

Jan Wesemann

Ein Beitrag zum Benchmarking von Planungsverfahren und zur Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter der Fachgruppe „Wirtschaftsinformatik, insbesondere CIM“ am Heinz Nixdorf Institut der Universität Paderborn.

Für die während dieser Zeit erfahrene Betreuung und Unterstützung, aber auch für die Möglichkeit, mich fachlich und persönlich weiterzuentwickeln, danke ich herzlich dem Leiter der Fachgruppe Herrn Prof. Dr.-Ing. habil. Wilhelm Dangelmaier.

Herrn Prof. Dr.-Ing. Bernd Hellingrath danke ich für die Durchsicht der Dissertation und die Übernahme des Korreferates.

Für die Unterstützung meines Dissertationsvorhabens bedanke ich mich herzlich bei der Begabtenförderung der Konrad-Adenauer-Stiftung.

Mein Dank gilt weiterhin allen wissenschaftlichen und studentischen Mitarbeitern der Fachgruppe, die durch gemeinsame Diskussionen und wertvolle Anregungen zum Gelingen dieser Arbeit beigetragen haben und meinen Eltern, die mich immer unterstützt haben.

Paderborn, Januar 2009

Jan Wesemann

Ein Beitrag zum Benchmarking von Planungsverfahren und zur
Produktionsplanung und -steuerung für die mehrstufige auftragsbe-
zogene Serienfertigung in der Automobilzulieferindustrie.

Dissertation
zur Erlangung der Würde eines
DOKTORS DER WIRTSCHAFTSWISSENSCHAFTEN
(Dr. rer. pol.)
der Universität Paderborn

vorgelegt von
Dipl.-Wirt.-Inf. Jan Wesemann

Paderborn, den 2. Mai 2008

Dekan: Prof. Dr. Peter F. E. Sloane

Referent: Prof. Dr.-Ing. habil. Wilhelm Dangelmaier

Korreferent: Prof. Dr.-Ing. Bernd Hellingrath

Inhaltsverzeichnis

ABBILDUNGSVERZEICHNIS	V
TABELLENVERZEICHNIS	IX
DEFINITIONSVERZEICHNIS	XI
ABKÜRZUNGSVERZEICHNIS.....	XIII
SYMBOLVERZEICHNIS.....	XV
1 EINLEITUNG	19
2 PROBLEMDEFINITION	23
2.1 AUSGANGSLAGE.....	23
2.2 FACHBEGRIFFE	27
2.3 ZIELSETZUNG DER ARBEIT	35
2.3.1 Zielsetzung für das Benchmarking von Planungsverfahren.....	35
2.3.1.1 Zielsetzung für die Modellierung des Planungsproblems	35
2.3.1.2 Zielsetzung für die Simulation des Produktionsablaufs	36
2.3.2 Zielsetzung für die Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie.....	37
2.3.2.1 Zielsetzung für Planungsstrategien	37
2.3.2.2 Zielsetzung für die Produktionsplanung und -steuerung	37
2.3.2.3 Zielsetzung für die Produktionsplanung und -steuerung unter Unsicherheit 38	
2.4 STRUKTURIERUNG DER PROBLEMSTELLUNG	38
2.5 ANFORDERUNGEN AN DIE PROBLEMLÖSUNG.....	39
2.5.1 Anforderungen an das Benchmarking von Planungsverfahren.....	39
2.5.1.1 Anforderungen an die Modellierung des Planungsproblems.....	39
2.5.1.2 Anforderungen an die Simulation des Produktionsablaufs.....	42
2.5.2 Anforderungen an die Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie.....	43
2.5.2.1 Anforderungen an Planungsstrategien	43
2.5.2.2 Anforderungen an die Produktionsplanung und -steuerung	43
2.5.2.3 Anforderungen an die Produktionsplanung und -steuerung unter Unsicherheit	45
3 STAND DER TECHNIK.....	47
3.1 ANSÄTZE FÜR DAS BENCHMARKING VON PLANUNGSVERFAHREN.....	47
3.1.1 Ansätze für die Modellierung des Planungsproblems	47
3.1.2 Ansätze für die Simulation des Produktionsablaufs	56
3.2 ANSÄTZE ZUR PRODUKTIONSPLANUNG UND -STEUERUNG FÜR DIE MEHRSTUFIGE AUFTRAGSBEZOGENE SERIENFERTIGUNG IN DER AUTOMOBILZULIEFERINDUSTRIE.....	60
3.2.1 Ansätze für Planungsstrategien.....	60
3.2.2 Ansätze für die Produktionsplanung und -steuerung	66
3.2.2.1 Materialbedarfsplanung	66
3.2.2.2 Losgrößenplanung	67
3.2.2.3 Termin- und Kapazitätsplanung.....	72

