo, J. C. ; Waefler, T. ; Wilson, J. R. (Hrsg.): *Behavioral Operations in Planning and Scheduling* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-13381-7, 978-3-642-13382-4, S. 11–30

- [Ga]S76] Garey, Michael R ; Johnson, David S ; Sethi, Ravi: The complexity of flowshop and jobshop scheduling. In: *Mathematics of operations research* Bd. 1 (1976), Nr. 2, S. 117–129
- [Gaze06] Gazendam, Henk W.M.: Coordination Mechanisms in Multi-Actor Systems. In: Wezel, W. van ; Jorna, R. ; Meystel, A. (Hrsg.): *Planning in Intelligent Systems* : John Wiley & Sons, Inc., 2006 — ISBN 9780471781264, S. 137–173
- [GeCe04] Geyik, Faruk ; Cedimoglu, Ismail Hakki: The strategies and parameters of tabu search for job-shop scheduling. In: *Journal of intelligent manufacturing* Bd. 15 (2004), Nr. 4, S. 439–448
- [GeTK94] Gen, Mitsuo ; Tsujimura, Yasuhiro ; Kubota, Erika: Solving job-shop scheduling problems by genetic algorithm. In: *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology, 1994 IEEE International Conference on*. Bd. 2 : IEEE, 1994, S. 1577–1582
- [GiTh60] Giffler, B. ; Thompson, G. L.: Algorithms for Solving Production-Scheduling Problems. In: *Operations Research* Bd. 8 (1960), Nr. 4, S. 487–503
- [GLLK79] Graham, R. L. ; Lawler, E. L. ; Lenstra, J. K. ; Kan, A. H. G. Rinnooy: Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In: P.L. Hammer, E. . J. ; Korte, B. . (Hrsg.): *Annals of Discrete Mathematics : Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver*. Bd. 5 : Elsevier, 1979 — ISBN 0167-5060, S. 287–326
- [GoBN05] Gomes, M.C. ; Barbosa-Póvoa, A.P. ; Novais, A.Q.: Optimal scheduling for flexible job shop operation. In: *International Journal of Production Research* Bd. 43 (2005), Nr. 11, S. 2323–2353

- [Gold89] Goldberg, David E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Mass : Addison Wesley Pub Co Inc, 1989 — ISBN 9780201157673
- [GoMR05] Gonçalves, José Fernando ; de Magalhães Mendes, Jorge José ; Resende, Maurício GC: A hybrid genetic algorithm for the job shop scheduling problem. In: *European journal of operational research* Bd. 167 (2005), Nr. 1, S. 77–95
- [Gonz04] Gonzalez, Teofilo: Open Shop Scheduling. In: Leung, J. Y.-T. (Hrsg.): *Handbook of scheduling: algorithms, models and performance analysis*. Boca Raton : Chapman & Hall/CRC, 2004 — ISBN 1584883979 9781584883975
- [GoSa78] Gonzalez, Teofilo ; Sahni, Sartaj: Flowshop and jobshop schedules: complexity and approximation. In: *Operations research* Bd. 26 (1978), Nr. 1, S. 36–52
- [GüGN06] Günther, H-O ; Grunow, Martin ; Neuhaus, Ulf: Realizing block planning concepts in make-and-pack production using MILP modelling and SAP APO\copyright. In: *International Journal of Production Research* Bd. 44 (2006), Nr. 18-19, S. 3711–3726
- [Gura89] Gurari, Eitan: *An Introduction to the Theory of Computation* : Computer Science Press, 1989 — ISBN 9780716781820
- [GuSi02] Gupta, Amit K ; Sivakumar, Appa Iyer: Simulation based multiobjective schedule optimization in semiconductor manufacturing. In: *Simulation Conference, 2002. Proceedings of the Winter*. Bd. 2 : IEEE, 2002, S. 1862–1870
- [Hack89] Hackstein, Rolf: *Produktionsplanung und -steuerung (PPS): ein Handbuch für die Betriebspraxis*. Düsseldorf : VDI-Verlag, 1989 — ISBN 3184009246
- [HaRC05] Hart, Emma ; Ross, Peter ; Corne, David: Evolutionary scheduling: A review. In: *Genetic Programming and Evolvable Machines* Bd. 6 (2005), Nr. 2, S. 191–220
- [Haupt89] Haupt, R.: A survey of priority rule-based scheduling. In: *OR Spektrum* Bd. 11 (1989), Nr. 1, S. 3–16

- [HBBW01] Hestermann, C. ; Beck, J. ; Busch, C. ; Wolber, M. ; Puppe, F. ; Herrler, R. ; Oechslein, C. ; Dilger, W. ; u. a. (Hrsg.): *Wissensbasiertes Scheduling: Ressourcenbelegungsplanung mit heuristischen Suchverfahren; Abschlussbericht des BMBF-Verbundprojektes „INKAD-Intelligentes, kooperatives Assistenzsystem zur Dispositionsunterstützung in Produktionsplanungs-und-steuerungssystemen“* : Univ. Würzburg, Lehrstuhl für Künstliche Intelligenz und Angewandte Informatik, 2001
- [HeLe05] Herroelen, Willy ; Leus, Roel: Project scheduling under uncertainty: Survey and research potentials. In: *European journal of operational research* Bd. 165 (2005), Nr. 2, S. 289–306
- [HeMi09] Hentenryck, Pascal Van ; Michel, Laurent: *Constraint-based local search* : The MIT Press, 2009
- [HJPZ93] Holsapple, Clyde W ; Jacob, Varghese S ; Pakath, Ramakrishnan ; Zaveri, Jigish S: A genetics-based hybrid scheduler for generating static schedules in flexible manufacturing contexts. In: *Systems, Man and Cybernetics, IEEE Transactions on* Bd. 23 (1993), Nr. 4, S. 953–972
- [HKSC06] Hartman, C. ; Kuppens, R. ; Schlesinger, D. ; Cisco Systems, Inc ; Group, Cisco Internet Business Solutions: *Connected Manufacturing: Thoughts-provoking Essays from Industry Leaders* : Premium Pub., 2006 — ISBN 9780955041143
- [Hoc06] Hoc, Jean-Michel: Planning in Dynamic Situations: Some Findings in Complex Supervisory Control. In: Wezel, W. van ; Jorna, R. ; Meystel, A. (Hrsg.): *Planning in Intelligent Systems* : John Wiley & Sons, Inc., 2006 — ISBN 9780471781264, S. 57–97
- [Hoog05] Hoogeveen, Han: Multicriteria scheduling. In: *European Journal of Operational Research, Multicriteria Scheduling*. Bd. 167 (2005), Nr. 3, S. 592–623
- [Hopp12] Hoppe, Marc: *Bestandsoptimierung mit SAP*. Auflage: 3. Aufl. Bonn : SAP PRESS, 2012 — ISBN 9783836218412