3.2.2.4	Verfahren des Operations Research	74
3.2.2.5	Änderung eines Produktionsplans	76
3.2.3	Ansätze für die Produktionsplanung und -steuerung unter Unsicherheit	78
4	ZU LEISTENDE ARBEITEN	85
4.1	VERBLEIBENDE AUFGABEN FÜR DAS BENCHMARKING VON PLANUNGSVERFAHREN.....	85
4.1.1	Verbleibende Aufgaben für die Modellierung des Planungsproblems	85
4.1.2	Verbleibende Aufgaben für die Simulation des Produktionsablaufs	86
4.2	VERBLEIBENDE AUFGABEN ZUR PRODUKTIONSPLANUNG UND -STEUERUNG FÜR DIE MEHRSTUFIGE AUFTRAGSBEZOGENE SERIENFERTIGUNG IN DER AUTOMOBILZULIEFERINDUSTRIE	88
4.2.1	Verbleibende Aufgaben für Planungsstrategien.....	88
4.2.2	Verbleibende Aufgaben für die Produktionsplanung und -steuerung.....	89
4.2.3	Verbleibende Aufgaben für die Produktionsplanung und -steuerung unter Unsicherheit.....	90
5	UMSETZUNG	91
5.1	EIN BEITRAG FÜR DAS BENCHMARKING VON PLANUNGSVERFAHREN	91
5.1.1	Ein Beitrag für die Modellierung des Planungsproblems.....	91
5.1.1.1	Entwicklung des deterministischen Modells	91
5.1.1.1.1	Betrachtungshorizont.....	92
5.1.1.1.2	Parameter und Variablen.....	92
5.1.1.1.3	Kostenfunktion	94
5.1.1.1.4	Restriktionen.....	94
5.1.1.2	Entwicklung des stochastischen Modells.....	97
5.1.1.2.1	Betrachtungshorizont.....	97
5.1.1.2.2	Parameter und Variablen.....	97
5.1.1.2.3	Kostenfunktion	98
5.1.1.2.4	Restriktionen.....	99
5.1.2	Ein Beitrag für die Simulation des Produktionsablaufs.....	101
5.1.2.1	Entwicklung der Produktionsdatenbank.....	101
5.1.2.1.1	Determinanten	101
5.1.2.1.2	Funktionsprinzip	102
5.1.2.1.2.1	Kundenaufträge.....	102
5.1.2.1.2.2	Arbeitsplan	103
5.1.2.1.2.3	Produktionssystem	106
5.1.2.1.2.4	Stochastische Planungsdaten	109
5.1.2.2	Entwicklung des Simulators.....	110
5.1.2.2.1	Determinanten	110
5.1.2.2.2	Funktionsprinzip	112
5.1.2.3	Entwicklung der Schnittstelle für Planungsverfahren	114
5.1.2.3.1	Determinanten	114
5.1.2.3.2	Funktionsprinzip	115
5.2	EIN BEITRAG ZUR PRODUKTIONSPLANUNG UND -STEUERUNG FÜR DIE MEHRSTUFIGE AUFTRAGSBEZOGENE SERIENFERTIGUNG IN DER AUTOMOBILZULIEFERINDUSTRIE.....	117
5.2.1	Ein Beitrag zu Planungsstrategien	117

5.2.1.1	Strategie 1: Erstelle Lösungspopulationen	117
5.2.1.2	Strategie 2: Führe Änderungen lokal durch	117
5.2.1.3	Strategie 3: Plane und nutze die Kapazitäten.....	118
5.2.1.4	Strategie 4: Vergrößere den Planungshorizont	118
5.2.1.5	Strategie 5: Plane nicht zu fest und halte Optionen offen	119
5.2.1.6	Strategie 6: Verwende Jobprioritäten und Prioritätsreihenfolgen	119
5.2.1.7	Strategie 7: Erstelle einen Masterplan und beachte alle Restriktionen	119
5.2.1.8	Strategie 8: Gewichte die Perioden und verwende variable Auftragspakete	119
5.2.1.9	Strategie 9: Verringere die Bestände durch ein build-to-order Szenario	120
5.2.1.10	Strategie 10: Puffere Engpässe durch flexible Zwischenprodukte proaktiv.....	121
5.2.1.11	Strategie 11: Reagiere sofort auf Ereignisse und repariere den Masterplan	121
5.2.1.12	Strategie 12: Nutze bestmöglich die Potentiale: Bestand, Kapazität und Zeit	121
5.2.1.13	Strategie 13: Optimierte fortlaufend und verknüpfe Vorgänger und Nachfolger.....	122
5.2.1.14	Strategie 14: Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans.....	122
5.2.2	Ein Beitrag zur Produktionsplanung und -steuerung.....	124
5.2.2.1	Erstellung eines gültigen Produktionsplans	124
5.2.2.1.1	Kapazitätsorientierte Bedarfsplanung	126
5.2.2.1.2	Einlastung der Aufträge	130
5.2.2.1.3	Auflösung eines Planungskonflikts.....	133
5.2.2.2	Änderung eines Produktionsplans	147
5.2.2.2.1	Erhöhung der Produktionsmenge.....	149
5.2.2.2.2	Senkung der Produktionsmenge	153
5.2.2.3	Pufferung von Engpässen.....	154
5.2.2.4	Optimierung eines gültigen Produktionsplans	158
5.2.2.5	Behandlung von Ausnahmen.....	166
5.2.2.6	Verbesserung der Auslastung eines Produktionssystems	167
5.2.3	Ein Beitrag zur Produktionsplanung und -steuerung unter Unsicherheit	170
5.2.3.1	Planung der Kapazitäten.....	170
5.2.3.1.1	Lösungsraum	170
5.2.3.1.2	Spielbaum.....	171
5.2.3.1.3	Funktionsprinzip	172
5.2.3.2	Erstellung eines Masterplans.....	177
5.2.3.2.1	Lösungsraum	177
5.2.3.2.2	Spielbaum.....	177
5.2.3.2.3	Funktionsprinzip	178
5.2.3.3	Gewichtung der Perioden	183
6	FAZIT UND AUSBLICK	187
7	LITERATURVERZEICHNIS	189