- [Horn96] Hornung, V.: *Aachener PPS-Modell: das Prozeßmodell, Forschungsinstitut für Rationalisierung an der RWTH Aachen* : Forschungsinstitut für Rationalisierung an der RWTH Aachen, 1996
- [HSEC09] Hasan, SM Kamrul ; Sarker, Ruhul ; Essam, Daryl ; Cornforth, David: Memetic algorithms for solving job-shop scheduling problems. In: *Memetic Computing* Bd. 1 (2009), Nr. 1, S. 69–83
- [HuJT94] Hurink, Johann ; Jurisch, Bernd ; Thole, Monika: Tabu search for the job-shop scheduling problem with multi-purpose machines. In: *Operations-Research-Spektrum* Bd. 15 (1994), Nr. 4, S. 205–215
- [HuWL10] Huang, Ming ; Wu, Tongwei ; Liang, Xu: GA-ACO in Job-Shop Schedule Problem Research. In: Cai, Z. ; Tong, H. ; Kang, Z. ; Liu, Y. (Hrsg.): *Computational Intelligence and Intelligent Systems, Communications in Computer and Information Science* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-16387-6, 978-3-642-16388-3, S. 226–233
- [Iso14] ISO 22400-2: *Automation systems and integration — Key performance indicators (KPIs) for manufacturing operations management — Part 2: Definitions and descriptions* : ISO/TC 184/SC 5, 2014
- [IvLa96] Ivens, Philip ; Lambrecht, Marc: Extending the shifting bottleneck procedure to real-life applications. In: *European Journal of Operational Research* Bd. 90 (1996), Nr. 2, S. 252–268
- [Jorn06] Jorna, René: Cognition, Planning and Domains: An Empirical Study into the Planning Processes of Planners. In: Wezel, W. van ; Jorna, R. ; Meystel, A. (Hrsg.): *Planning in Intelligent Systems* : John Wiley & Sons, Inc., 2006 — ISBN 9780471781264, S. 99–135
- [JoWS94] Jorge Leon, V. ; Wu, S. D. ; Storer, R. H.: Robustness measures and robust scheduling for job shops. In: *IIE transactions* Bd. 26 (1994), Nr. 5, S. 32–43

- [JuLh02] Jussien, Narendra ; Lhomme, Olivier: Local search with constraint propagation and conflict-based heuristics. In: *Artificial Intelligence* Bd. 139 (2002), Nr. 1, S. 21–45
- [KaHB02] Kacem, Imed ; Hammadi, Slim ; Borne, Pierre: Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. In: *Mathematics and computers in simulation* Bd. 60 (2002), Nr. 3, S. 245–276
- [KáPM04] Kádár, Botond ; Pfeiffer, András ; Monostori, László: Discrete event simulation for supporting production planning and scheduling decisions in digital factories. In: *Proceedings of the 37th CIRP international seminar on manufacturing systems, 2004*, S. 444–448
- [KäTe01] Käschel, Joachim ; Teich, Tobias: Reihenfolgeplanung in Produktionsnetzwerken. In: Jahnke, P. D. B. ; Wall, P. D. F. (Hrsg.): *IT-gestützte betriebswirtschaftliche Entscheidungsprozesse* : Gabler Verlag, 2001 — ISBN 978-3-663-05865-6, 978-3-663-05864-9, S. 239–259
- [KaWH12] Kagermann, H ; Wahlster, W ; Helbig, J: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0–Abschlussbericht des Arbeitskreises Industrie 4.0. In: *Forschungsunion im Stifterverband für die Deutsche Wissenschaft. Berlin* (2012)
- [KCPL13] Krause, Jonas ; Cordeiro, Jelson ; Parpinelli, Rafael Stubs ; Lopes, Heitor Silverio: A survey of swarm algorithms applied to discrete optimization problems. In: *Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Elsevier Science & Technology Books* (2013), S. 169–191
- [King76] King, J. R.: The theory-practice gap in job-shop scheduling. In: *Production Engineer* Bd. 55 (1976), Nr. 3, S. 137–143
- [KLMM02] Klau, Gunnar W ; Lesh, Neal ; Marks, Joe ; Mitzenmacher, Michael ; Schafer, Guy T: The HuGS platform: A toolkit for interactive optimization. In: *Proceedings of the Working Conference on Advanced Visual Interfaces* : ACM, 2002, S. 324–330

- [KoPa01] Kolisch, Rainer ; Padman, Rema: An integrated survey of deterministic project scheduling. In: *Omega* Bd. 29 (2001), Nr. 3, S. 249–272
- [KSWA03] Kunkel-Razum, Kathrin ; Scholze-Stubenrecht, Werner ; Wermke, Matthias ; Auberle, Anette ; Dudenredaktion (Bibliographisches Institut) ; Wissenschaftlicher Rat: *Duden, deutsches Universalwörterbuch*. Mannheim : Dudenverlag, 2003 — ISBN 3411055057 9783411055050
- [Kühn06] Kühn, Wolfgang: *Digitale Fabrik: Fabriksimulation für Produktionsplaner*. München u.a. : Carl Hanser Verlag GmbH & Co. KG, 2006 — ISBN 9783446406193
- [Kuo99] Kuo, Wen-Hung: The development of a human-computer interactive scheduling system. In: *International Journal of Computer Integrated Manufacturing* Bd. 12 (1999), Nr. 2, S. 156–167
- [KuSa99] Kutanoglu, E ; Sabuncuoglu, I.: An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. In: *International Journal of Production Research* Bd. 37 (1999), Nr. 1, S. 165–187
- [LaAL92] Van Laarhoven, Peter JM ; Aarts, Emile HL ; Lenstra, Jan Karel: Job shop scheduling by simulated annealing. In: *Operations research* Bd. 40 (1992), Nr. 1, S. 113–125
- [LaSe97] Lawrence, Stephen R. ; Sewell, Edward C.: Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. In: *Journal of Operations Management* Bd. 15 (1997), Nr. 1, S. 71–82
- [Lawr84] Lawrence, S: Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (supplement). In: *Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania* (1984)
- [LeBT12] Leitão, Paulo ; Barbosa, José ; Trentesaux, Damien: Bio-inspired multi-agent systems for reconfigurable manufacturing systems. In: *Engineering Applications of Artificial Intelligence* Bd. 25 (2012), Nr. 5, S. 934–944

- [Lee08] Lee, E.A.: Cyber Physical Systems: Design Challenges. In: *2008 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 2008, S. 363–369
- [LeHe05] Leus, Roel ; Herroelen, Willy: The complexity of machine scheduling for stability with a single disrupted job. In: *Operations Research Letters* Bd. 33 (2005), Nr. 2, S. 151–156
- [LeKB77] Lenstra, J.K. ; Kan, A.H.G. Rinnooy ; Brucker, P.: Complexity of Machine Scheduling Problems. In: P.L. Hammer, E. . J. B. . K. ; Nemhauser, G. L. (Hrsg.): *Studies in Integer Programming, Annals of Discrete Mathematics*. Bd. 1 : Elsevier, 1977, S. 343–362
- [Leun04a] Leung, Joseph Y-T: *Handbook of scheduling: algorithms, models, and performance analysis*. Boca Raton : Chapman & Hall/CRC, 2004 — ISBN 1584883979 9781584883975
- [Leun04b] Leung, Joseph Y-T: A Tutorial on Complexity. In: *Handbook of scheduling: algorithms, models and performance analysis*. Boca Raton : Chapman & Hall/CRC, 2004 — ISBN 1584883979 9781584883975
- [LiGI97] Lin, Shyh-Chang ; Goodman, Erik D. ; Iii, William F. Punch: Investigating parallel genetic algorithms on job shop scheduling problems. In: Angeline, P. J. ; Reynolds, R. G. ; McDonnell, J. R. ; Eberhart, R. (Hrsg.): *Evolutionary Programming VI, Lecture Notes in Computer Science* : Springer Berlin Heidelberg, 1997 — ISBN 978-3-540-62788-3, 978-3-540-68518-0, S. 383–393
- [LKLB94] Lin, Neng-Pai ; Krajewski, Lee ; Leong, G.K. ; Benton, W.C.: The effects of environmental factors on the design of master production scheduling systems. In: *Journal of Operations Management* Bd. 11 (1994), Nr. 4, S. 367–384

- [LLDV07] Liret, A. ; Lesaint, D. ; Dorne, R. ; Voudouris, C.: iSchedule, an Optimisation Toolkit for Complex Scheduling. In: Baptiste, P. ; Kendall, G. ; Munier-Kordon, A. ; Sourd, F. (Hrsg.): *In proceedings of the 3rd Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2007), 28 -31 August 2007, Paris, France*, 2007. — Paper, S. 344–352
- [Lödd04] Lödding, Hermann: *Verfahren der Fertigungssteuerung. Grundlagen, Beschreibung, Konfiguration*. 1. Aufl. : Springer-Verlag GmbH, 2004 — ISBN 3540202323
- [LoMS01] Lourenço, Helena R ; Martin, Olivier C ; Stutzle, Thomas: Iterated local search. In: *arXiv preprint math/0102188* (2001)
- [Loos99] Loos, Peter: Grunddatenverwaltung und Betriebsdatenerfassung als Basis der Produktionsplanung und–steuerung. In: *Produktionscontrolling. Vahlen, München* (1999), S. 227–252
- [LoWH05] Low, Chinyao ; Wu, Tai-Hsi ; Hsu, Chih-Ming: Mathematical modelling of multi-objective job shop scheduling with dependent setups and re-entrant operations. In: *The International Journal of Advanced Manufacturing Technology* Bd. 27 (2005), Nr. 1-2, S. 181–189
- [LuAl13] Luque, Gabriel ; Alba, Enrique: *Parallel Genetic Algorithms: Theory and Real World Applications* : Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2013
- [MaBi04] Mattfeld, Dirk C ; Bierwirth, Christian: An efficient genetic algorithm for job shop scheduling with tardiness objectives. In: *European Journal of Operational Research, Traffic and Transportation Systems Analysis*. Bd. 155 (2004), Nr. 3, S. 616–630
- [Macc01] MacCarthy, B. L. ; MacCarthy, B. ; Wilson, J. R. (Hrsg.): *Human Performance in Planning and Scheduling*. Auflage: 0005. Aufl. Tokyo : Crc Pr Inc, 2001 — ISBN 9784431243328

- [Mack11] Mack, C.A.: Fifty Years of Moore's Law. In: *IEEE Transactions on Semiconductor Manufacturing* Bd. 24 (2011), Nr. 2, S. 202–207
- [MaIN85] Masuda, Teruo ; Ishii, Hiroaki ; Nishida, Toshio: The mixed shop scheduling problem. In: *Discrete applied mathematics* Bd. 11 (1985), Nr. 2, S. 175–186
- [MaLu93] MacCarthy, B. L. ; Lui, Jiyin: Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. In: *International Journal of Production Research* Bd. 31 (1993), Nr. 1, S. 59–79
- [Mark13] MarketsandMarkets: *Smart Factory Market by Field Devices, Application & Technology 2013 - 2018*, 2013. — market size, market share, market survey, market intelligence, market trends, market strategy, market research report, analysis, survey, market research surveys
- [McSB95] McKay, Kenneth N ; Safayeni, Frank R ; Buzacott, John A: A review of hierarchical production planning and its applicability for modern manufacturing. In: *Production Planning & Control* Bd. 6 (1995), Nr. 5, S. 384–394
- [McWi06] McKay, Kenneth N. ; Wiers, Vincent C.S.: The Organizational Interconnectivity of Planning and Scheduling. In: Wezel, W. van ; Jorna, R. ; Meystel, A. (Hrsg.): *Planning in Intelligent Systems* : John Wiley & Sons, Inc., 2006 — ISBN 9780471781264, S. 175–201
- [McWi99] McKay, Kenneth N. ; Wiers, Vincent C. S.: Unifying the theory and practice of production scheduling. In: *Journal of Manufacturing Systems, Special issue on scheduling: From Research Into Practice*. Bd. 18 (1999), Nr. 4, S. 241–255
- [MeGr09] Mell, Peter ; Grance, Tim: The NIST definition of cloud computing. In: *National Institute of Standards and Technology* Bd. 53 (2009), Nr. 6, S. 50
- [Mesa97] MESA: MES Explained: A High Level Vision. In: *International White Paper* (1997), Nr. 6