Abbildungsverzeichnis

Abbildung 1: Schwachstellenmatrix	26
Abbildung 2: Benchmarkplattform	88
Abbildung 3: Klassifikation des Betrachtungshorizonts	92
Abbildung 4: Komponente Produktionsdatenbank	101
Abbildung 5: Datenbankkomponente Kundenaufträge.....	103
Abbildung 6: Datenbanktabelle Produkt.....	104
Abbildung 7: Datenbanktabelle Teilaufgabe	104
Abbildung 8: Datenbanktabelle Stueckliste	105
Abbildung 9: Datenbankkomponente Arbeitsplan.....	105
Abbildung 10: Datenbanktabelle Maschine	106
Abbildung 11: Datenbanktabelle Teilaufgabe_Maschine.....	107
Abbildung 12: Datenbanktabelle Ruestkosten	107
Abbildung 13: Datenbanktabelle Nutzungsgrad	108
Abbildung 14: Datenbankkomponente Produktionssystem.....	108
Abbildung 15: Datenbanktabelle Zufallsverteilung.....	109
Abbildung 16: Datenbanktabelle Verteilungstabelle	109
Abbildung 17: Datenbankkomponente Stochastische Planungsdaten	110
Abbildung 18: Komponente Simulator.....	111
Abbildung 19: Determinanten des Simulators	112
Abbildung 20: Algorithmus des Simulators	114
Abbildung 21: Komponente Optimierungseengine	114
Abbildung 22: Funktionsprinzip der Optimierungseengine	115
Abbildung 23: Algorithmus der Optimierungseengine	116
Abbildung 24: Algorithmus des Komponentencusters Planerstellung.....	126
Abbildung 25: Deadlinekonzept.....	128
Abbildung 26: Algorithmus der Komponente KapazitätsorientierteBedarfsermittlung ..	130
Abbildung 27: Algorithmus der Komponente Einlastungsalgorithmus	132
Abbildung 28: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Eins.....	134
Abbildung 29: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Zwei	135
Abbildung 30: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Drei	136
Abbildung 31: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Vier.....	138
Abbildung 32: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Fünf.....	139
Abbildung 33: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Sechs.....	140
Abbildung 34: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Sieben.....	141
Abbildung 35: Algorithmus der Komponente Konfliktbebungsalgorithmus - Schritt Acht	143

Abbildung 36: Algorithmus der Komponente Konfliktbelegungsalgorithmus - Schritt Neun	144
Abbildung 37: Algorithmus der Komponente Konfliktbelegungsalgorithmus - Schritt Zehn.....	146
Abbildung 38: Algorithmus des Komponentencuster Änderungsplanung	148
Abbildung 39: Algorithmus der Komponente Erhöhungsalgorithmus.....	150
Abbildung 40: Algorithmus der Komponente Erhöhungsalgorithmus - Fall Zwei	151
Abbildung 41: Algorithmus der Komponente Erhöhungsalgorithmus - Fall Drei	152
Abbildung 42: Algorithmus der Komponente Erhöhungsalgorithmus - Fall Vier	153
Abbildung 43: Algorithmus der Komponente Senkungsalgorithmus	154
Abbildung 44: Algorithmus der Komponente Pufferungsalgorithmus.....	157
Abbildung 45: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Eins..	159
Abbildung 46: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Zwei .	160
Abbildung 47: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Drei ..	161
Abbildung 48: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Vier...	162
Abbildung 49: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Fünf..	163
Abbildung 50: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Sechs.....	164
Abbildung 51: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Sieben.....	164
Abbildung 52: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Acht .	165
Abbildung 53: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Neun	166
Abbildung 54: Algorithmus der Komponente Backloggingalgorithmus.....	167
Abbildung 55: Verbesserung der Auslastung eines Produktionssystems	168
Abbildung 56: Algorithmus der Komponente Nutzungsgradoptimierung	169
Abbildung 57: Lösungsraum des Planungsproblems.....	170
Abbildung 58: Beschränkter Lösungsraum des Planungsproblems.....	171
Abbildung 59: Spielbaum zur Planung der Kapazitäten unter Unsicherheit	172
Abbildung 60: Algorithmus der Komponente Kapazitätsanalysealgorithmus	173
Abbildung 61: Algorithmus der Komponente Kapazitätsanalysealgorithmus - KapazitätsanalyseOptimierer	174
Abbildung 62: Algorithmus der Komponente Kapazitätsanalysealgorithmus - RealisierungenNatur	175
Abbildung 63: Algorithmus der Komponente Kapazitätsanalysealgorithmus - BerechneEinmalGewichteterGewinn	176
Abbildung 64: Algorithmus der Komponente Kapazitätsanalysealgorithmus - WähleKapazität.....	176
Abbildung 65: Spielbaum zur Erstellung eines Masterplans unter Unsicherheit.....	178
Abbildung 66: Algorithmus des Komponentencusters Plananalysealgorithmus	179
Abbildung 67: Algorithmus der Komponente Plananalysealgorithmus - PlananalyseOptimierer	180
Abbildung 68: Algorithmus der Komponente Plananalysealgorithmus - RealisierungenNatur	181
Abbildung 69: Algorithmus der Komponente Plananalysealgorithmus - BerechneZweifachGewichteterGewinn.....	182
Abbildung 70: Algorithmus der Komponente Plananalysealgorithmus - WählePlan	183

Abbildung 71: Periodengewichtung - Linear fallende Funktion.....	184
Abbildung 72: Periodengewichtung - Invertierte Funktion.....	184
Abbildung 73: Periodengewichtung - Exponentielle Abnahme.....	185
Abbildung 74: Periodengewichtung - Exponentialfunktion.....	185
Abbildung 75: Algorithmus der Komponente Periodengewichtung.....	186

Tabellenverzeichnis

Tabelle 1: Klassifikation des SLULSP	48
Tabelle 2: Parameter des MLCLSP	49
Tabelle 3: Klassifikation des MLCLSP	50
Tabelle 4: Parameter des PLSP-PM	52
Tabelle 5: Klassifikation des PLSP-PM	54
Tabelle 6: Entwicklung der Parameter des deterministischen Optimierungsmodells.....	93
Tabelle 7: Entwicklung der Variablen des deterministischen Optimierungsmodells	93

Definitionsverzeichnis

Definition 1: Produktion.....	27
Definition 2: Produktionssystem	28
Definition 3: Serienfertigung	29
Definition 4: Lagerhaltung.....	29
Definition 5: Lagerkosten	30
Definition 6: Produktionsplanung und -steuerung	30
Definition 7: Produktionsplanungsaufgabe.....	31
Definition 8: Produktionsplanungsverfahren	32
Definition 9: Störung.....	32
Definition 10: Robustheit.....	33
Definition 11: Heuristik	33
Definition 12: Benchmarkplattform	86
Definition 13: Auftragspaket	120
Definition 14: Komponente.....	124
Definition 15: Komponentencluster	124
Definition 16: Planungskonflikt.....	125
Definition 17: Deadline.....	127

Abkürzungsverzeichnis

Abkürzung	Bedeutung
APO	Advanced Planner and Optimizer
CLSP	Capacitated Lot Sizing Problem
CSLP	Continuous Setup Lot Sizing Problem
DL _{NF}	Deadline Nachfolger
DL _P	Deadline nach Plan
DL _{Opt}	Deadline Optimal
DLSP	Discrete Lot Sizing and Scheduling Problem
FK	Fremdschlüssel
FS	Fertigstellungstermin
ID	Identifikationsnummer
MLCLSP	Multi-Level Capacitated Lot Sizing Problem
OOPUS	Objektorientierte Plattform für die Generierung und Integration individueller Fabrikplanungs- und -steuerungssysteme.
OPT	Optimized Production Technology
OR	Operations Research
PK	Primärschlüssel
PPS	Produktionsplanung und -steuerung
PLSP	Proportional Lot Sizing and Scheduling
PLSP-PM	Proportional Lot Sizing and Scheduling with parallel machines

Symbolverzeichnis

Symbol	Bedeutung
a_{ki}	Direktbedarfskoeffizient bezüglich Produkt k und i . Der Wert ist Null falls Produkt i kein unmittelbarer Nachfolger von Produkt k ist, andernfalls gibt a_{ki} die Menge von Produkt k an, die benötigt wird, um Produkt i herzustellen.
a_{ji}	Direktbedarfskoeffizient bezüglich Periode j und i .
a_{pq}	Direktbedarfskoeffizient bezüglich Produkt p und q .
B_p	Kosten für das Backlogging einer Einheit des Produktes p für eine Periode (der Bedarf in einer bestimmten Periode kann auch noch durch Produktion in einer späteren Periode gedeckt werden).
b_{pt}	Zurückstellen (Backlogging) eines Produktes p am Ende der Periode t .
b_{jt}	verfügbare Kapazität der Ressource j in Periode t .
C_{mt}	verfügbare Kapazität der Maschine m in Periode t .
C_{it}	Maximale Produktionsmenge von Produkt P_i mit Aufgabe i während der Periode t .
D_{pt}	externer Bedarf nach Produkt p in Periode t .
d_{kt}	Primärbedarf für Produkt k in Periode t .
d_{jt}	Primärbedarf für Produkt j in Periode t .
E_{kt}	systemweiter Lagerbestand.
h_k	voller Lagerkostensatz des Produkts k (auch periodenabhängig definierbar).
h_j	voller Lagerkostensatz des Produkts j (auch periodenabhängig definierbar).
l_{j0}	Lagerbestand von Produkt j in Periode Null.
I	Menge der Aufgaben.
l_i	Menge der Aufgaben, die die gleiche Maschine wie Aufgabe i benutzen.
J	Anzahl der Produkte; Anzahl der Ressourcen ($j = 1, 2, \dots, J$) (nur MLCLSP).
K	Anzahl der Produkte bzw. Arbeitsgänge ($k = 1, 2, \dots, K$).

K_j	Indexmenge der Arbeitsgänge, die durch die Ressource j vollzogen werden.
M	Anzahl der Maschinen; maximal mögliche Losgröße (nur MLCLSP).
M_i	Maschine, die von Aufgabe i benötigt wird.
m_j	Maschine auf der Produkt j produziert wird.
N_k	Indexmenge der Nachfolger des Produkts k (direkt übergeordnete Produkte bzw. nachfolgende Arbeitsgänge).
N_j	Menge aller Maschinen, die in der Lage sind, Produkt j zu produzieren.
O_p	Kosten für das Nichterfüllen eines externen Bedarfs für eine Einheit von Produkt p .
o_{pt}	Nichterfüllen des Bedarfes nach Produkt p am Ende der Periode t .
P	Menge der Produkte.
P_i	Produkt, das von Aufgabe i produziert wird.
P_m	Menge aller Produkte die Maschine m nutzen.
p_{jm}	benötigte Kapazität, um eine Einheit von Produkt j auf Maschine m zu produzieren; p_{jm} wird auf ∞ gesetzt, falls Maschine m das Produkt j nicht produzieren kann.
p_{kt}	variable Produktionskosten für Produkt k in Periode t .
Q_i	Produktionskosten für eine Einheit von Produkt P_i mit Aufgabe i .
q_{jmt}	Produktionsmenge von Produkt j auf Maschine m in Periode t .
q_{it}	Produktionsmenge von Aufgabe i in Periode t .
R_{ij}	Anzahl der Perioden, die ausschließlich für das Rüsten von Aufgabe i zu Aufgabe j benötigt werden.
R'_{ij}	$= T_{i,j} - R_{ij}$; fraktionaler Anteil der $(R_{ij}+1)$ sten-Periode nach dem Beginn des Umrüstens von Aufgabe i auf Aufgabe j .
$S(D_{pt})$	stochastischer externer Bedarf nach Produkt p in Periode t .
$S(q_{it})$	stochastische Produktionsmenge von Aufgabe i in Periode t .
S_p	Lagerhaltungskosten, die anfallen, wenn Produkt p eine Periode im Lager gehalten wird.
s_{pt}	Lagerbestand von Produkt p am Ende der Periode t .