- [MeTo95] Meinberg, Uwe ; Topolewski, Frank: *Lexikon der Fertigungsleittechnik: Begriffe, Erläuterungen, Beispiele*. Berlin; New York : Springer-Verlag, 1995 — ISBN 3540585230 9783540585237
- [MeUz98] Mehta, S.V. ; Uzsoy, R.M.: Predictable scheduling of a job shop subject to breakdowns. In: *IEEE Transactions on Robotics and Automation* Bd. 14 (1998), Nr. 3, S. 365–378
- [MeWR08] Meyr, Herbert ; Wagner, Michael ; Rohde, Jens: Structure of Advanced Planning Systems. In: Stadler, P. D. H. ; Kilger, D. C. (Hrsg.): *Supply Chain Management and Advanced Planning* : Springer Berlin Heidelberg, 2008 — ISBN 978-3-540-74511-2, 978-3-540-74512-9, S. 109–115
- [MFDM11] Mönch, Lars ; Fowler, John W. ; Dauzère-Pérès, Stéphane ; Mason, Scott J. ; Rose, Oliver: A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. In: *Journal of scheduling*. - Norwell, Mass : Springer Science Business Media, ISSN 1094-6136, ZDB-ID 14924869. - Vol. 14.2011, 6, S. 583-599 (2011)
- [Miet99] Miettinen, Kaisa: *Nonlinear Multiobjective Optimization* : Springer, 1999 — ISBN 9780792382782
- [MöFM13] Mönch, Lars ; Fowler, John W. ; Mason, Scott J.: Dispatching Approaches. In: *Production Planning and Control for Semiconductor Wafer Fabrication Facilities, Operations Research/Computer Science Interfaces Series* : Springer New York, 2013 — ISBN 978-1-4614-4471-8, 978-1-4614-4472-5, S. 65–104
- [Mönc06] Mönch, Lars: *Agentenbasierte Produktionssteuerung komplexer Produktionssysteme*. Auflage: 2006. Aufl. Wiesbaden : Vieweg+Teubner Verlag, 2006 — ISBN 9783835002494
- [MoPV11] Moratori, Patrick ; Petrovic, Sanja ; Vázquez-Rodríguez, José Antonio: Match-up approaches to a dynamic rescheduling problem. In: *International Journal of Production Research* Bd. 50 (2011), Nr. 1, S. 261–276

- [Mose93] Moser, Manfred: *Regelung der Maschinenbelegung in der flexiblen Fertigung*, Zugl.: Karlsruhe, Univ., Diss., 1993, 1993. — 206 S.
- [MSPF07] Mönch, Lars ; Schabacker, Rene ; Pabst, Detlef ; Fowler, John W.: Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. In: *European Journal of Operational Research* Bd. 177 (2007), Nr. 3, S. 2100–2118
- [NaDZ02] Najid, NM ; Dauzere-Peres, S ; Zaidat, A: A modified simulated annealing method for flexible job shop scheduling problem. In: *Systems, Man and Cybernetics, 2002 IEEE International Conference on*. Bd. 5 : IEEE, 2002, S. 6–pp
- [NaHH95] Nagar, Amit ; Haddock, Jorge ; Heragu, Sunderesh: Multiple and bicriteria scheduling: A literature survey. In: *European Journal of Operational Research* Bd. 81 (1995), Nr. 1, S. 88 – 104
- [NaYa91] Nakano, Ryohei ; Yamada, Takeshi: Conventional Genetic Algorithm for Job Shop Problems. In: *ICGA*. Bd. 91, 1991, S. 474–79
- [NoGr91] Nof, Shimon Y ; Grant, Hank F: Adaptive/predictive scheduling: review and a general framework. In: *Production Planning & Control* Bd. 2 (1991), Nr. 4, S. 298–312
- [OuPe09] Ouelhadj, Djamila ; Petrovic, Sanja: A survey of dynamic scheduling in manufacturing systems. In: *J. of Scheduling* Bd. 12 (2009), Nr. 4, S. 417–431
- [OvUz95] Ovacik, Irfan M. ; Uzsoy, Reha M.: Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. In: *International Journal of Production Research* Bd. 33 (1995), Nr. 11, S. 3173–3192

- [OvUz97] Ovacik, Irfan M. ; Uzsoy, Reha: Review of Decomposition Methods for Factory Scheduling Problems. In: *Decomposition Methods for Complex Factory Scheduling Problems* : Springer US, 1997 — ISBN 978-1-4613-7906-5, 978-1-4615-6329-7, S. 31–45
- [OwST88] Ow, Peng Si ; Smith, Stephen F ; Thirlez, Alfred: Reactive Plan Revision. In: *AAAI*, 1988, S. 77–82
- [PaBa98] Le Pape, Claude ; Baptiste, Philippe: Resource constraints for preemptive job-shop scheduling. In: *Constraints* Bd. 3 (1998), Nr. 4, S. 263–287
- [Pape94] Le Pape, Claude: Implementation of resource constraints in ILOG Schedule: A library for the development of constraint-based scheduling systems. In: *Intelligent Systems Engineering* Bd. 3 (1994), Nr. 2, S. 55–66
- [Paul95] Paulli, Jan: A hierarchical approach for the FMS scheduling problem. In: *European Journal of Operational Research* Bd. 86 (1995), Nr. 1, S. 32–42
- [PaWi79] Page, Ewan Stafford ; Wilson, Leslie Blackett: *An introduction to computational combinatorics*. Bd. 9 : CUP Archive, 1979
- [PeMC08] Pezzella, F. ; Morganti, G. ; Ciaschetti, G.: A genetic algorithm for the Flexible Job-shop Scheduling Problem. In: *Computers & Operations Research, Part Special Issue: Search-based Software Engineering*. Bd. 35 (2008), Nr. 10, S. 3202–3212
- [PfkM06] Pfeiffer, András ; Kádár, Botond ; Monostori, László: Stability-oriented evaluation of hybrid rescheduling methods in a job-shop with machine breakdowns. In: *CIRP Manuf Syst* Bd. 35 (2006), Nr. 6, S. 563–570
- [Pine14] Pinedo, Michael L.: *Scheduling Theory, Algorithms, and Systems*. : Springer Verlag, 2014 — ISBN 9781489990433 1489990437
- [Pine95] Pinedo, Michael L.: *Scheduling: theory, algorithms, and systems*. : Prentice-Hall, Englewood Cliffs, NJ, 1995