q_{kt}	Losgröße für Arbeitsgang k in Periode t .
S_j	Anzahl der unmittelbaren Nachfolger von Produkt j .
s_{jm}	nicht negative Rüstkosten für Produkt j auf Maschine m .
s_k	Rüstkostensatz des Produkts k (dieser kann auch periodenabhängig definiert werden).
T	Länge des Planungszeitraums in Perioden ($t = 1, 2, \dots, T$).
T_{ij}	Rüstzeit um von Aufgabe i auf Aufgabe j zu rüsten.
tb_k	Stückbearbeitungszeit für Arbeitsgang k .
tr_k	Rüstzeit für Arbeitsgang k .
v_j	Vorlaufzeit von Produkt j .
V_m	Opportunitätskosten für ungenutzte Kapazität der Maschine m .
v_{mt}	Binärvariable, die anzeigt, ob die Maschine m in Periode t leer steht.
X_{ij}	Rüstkosten, die anfallen wenn von Aufgabe i zu Aufgabe j gerüstet wird (auf Maschine $M_i = M_j$).
x_{jmt}	Binärvariable, die anzeigt, ob für Produkt j eine Rüstung auf Maschine m in Periode t erfolgt ($x_{jmt} = 1$) oder nicht ($x_{jmt} = 0$).
x_{ijt}	Binärvariable, die anzeigt, ob eine Umrüstung von Aufgabe i zu Aufgabe j am Anfang der Periode t erfolgt ($x_{ijt} = 1$) oder nicht ($x_{ijt} = 0$).
y_{jm0}	Anfangsrüstkzustand.
y_{jmt}	Binärvariable, die anzeigt, ob Maschine m für Produkt j am Ende der Periode t gerüstet ist ($y_{jmt} = 1$) oder nicht ($y_{jmt} = 0$).
y_{it}	Binärvariable, die anzeigt, ob eine Maschine M_i für Aufgabe i am Ende der Periode t gerüstet ist ($y_{it} = 1$) oder nicht ($y_{it} = 0$).
y_{kt}	Lagerbestand für Produkt k am Ende der Periode t .
z_k	Mindestvorlaufzeit eines Auftrags für Produkt k .
y_{kt}	binäre Rüstvariable für Arbeitsgang bzw. Produkt k in Periode t .
Δ_{pi}	ganzzahlige Anzahl von Perioden für den Transport von Produkt p zu Maschine M_i (für Aufgabe i).

1 Einleitung

„Heute kann ein Unternehmen nur bestehen, wenn es in der Lage ist, Qualitätssteigerungen bei gleichzeitigen Zeitreduzierungen und Kosteneinsparungen durchzuführen [...]. Ein Unternehmen ist also nur dann auf Dauer erfolgreich, wenn es gleichzeitig besser, schneller und schlanker wird.“

Hartmut Mehdorn (*1942) und Armin Töpfer (*1944)

Schon immer versuchen Unternehmen aller Branchen mit massiven Anstrengungen, ihren Gewinn zu maximieren. Um dies zu erreichen, entstanden im Laufe der Zeit, begleitet durch sich wandelnde Wettbewerbs- und Produktionsbedingungen, verschiedenste Ansätze. Die Hauptstrategie der Unternehmen in den siebziger Jahren war es, die knappen Ressourcen bestmöglich auszulasten.¹ In der folgenden Zeit verschärfte sich mit Beginn der Ölkrise und folgend durch die Globalisierung, sowie einem Wechsel vom Verkäufer- zum Käufermarkt der Wettbewerb zunehmend und die Konkurrenz wurde nicht nur für große, sondern auch für mittelständische Unternehmen weltweit unerbittlich.²

Die in dieser Arbeit betrachteten Automobilzulieferer, wie z.B. Continental, kämpfen auf einem Markt um ihr Überleben, der durch schnelle Modellwechsel, eine breite Produktpalette, kurze Lieferzeiten und sinkende Margen gekennzeichnet ist.³ Simultan zu der wachsenden Konkurrenz ergeben sich allerdings auch riesige Potentiale für neue Absatzmärkte. In Deutschland und Amerika fahren jeweils fünfzig Mal mehr Menschen ein Auto, als in den massiv wirtschaftlich aufstrebenden Staaten China und Indien.⁴ Bei einer Einwohnerzahl von zusammen über zwei Milliarden und unter der Annahme einer Angleichung des Wohlstandsniveaus in den nächsten fünfzig Jahren, fahren potentiell im Jahr 2060 eine Milliarde Autos mehr auf den Straßen Chinas und Indiens als heute.

¹ Vgl. [GLW+03]

² Vgl. [Mer02]

³ Vgl. [FTD07]

⁴ Auf tausend Einwohner kamen im Jahr 2006 in Deutschland 573 Autos, in Amerika 535 Autos, in China 12 Autos und in Indien 9 Autos [WIZ07-ol].

Im weltweiten Kampf um Marktanteile versuchen sich Unternehmen, z.B. durch ein auf den Kunden individuell abgestimmtes Produktangebot zu differenzieren. Dieser Trend galt zunächst für hochpreisige Produkte aus dem Fahrzeug- oder Maschinenbau und ist inzwischen in der Textilindustrie bei Turnschuhen angekommen. Dabei wächst die Anzahl an Varianten stark, weil, angefangen von z.B. der Lackierung eines Fahrzeugs, bis hin zu Sonderzubehör wie einer Sitzheizung, fast alles individuell konfigurierbar ist. Es genügt dabei aber nicht sich ausschließlich auf die Kosten zu konzentrieren, weil eine schnelle Belieferung für die Kunden ebenso wichtig ist. Darüber hinaus müssen Automobilzulieferer bei einem schwankenden Bedarf der Kunden, wie z.B. Volkswagen, und höchster Vernetzung aller Entscheidungen jederzeit lieferfähig sein. Hierbei besteht ein Konflikt mit den in der Produktion verursachten Kosten. Die damit an die Produktion gestellten Anforderungen sind enorm und werden zukünftig weiter steigen. Dieses Problem greife ich mit dem Ziel auf, die Wettbewerbsfähigkeit eines Unternehmens zu verbessern und leiste in meiner Arbeit einen Beitrag zum Benchmarking von Planungsverfahren und zur Produktionsplanung und -steuerung (PPS) für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie. Im Folgenden wird dazu das betrachtete Problem eingegrenzt.

Weltweite und damit rechtssystemübergreifende Unternehmensnetzwerke werden in dieser Arbeit ebenso wie verteilte und nicht deutsche Standorte nicht betrachtet. Die Produktkonstruktion und die Auswahl von Standorten werden ebenfalls ausgeschlossen, weil die Produktion für ein bestehendes Produktionssystem mit mehreren gegebenen Endprodukten optimiert wird. Dabei liegt der Fokus auf der Produktion in der mehrstufigen auftragsbezogenen Serienfertigung, wobei die Einzel- und Massenfertigung explizit ausgeschlossen sind. Beziehungen zu anderen Unternehmen im Rahmen einer Supply Chain werden nicht berücksichtigt. Es werden mehrere Produkte produziert, um einen dynamischen Bedarf zu decken. Die Personalwirtschaft wird in dieser Arbeit nicht betrachtet, infolgedessen wird davon ausgegangen, dass Mitarbeiter immer verfügbar sind. Für ungenutzte Kapazitäten fallen Opportunitätskosten an, wobei die Kapazitäten für die übernächste Woche verkleinert werden können. Anstatt einer einzelnen Maschine kann auch eine Gruppe von Maschinen bereit stehen, die potentiell alternativ zueinander die gleichen Produkte produzieren können. Die Inbetriebnahme zusätzlicher Maschinen wird dabei ebenso nicht fokussiert wie Schichtarbeitszeiten. Der Leistungsgrad einer Maschine kann nicht beeinflusst werden und die Produktionsgeschwindigkeit ist stochastisch verteilt, wobei Störungen und Maschinenausfälle auftreten können.

Ein Produkt kann nur auf einer Maschine produziert werden, wenn die benötigte Maschine auch für dieses Produkt gerüstet ist. Ein für die Produktion notwendiger Rüstvorgang blockiert die Maschine für eine bestimmte Zeit. Wird ein Produkt produziert, so fallen Produktionskosten an. Dabei sind alle Produkte Teil einer mehrstufigen Erzeugnisstruktur und können untereinander abhängig sein. Falls also

ein Produkt produziert werden soll, so muss zunächst sichergestellt sein, dass die Vorgängerprodukte vorrätig sind bzw. rechtzeitig produziert werden. Ein Zwischen- oder Endprodukt, das nicht sofort weiter verwendet oder ausgeliefert wird, wird bis zu der geplanten Verwendung gelagert und verursacht spezifische Lagerkosten, wobei die Lagerkapazität unbeschränkt ist. Transportzeiten zwischen Standorten werden nicht betrachtet, weil der Fokus auf der Produktion für einen Standort liegt und somit die Transportzeiten vernachlässigt werden können. Transportzeiten zwischen Maschinen werden ebenfalls nicht fokussiert. In der Lagerhaltung besteht das Zugangsrisiko und das Produktionsrisiko, ein Bestandsrisiko wird ausgeschlossen.

Produziert wird nicht kundenneutral, sondern kundenspezifisch, wobei ein Kunde Bestellungen aufgeben und bestehende Bestellungen ändern oder stornieren kann. Kein Kunde besitzt einen exklusiven Vorrechtsstatus, sondern alle Kunden sind gleich wichtig, wobei das Bestellverhalten unterschiedlich sein kann. Ein bestehender Kundenbedarf kann nicht abgelehnt werden und dem Kunden wird ein Termin für die Fertigstellung des Auftrags zugesagt. Es ist aber möglich, einen noch nicht zugesagten Auftrag zurückzustellen und einen späteren Liefertermin zu bestätigen. Für einen bereits verbindlich bestätigten Termin darf eine solche Zurückstellung nur im Ausnahmefall durchgeführt werden, nachdem alle anderen Versuche einer Konfliktbehebung fehlgeschlagen sind, wobei für die Zurückstellung eines Bedarfs hohe Kosten anfallen. Tertiärbedarf wird bei der Bedarfsermittlung nicht berücksichtigt. Prognosen von Produktionsgeschwindigkeiten und Kundenbestellungen werden extern vorgegeben und zyklische Erzeugnisstrukturen ausgeschlossen. Die Erstellung eines Produktionsplans erfolgt zentral und die Periodenlänge ist immer größer als die Fertigungszeit für eine Einheit eines Produktes. Nicht fokussiert werden die strategische, taktische und dezentrale Planung. Die Erlöse für ein Produkt sind fest und der Gewinn wird somit maximiert, indem die in der Produktion verursachten Kosten minimiert und alle Aufträge erfüllt werden.

Im Anschluss an die Einleitung in Kapitel 1 definiert Kapitel 2 das Problem. In Kapitel 3 erfolgt die Analyse des Stands der Technik im Hinblick auf die Lösung des Problems. Bereits vorhandene Konzepte werden analysiert, hieraus Defizite abgeleitet und anschließend als noch zu leistende Arbeiten in Kapitel 4 formalisiert. Die Konzeption der fehlenden Komponenten zur Problemlösung erfolgt in Kapitel 5. Kapitel 6 fasst die Arbeit zusammen und gibt einen Ausblick auf zukünftige Forschungsarbeiten. In Kapitel 7 wird das Literaturverzeichnis aufgeführt.