- [PiSi99] Pinedo, Michael ; Singer, Marcos: A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. In: *Naval Research Logistics (NRL)* Bd. 46 (1999), Nr. 1, S. 1–17
- [PKCM05] Pfeiffer, András ; Kádár, Botond ; Csáji, Balázs Csanád ; Monostori, László: Simulation supported analysis of a dynamic rescheduling system. In: *Manufacturing, Modelling, Management and Control 2005* (2005), S. 25–30
- [PoWo06] Pochet, Yves ; Wolsey, Laurence A.: *Production Planning by Mixed Integer Programming* : Springer, 2006 — ISBN 9780387299594
- [PrST04] Pruhs, Kirk ; Sgall, Jiri ; Tornø, Eric: Online Scheduling. In: Leung, J. Y.-T. (Hrsg.): *Handbook of scheduling: algorithms, models and performance analysis*. Boca Raton : Chapman & Hall/CRC, 2004 — ISBN 1584883979 9781584883975
- [Puja04] Pujawan, I Nyoman: Schedule nervousness in a manufacturing system: a case study. In: *Production planning & control* Bd. 15 (2004), Nr. 5, S. 515–524
- [PuRa05] Puchinger, Jakob ; Raidl, Günther R.: *Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification* : Springer, 2005
- [QWHW08] Qian, Bin ; Wang, Ling ; Huang, De-Xian ; Wang, Xiong: Scheduling multi-objective job shops using a memetic algorithm based on differential evolution. In: *The International Journal of Advanced Manufacturing Technology* Bd. 35 (2008), Nr. 9-10, S. 1014–1027
- [RaFK04] Rangsaritratsamee, Ruedee ; Ferrell Jr., William G. ; Kurz, Mary Beth: Dynamic rescheduling that simultaneously considers efficiency and stability. In: *Computers & Industrial Engineering* Bd. 46 (2004), Nr. 1, S. 1–15
- [Refa76] REFA: *Datenermittlung Teil 2: Methodenlehre des Arbeitsstudiums*. München : Hanser, 1976 — ISBN 3446142347 9783446142343 3446127046 9783446127043 3446142363 9783446142367

- [Refa90] REFA: *Methodenlehre der Betriebsorganisation, Planung und Gestaltung komplexer Produktionssysteme*. München : Hanser, 1990 — ISBN 3446159673
- [RFWF10] Riedel, Ralph ; Fransoo, Jan ; Wiers, Vincent ; Fischer, Katrin ; Cegarra, Julien ; Jentsch, David: Building Decision Support Systems for Acceptance. In: Fransoo, J. C. ; Waefler, T. ; Wilson, J. R. (Hrsg.): *Behavioral Operations in Planning and Scheduling* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-13381-7, 978-3-642-13382-4, S. 231–295
- [RHMD10a] Riezebos, Jan ; Hoc, Jean-Michel ; Mebarki, Nasser ; Dimopoulos, Christos ; Wezel, Wout van ; Pinot, Guillaume: Design of Scheduling Algorithms. In: Fransoo, J. C. ; Waefler, T. ; Wilson, J. R. (Hrsg.): *Behavioral Operations in Planning and Scheduling* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-13381-7, 978-3-642-13382-4, S. 299–321
- [RHMD10b] Riezebos, Jan ; Hoc, Jean-Michel ; Mebarki, Nasser ; Dimopoulos, Christos ; Wezel, Wout Van ; Pinot, Guillaume: Design of Scheduling Algorithms: Applications. In: Fransoo, J. C. ; Waefler, T. ; Wilson, J. R. (Hrsg.): *Behavioral Operations in Planning and Scheduling* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-13381-7, 978-3-642-13382-4, S. 371–412
- [Rinn76] Rinnooy Kan, A.H.G.: *Machine scheduling problems: classification, complexity and computations* : Stenfert Kroese, 1976 — ISBN 9789020705676
- [RoDi00] Rossi, Andrea ; Dini, Gino: Dynamic scheduling of FMS using a real-time genetic algorithm. In: *International Journal of Production Research* Bd. 38 (2000), Nr. 1, S. 1–20
- [RoDi07] Rossi, Andrea ; Dini, Gino: Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. In: *Robotics and Computer-Integrated Manufacturing* Bd. 23 (2007), Nr. 5, S. 503–516
- [RuVá10] Ruiz, Rubén ; Vázquez-Rodríguez, José Antonio: The hybrid flow shop scheduling problem. In: *European Journal of Operational Research* Bd. 205 (2010), Nr. 1, S. 1–18

- [RWM005] Reeves, Daniel M ; Wellman, Michael P ; MacKie-Mason, Jeffrey K ; Osepayshvili, Anna: Exploring bidding strategies for market-based scheduling. In: *Decision Support Systems* Bd. 39 (2005), Nr. 1, S. 67–85
- [SaBa00] Sabuncuoglu, Ihsan ; Bayız, M: Analysis of reactive scheduling problems in a job shop environment. In: *European Journal of operational research* Bd. 126 (2000), Nr. 3, S. 567–586
- [SaBr97] Sauer, Jurgen ; Bruns, Ralf: Knowledge-based scheduling systems in industry and medicine. In: *IEEE Expert* Bd. 12 (1997), Nr. 1, S. 24–31
- [SaCh79] Sahni, Sartaj ; Cho, Yookun: Complexity of scheduling shops with no wait in process. In: *Mathematics of Operations Research* Bd. 4 (1979), Nr. 4, S. 448–457
- [SaKi03] Sabuncuoglu, Ihsan ; Kizilisik, Omer Batuhan: Reactive scheduling in a dynamic and stochastic FMS environment. In: *International Journal of Production Research* Bd. 41 (2003), Nr. 17, S. 4211–4231
- [SaVW11] Sarin, Subhash C ; Varadarajan, Amrusha ; Wang, Lixin: A survey of dispatching rules for operational control in wafer fabrication. In: *Production Planning and Control* Bd. 22 (2011), Nr. 1, S. 4–24
- [SBMR02] Sampels, Michael ; Blum, Christian ; Mastrolilli, Monaldo ; Rossi-Doria, Olivia: Metaheuristics for Group Shop Scheduling. In: Guervós, J. J. M. ; Adamidis, P. ; Beyer, H.-G. ; Schwefel, H.-P. ; Fernández-Villacañes, J.-L. (Hrsg.): *Parallel Problem Solving from Nature — PPSN VII, Lecture Notes in Computer Science* : Springer Berlin Heidelberg, 2002 — ISBN 978-3-540-44139-7, 978-3-540-45712-1, S. 631–640
- [ScGi06] Schuh, Günther ; Gierrth, Andreas: Aachener PPS-Modell. In: Schuh, P. D.-I. D.-W. I. G. (Hrsg.): *Produktionsplanung und -steuerung, VDI-Buch* : Springer Berlin Heidelberg, 2006 — ISBN 978-3-540-40306-7, 978-3-540-33855-0, S. 11–27