2 Problemdefinition

„Wettbewerb ist mehr und mehr eine Frage richtiger Beherrschbarkeit von Zeit. Nicht die Großen fressen die Kleinen, sondern die Schnellen überholen die Langsamen.“

Eberhard von Kuenheim (*1928)

Kapitel 2.1 verdeutlicht die Ausgangslage und in Kapitel 2.2 werden die in der Arbeit verwendeten Fachbegriffe erläutert, um Missverständnisse, die auf ein unterschiedliches Begriffsverständnis zurückzuführen sind, zu vermeiden. Kapitel 2.3 stellt die Zielsetzung der Arbeit auf und in Kapitel 2.4 wird die Strukturierung der Arbeit festgelegt. Die Anforderungen an die Problemlösung definiert Kapitel 2.5.

2.1 Ausgangslage

Unternehmen der Automobilzulieferindustrie müssen in der mehrstufigen auftragsbezogenen Serienfertigung bei einem schwankenden Bedarf der Kunden und höchster Vernetzung aller Entscheidungen jederzeit lieferfähig sein und das zu minimalen Kosten über alle Perioden. Unternehmen stehen dabei vor riesigen Herausforderungen. Viele Entscheidungen werden unter Unsicherheit getroffen, was sich nicht nur aus den gestiegenen Kundenanforderungen, sondern insbesondere durch die zunehmende Globalisierung und die wachsende Produktvielfalt ergibt.⁵

Die Produktionsplanung behandelt dabei folgende Problembereiche:⁶

- Mengenprobleme.
- Terminprobleme.
- Zuordnungsprobleme.
- Reihenfolgeprobleme.

Dabei stellt sich folgende Ausgangslage:⁷

- Zukünftige Kundenaufträge sind unbekannt.
- Bestehende Kundenaufträge ändern sich.

⁵ Vgl. [She04]

⁶ Vgl. [Rüt05]

⁷ Vgl. [Tem04]

- Maschinen produzieren mit variierender Geschwindigkeit und die Ausbringungsmenge kann nur geschätzt werden.
- Maschinen fallen aus.

Bedarfsmengen schwanken aus unterschiedlichen Gründen, z.B. ändert ein Kunde seine Auftragsmenge oder die ursprüngliche Planung beruht schlicht auf falschen Planungsdaten. Neben den Bedarfsmengen können sich auch die Verfügbarkeiten von Maschinen ändern z.B. durch eine Störung. Des Weiteren können aus dem vorherigen Planungszeitraum Unter- oder Übermengen übernommen werden. Als Folge kann die Änderung des ursprünglichen Produktionsplans oder sogar die Erstellung eines neuen Produktionsplans notwendig sein. Störungen erfordern damit potentiell eine Änderungs- oder Neuplanung, die sich wiederum auf nachgelagerte Produktionsstufen auswirkt und Planungsnervosität verursacht.⁸

Ein alter Produktionsplan ist dabei das Ergebnis einer ggf. aufwändigen initialen Planung für die alten Ausgangsdaten und stellt den Ausgangspunkt jeder Reaktion dar. Das Problem besteht darin, dass der vor einer Änderung gültige Produktionsplan mit seinen realisierten Vorgaben Zustände induziert hat, die bezogen auf die geänderten Ausgangsdaten ggf. für den weiteren Produktionsablauf suboptimal sind. Die Losgröße von gestern kann heute einfach zu groß oder zu klein sein und Vorleistungen, die gestern sinnvoll waren, müssen heute verschrottet werden.

Kein Lieferant kann sich sicher sein, dass er einen Folgeauftrag erhält und wenn die Bestellung kommt, führt dies möglicherweise zu kostenintensiven Änderungen im geplanten Produktionsablauf. Um Kunden kurzfristige Zusagen zu geben und Lieferschwierigkeiten zu vermeiden, müssen deshalb Leistungserstellungsprozesse ausgelöst werden, bevor die Bestellung des Endkunden vorliegt. Jede derartige Vorleistung ist mit Unsicherheiten und deshalb auch Risiken verbunden.

Um der diskutierten Unsicherheit zu begegnen, existieren verschiedene Möglichkeiten z.B. eine flexible Parametrisierung der Disposition hinsichtlich der Zyklizität, der Bestellumfänge, der Höhe der Bevorratung auf den verschiedenen Stufen sowie die Vorhaltung von Kapazitätsreserven.⁹ Der Bestand ist dabei eine wichtige Regelgröße. Einerseits puffern Lagerbestände Prozessstörungen, andererseits aber verdecken sie diese auch und verursachen hohe Lagerkosten. Bestandssenkungen führen zu gesenkten Lagerkosten und beherrschten Prozessen, wobei als Folge die Herstellkosten sinken und die Qualität steigt.¹⁰ Erhöhte Bestände verursachen damit nicht nur enorme Kosten, sondern verdecken auch Optimierungspotential.

⁸ Vgl. [BP79], [MD90], [HLR95], [HC96]

⁹ Vgl. [DW97]

¹⁰ Vgl. [Löd05]

Bevor produziert wird, sind die erforderlichen Produktionsfaktoren bereitzustellen z.B. Betriebsmittel, Arbeitskräfte und Betriebsstoffe.¹¹ Diese Aspekte werden in dieser Arbeit aber wie in der Einleitung abgegrenzt nicht betrachtet, weil die Produktion für ein bestehendes Produktionssystem optimiert wird.

Erhält ein Unternehmen einen Kundenauftrag, hat es sich an bestimmte Termine zu halten, zu denen der Auftrag zu Ende geführt sein muss. Wurde die Zusage eines Lieferzeitpunkts gemacht und entstehen dennoch Lieferverzögerungen, müssen ggf. hohe Konventionalstrafen gezahlt werden. Ein nicht erfüllter Bedarf entsteht dabei insbesondere durch eine falsche Planung und wird durch knappe Ressourcen, Maschinenausfälle und Bedarfsschwankungen verstärkt.