- [Sche87] Scheer, August-Wilhelm: *CIM Computer integrated manufacturing: Der computergesteuerte Industriebetrieb*. Berlin; Heidelberg; New York; London; Paris; Tokyo : Springer, 1987 — ISBN 3540182829 9783540182825 0387182829 9780387182827 0387177426 9780387177427 3540177426 9783540177425
- [SeAy07] Sevkli, Mehmet ; Aydin, M Emin: Parallel variable neighbourhood search algorithms for job shop scheduling problems. In: *IMA Journal of Management Mathematics* Bd. 18 (2007), Nr. 2, S. 117–133
- [SFPP10] Schuh, Günther ; Franzkoch, B. ; Potente, T. ; Fuchs, S.: Simulation based configuration of value stream oriented production control. In: *POM 2010 Conference program bulletin*. Vancouver, 2010, S. 144
- [SGLW08] Sha, Lui ; Gopalakrishnan, S. ; Liu, Xue ; Wang, Qixin: Cyber-Physical Systems: A New Frontier. In: *IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08*, 2008, S. 1–9
- [ShNo99] Shen, Weiming ; Norrie, Douglas H: Agent-based systems for intelligent manufacturing: a state-of-the-art survey. In: *Knowledge and information systems* Bd. 1 (1999), Nr. 2, S. 129–156
- [ShWH06] Shen, Weiming ; Wang, Lihui ; Hao, Qi: Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* Bd. 36 (2006), Nr. 4, S. 563–577
- [SHYN06] Shen, Weiming ; Hao, Qi ; Yoon, Hyun Joong ; Norrie, Douglas H: Applications of agent-based systems in intelligent manufacturing: An updated review. In: *Advanced engineering INFORMATICS* Bd. 20 (2006), Nr. 4, S. 415–431
- [Smit92] Smith, Stephen F: Knowledge-based production management approaches, results and prospects. In: *Production Planning & Control* Bd. 3 (1992), Nr. 4, S. 350–380

- [Smit95] Smith, Stephen F: Reactive scheduling systems. In: *Intelligent scheduling systems* : Springer, 1995, S. 155–192
- [SmLB96] Smith, Stephen F ; Lassila, Ora ; Becker, Marcel: Configurable, mixed-initiative systems for planning and scheduling. In: *Advanced Planning Technology* (1996), S. 235–241
- [SMMO90] Smith, S.F. ; Muscettola, N. ; Matthys, D.C. ; Ow, P.S. ; Potvin, J.Y.: OPIS: An opportunistic factory scheduling system. In: *Proceedings of the 3rd international conference on Industrial and engineering applications of artificial intelligence and expert systems-Volume 1, IEA/AIE '90*. New York, NY, USA : ACM, 1990 — ISBN 0-89791-372-8, S. 268–274
- [SnWe10] Snoo, Cees De ; Wezel, Wout van: The Interconnectivity of Planning and Shop Floor: Case Description and Relocation Analysis. In: Fransoo, J. C. ; Waefler, T. ; Wilson, J. R. (Hrsg.): *Behavioral Operations in Planning and Scheduling* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-13381-7, 978-3-642-13382-4, S. 31–43
- [SoSh95] Sotskov, Yu N ; Shakhlevich, Natalia V: NP-hardness of shop-scheduling problems with three jobs. In: *Discrete Applied Mathematics* Bd. 59 (1995), Nr. 3, S. 237–266
- [SrGh86] Srikar, Bellur N ; Ghosh, Soumen: A MILP model for the n-job, m-stage flowshop with sequence dependent set-up times. In: *International Journal of Production Research* Bd. 24 (1986), Nr. 6, S. 1459–1474
- [TBBL05] T'kindt, Vincent ; Billaut, J-C ; Bouquard, J-L ; Lenté, Christophe ; Martineau, Patrick ; Néron, Emmanuel ; Proust, Christian ; Tacquard, C: The e-OCEA project: towards an internet decision system for scheduling problems. In: *Decision support systems* Bd. 40 (2005), Nr. 2, S. 329–337
- [Tech13] TechNavio: *Global MES Market 2012-2016*, 2013

- [Thie11] Thiel, Klaus: *MES - Integriertes Produktionsmanagement. Leitfaden, Marktübersicht und Anwendungsbeispiele. Mit CD-ROM*. München : Carl Hanser Verlag GmbH & Co. KG, 2011 — ISBN 9783446421141
- [Thom96] Thomalla, Christoph Stephan: Job shop scheduling using Lagrangean relaxation. In: *Journal of decision systems* Bd. 5 (1996), Nr. 1-2, S. 111–127
- [Timk04] Timkovsky, Vadim G.: Cycle Shop Scheduling. In: Leung, J. Y.-T. (Hrsg.): *Handbook of scheduling: algorithms, models and performance analysis*. Boca Raton : Chapman & Hall/CRC, 2004 — ISBN 1584883979 9781584883975
- [UhKü88] Aue-Uhlhausen, H ; Kühnle, H: Von ABS bis OPT-PPS-Methoden im Vergleich. In: *PPS '88. Kongreß, Fachausstellung. Blick in die Forschung*, 1988, S. 179–230
- [Ullm75] Ullman, J. D.: NP-complete scheduling problems. In: *Journal of Computer and System Sciences* Bd. 10 (1975), Nr. 3, S. 384–393
- [VaAL96] Vaessens, R. J. M. ; Aarts, E. H. L. ; Lenstra, J. K.: Job Shop Scheduling by Local Search. In: *INFORMS Journal on Computing* Bd. 8 (1996), Nr. 3, S. 302–317
- [VaBQ08] Valls, Vicente ; Ballestín, Francisco ; Quintanilla, Sacramento: A hybrid genetic algorithm for the resource-constrained project scheduling problem. In: *European Journal of Operational Research* Bd. 185 (2008), Nr. 2, S. 495–508
- [Vdi07] VDI-5600: *Manufacturing execution system (MES) - VDI 5600 Part 1. : Verein Deutscher Ingenieure*, 2007
- [Vdiv83] VDI-Verlag: *Lexikon der Produktionsplanung und -steuerung: Begriffszusammenhänge und Begriffsdefinitionen*. 3. Aufl. edition. Aufl. : Springer, 1983 — ISBN 3184006050
- [Vdma09] VDMA-66412-1: *Manufacturing Execution Systems (MES) – Kennzahlen*. Beuth Verlag GmbH, Berlin, 2009

- [ViHL03] Vieira, Guilherme E. ; Herrmann, Jeffrey W. ; Lin, Edward: Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. In: *Journal of Scheduling* Bd. 6 (2003), Nr. 1, S. 39–62
- [VKBW09] Vogel-Heuser, B ; Kegel, G ; Bender, K ; Wucherer, K: Global information architecture for industrial automation. In: *ATP International-Automation Technology in Practice* Bd. 51 (2009), Nr. 1, S. 108–115
- [VoDH07] Van de Vonder, Stijn ; Demeulemeester, Erik ; Herroelen, Willy: A classification of predictive-reactive project scheduling procedures. In: *Journal of Scheduling* Bd. 10 (2007), Nr. 3, S. 195–207
- [WaG004] Wang, Xiaolei ; Gao, Xiao Zhi ; Ovaska, Seppo J: Artificial immune optimization methods and applications-a survey. In: *Systems, Man and Cybernetics, 2004 IEEE International Conference on*. Bd. 4 : IEEE, 2004, S. 3415–3420
- [WaIG05] Watanabe, Masato ; Ida, Kenichi ; Gen, Mitsuo: A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem. In: *Computers & Industrial Engineering, Selected Papers from The 30th International Conference on Computers & Industrial Engineering*. Bd. 48 (2005), Nr. 4, S. 743–752
- [Wall96] Wall, Matthew Bartschi: *A genetic algorithm for resource-constrained scheduling*, Massachusetts Institute of Technology, 1996
- [WBKK10] Wagner, Stefan ; Beham, Andreas ; Kronberger, GK ; Kommenda, Michael ; Pitzer, Erik ; Kofler, Monika ; Vonolfen, Stefan ; Winkler, SM ; u. a.: Heuristiclab 3.3: A unified approach to metaheuristic optimization. In: *Actas del séptimo congreso español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'2010)*, 2010, S. 8

- [WeCH10] Wezel, Wout van ; Cegarra, Julien ; Hoc, Jean-Michel: Allocating Functions to Human and Algorithm in Scheduling. In: Fransoo, J. C. ; Waefler, T. ; Wilson, J. R. (Hrsg.): *Behavioral Operations in Planning and Scheduling* : Springer Berlin Heidelberg, 2010 — ISBN 978-3-642-13381-7, 978-3-642-13382-4, S. 339–370
- [WeJo09] Wezel, Wout Van ; Jorna, René: Cognition, tasks and planning: supporting the planning of shunting operations at the Netherlands Railways. In: *Cogn. Technol. Work* Bd. 11 (2009), Nr. 2, S. 165–176
- [WeJR03] Westkämper, E. ; Jovanoski, D. ; Rist, T.: New framework for digital factory planning. In: *International Seminar on Manufacturing Systems*. Saarbrücken, 2003 — ISBN 3930429586, S. 191–198
- [WeYa08] Webster, Steve ; Yard, Todd: Getting Started with Actionscript 3.0. In: *Foundation ActionScript 3.0 with Flash CS3 and Flex* : Springer, 2008, S. 1–23
- [Weze06] van Wezel, Wout: Interactive Scheduling Systems. In: Wezel, W. van ; Jorna, R. ; Meystel, A. (Hrsg.): *Planning in Intelligent Systems* : John Wiley & Sons, Inc., 2006 — ISBN 9780471781264, S. 203–241
- [WhRo90] White, K. P. ; Rogers, R. V.: Job-shop scheduling: limits of the binary disjunctive formulation. In: *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH* Bd. 28 (1990), Nr. 12, S. 2187–2200
- [Wigh84] Wight, Oliver W: *Manufacturing resource planning, MRP II: unlocking America's productivity potential*. New York; Chichester : Wiley, 1984 — ISBN 0471132748 9780471132745
- [Will89] Williams, Theodore J: *A Reference model for computer integrated manufacturing (CIM): a description from the viewpoint of industrial automation*. Research Triangle Park, N.C. : Instrument Society of America, 1989 — ISBN 1556172257 9781556172250

- [Will98] Williams, T: The Purdue enterprise reference architecture and methodology (PERA). In: *Handbook of life cycle engineering: concepts, models, and technologies* Bd. 289 (1998)
- [WuSC93] Wu, David ; Storer, Robert ; Chang, Pei: One-Machine Rescheduling Heuristics with Efficiency and Stability as Criteria. In: *Computers \& Operations Research* Bd. 20 (1993), Nr. 1, S. 1–14
- [WZGL08] Wang, Xiao-juan ; Zhang, Chao-yong ; Gao, Liang ; Li, Pei-gen: A survey and future trend of study on multi-objective scheduling. In: *IEEE Fourth International Conference on Natural Computation, 2008*
- [XWZW12] Xu, Ye ; Wang, Ling ; Zhou, Gang ; Wang, Shengyao: An Effective Shuffled Frog Leaping Algorithm for Solving Hybrid Flow-Shop Scheduling Problem. In: Huang, D.-S. ; Gan, Y. ; Bevilacqua, V. ; Figueroa, J. C. (Hrsg.): *Advanced Intelligent Computing, Lecture Notes in Computer Science* : Springer Berlin Heidelberg, 2012 — ISBN 978-3-642-24727-9, 978-3-642-24728-6, S. 560–567
- [YaAZ10] Yazdani, M. ; Amiri, M. ; Zandieh, M.: Flexible job-shop scheduling with parallel variable neighborhood search algorithm. In: *Expert Systems with Applications* Bd. 37 (2010), Nr. 1, S. 678–687
- [YPHY10] Younus, M. ; Peiyong, Cong ; Hu, Lu ; Yuqing, Fan: MES development and significant applications in manufacturing -A review. In: *2010 2nd International Conference on Education Technology and Computer (ICETC)*. Bd. 5, 2010, S. V5–97–V5–101
- [ZDDD93] Zweben, M. ; Davis, E. ; Daun, B. ; Deale, M.J.: Scheduling and rescheduling with iterative repair. In: *IEEE Transactions on Systems, Man and Cybernetics* Bd. 23 (1993), Nr. 6, S. 1588–1596

Eigene Veröffentlichungen

- [BaDi08] Baumann, Michael ; Dimitrov, Todor: Hierarchisierte Modelle in Advanced Planning and Scheduling (APS). In: *Automatisierungstechnik : AT* Bd. 56 (2008), Nr. Nr.2, S. 90–97
- [BaDi11a] Baumann, Michael ; Dimitrov, Todor: Fertigungsmanagementsysteme werden zunehmend intelligenter. In: *Maschinenmarkt. MM, das Industriemagazin* (2011), Nr. Nr.12
- [BaDi11b] Baumann, Michael ; Dimitrov, Todor: Simulation zur Auswahl und Parametrierung einer Produktionssteuerung: Risikolos verstecktes Optimierungspotenzial erschließen. In: *Productivity management* Bd. 16 (2011), Nr. Nr.1, S. 23–25
- [BaDM06] Baumann, Michael ; Dimitrov, Todor ; Martinek, Robert: Komplexität in der Produktion bWeherrschen - Adaptivität und Interoperabilität in Produktionsplanung und -steuerung mit dem ISA-95 Standard. In: *PPS Management* Bd. 11 (2006), Nr. Nr.3, S. 29–32
- [DiBa09] Dimitrov, Todor ; Baumann, Michael: Simulationsumgebung zur Analyse von Algorithmen für dynamische Scheduling-Probleme im Produktionsumfeld. In: *Informatik 2009. Im Focus das Leben, 2009* — ISBN 9783885792482, S. 471, 15
- [DiBa11] Dimitrov, Todor ; Baumann, Michael: Genetic algorithm with genetic engineering technology for multi-objective dynamic job shop scheduling problems. In: *GECCO 2011, 13th Annual Conference Companion on Genetic and Evolutionary Computation. Proceedings, 2011* — ISBN 9781450306904, S. 833–834

- [DiBa12] Dimitrov, Todor ; Baumann, Michael: Criteria evaluation considering the current time used by a genetic scheduling algorithm for changeover and tardiness reduction. In: *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, 2012, S. 425 –430
- [DiBS10] Dimitrov, Todor ; Baumann, Michael ; Schenk, Manfred: Master data extraction and adaptation based on collected production data in manufacturing execution systems. In: *CIRP ICME 2010, 7th CIRP International Conference on Intelligent Computation in Manufacturing Engineering. Innovative and Cognitive Production Technology and Systems*, 2010
- [DiBS11] Dimitrov, Todor ; Baumann, Michael ; Schenk, Manfred: Scheduling of large dynamic job shop problems by concurrent non-blocking threads. In: *5th Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA 2011*, 2011

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

Herausgeber: Prof. Dr.-Ing. Jürgen Beyerer

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar
oder als Druckausgabe bestellbar.

- Band 1** Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz. 2006
ISBN 3-86644-070-7

- Band 2** Elisabeth Peinsipp-Byma
**Leistungserhöhung durch Assistenz in interaktiven Systemen
zur Szenenanalyse.** 2007
ISBN 978-3-86644-149-1

- Band 3** Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme. 2010
ISBN 978-3-86644-457-7

- Band 4** Jürgen Beyerer, Marco Huber (Hrsg.)
**Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and
Institute for Anthropomatics, Vision and Fusion Laboratory.** 2010
ISBN 978-3-86644-469-0

- Band 5** Thomas Usländer
Service-oriented design of environmental information systems. 2010
ISBN 978-3-86644-499-7

- Band 6** Giulio Milighetti
**Multisensorielle diskret-kontinuierliche Überwachung und
Regelung humanoider Roboter.** 2010
ISBN 978-3-86644-568-0

- Band 7** Jürgen Beyerer, Marco Huber (Hrsg.)
**Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and
Institute for Anthropomatics, Vision and Fusion Laboratory.** 2011
ISBN 978-3-86644-609-0

- Band 8** Eduardo Monari
**Dynamische Sensorselektion zur auftragsorientierten
Objektverfolgung in Kameranetzwerken.** 2011
ISBN 978-3-86644-729-5

- Band 9** Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen. 2011
ISBN 3-86644-760-8
- Band 10** Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile. 2012
ISBN 978-3-86644-798-1
- Band 11** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2012
ISBN 978-3-86644-855-1
- Band 12** Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES). 2013
ISBN 978-3-86644-955-8
- Band 13** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2013
ISBN 978-3-86644-988-6
- Band 14** Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip. 2013
ISBN 978-3-7315-0041-4
- Band 15** Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8
- Band 16** Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung. 2014
ISBN 978-3-7315-0167-1
- Band 17** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0212-8
- Band 18** Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance. 2015
ISBN 978-3-7315-0320-0

- Band 19** Marco Huber
**Nonlinear Gaussian Filtering:
Theory, Algorithms, and Applications.** 2015
ISBN 978-3-7315-0338-5
- Band 20** Jürgen Beyerer, Alexey Pak (Hrsg.)
**Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and
Institute for Anthropomatics, Vision and Fusion Laboratory.** 2014
ISBN 978-3-7315-0401-6
- Band 21** Todor Dimitrov
**Permanente Optimierung dynamischer Probleme
der Fertigungssteuerung unter Einbeziehung von
Benutzerinteraktionen.** 2015
ISBN 978-3-7315-0426-9

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

Trotz enormen Forschungsaufwands erhalten die Entscheider in der Fertigungssteuerung bei ihrem Tagesgeschäft nur rudimentäre Rechnerunterstützung. Diese Arbeit schlägt ein umfassendes Konzept für eine permanent laufende algorithmische Feinplanung vor, das den Weg zu einer intelligenten Kollaboration mit den Entscheidern in der Fertigungssteuerung erlaubt. Die erarbeiteten Ansätze ermöglichen eine zeitnahe Reaktionsfähigkeit auf Fertigungsereignisse und intelligente Umplanungen nach einer umfassenden Analyse des Optimierungspotentials und -bedarfs. Simulationen mit zwei Datensätzen von unterschiedlichen Fertigungsunternehmen bestätigen die Praxistauglichkeit der entwickelten Ansätze.

ISSN 1863-6489
ISBN 978-3-7315-0426-9