Darüber hinaus resultieren die Kurzfristigkeit jeder Festlegung und die Kürze des Horizonts mit stabilen Aussagen, in dynamischen Beziehungen, bei gleichzeitiger Vernetzung aller Entscheidungen. Ist zu wenig Kapazität vorhanden, kann der Kundenbedarf nicht gedeckt werden, wobei ungenutzte Kapazität hohe Opportunitätskosten verursacht. Eine veraltete Technik führt zu erhöhten Kosten und dazu, dass ein Bedarf evtl. nicht gedeckt werden kann.

Die diskutierten Zusammenhänge zwischen den Problemen werden verknüpft und in einer Schwachstellenmatrix verdeutlicht (vgl. Abbildung 1).

¹¹ Dabei werden die Bereitstellung von Grundstücken, Gebäuden, Maschinen und die Auswahl der Standorte langfristig geplant. Arbeitskräfte werden mittel- bis langfristig geplant, andere Betriebsmittel mittelfristig. Kurz- bis mittelfristig werden die Betriebsstoffe geplant. Kurzfristige Änderungen in der Planung können z.B. durch Überstunden ermöglicht werden [Sch92].

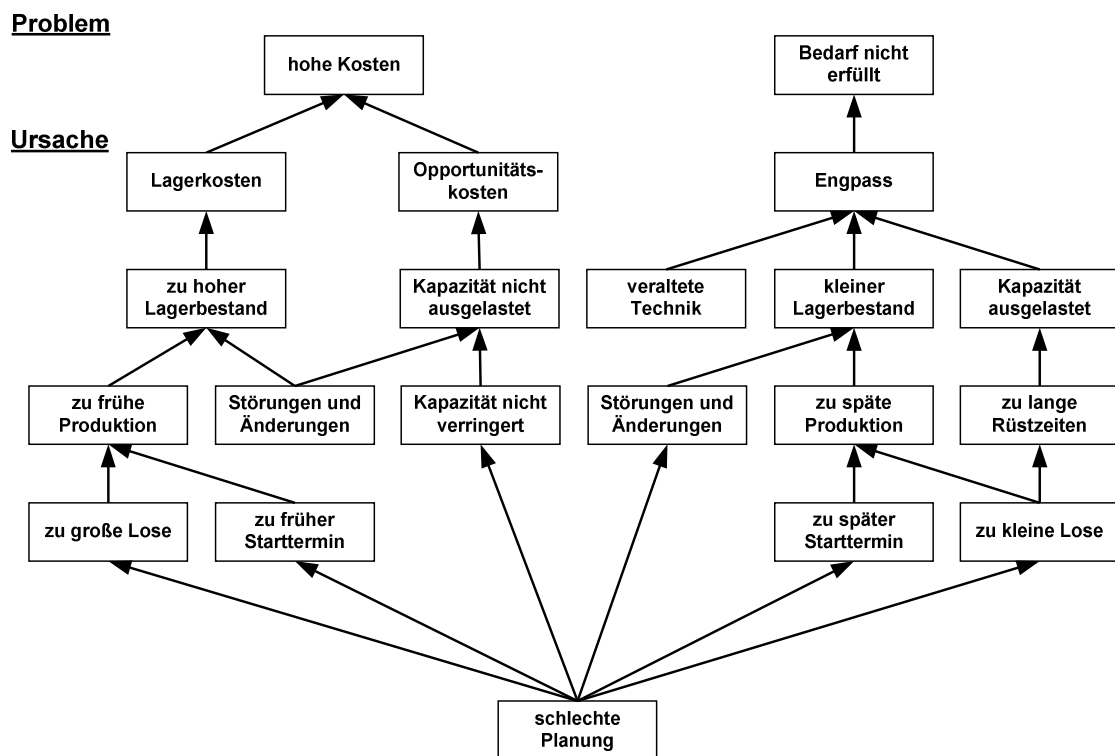


Abbildung 1: Schwachstellenmatrix

Ein guter Produktionsplan¹² ist entscheidend für die Lösung der diskutierten Probleme. Die Geschichte der Produktionsplanung ist wechselvoll und hat ihren Ursprung in den sechziger Jahren, als die Großfabriken mit tausenden von Mitarbeitern und Arbeitsgängen sowohl Meister als auch Betriebsleiter überforderte. Anhand von festen Regeln wurde versucht, den Zielkonflikt zwischen einer hohen Auslastung und pünktlicher Lieferung zu meistern. Alle diese Regeln unterstützten jeweils ein einzelnes Ziel, den Zielkonflikt lösten sie jedoch nicht. Mit den achtziger Jahren versprach der elektronische Leitstand eine zuverlässige Planung, jedoch wurde auch diese Hoffnung enttäuscht, da das zugrunde liegende deterministische Modell falsch und der Aufwand für Planung und Systempflege hoch war.¹³

Für die PPS existiert heute eine Fülle von Verfahren.¹⁴ Dabei besteht das Problem, dass sich gängige Planungsverfahren untereinander nicht objektiv vergleichen lassen, weil eine Möglichkeit für den Vergleich der Verfahren fehlt.

¹² Dieser enthält die gebündelten Informationen für die operative Produktion und löst ein Zuweisungsproblem über mehrere Zeitschritte [Pal99].

¹³ Vgl. [Löd05]

¹⁴ Vgl. [Rüt05]

Weiterhin sind nach gängiger Auffassung viele Probleme mit den bisher im Einsatz befindlichen Verfahren nur unzureichend lösbar. Die Verfahren werden zwar fortlaufend verbessert, weisen aber immer noch erhebliche Schwächen auf.

Zum einen vernachlässigen sie die Fortsetzung der Produktion über den Planungshorizont hinaus. Dies spiegelt jedoch nicht die Praxis wieder und laufend neu eingehende Kundenaufträge bewirken, dass ein auf den ursprünglichen Planungsdaten beruhender Produktionsplan jetzt potentiell suboptimal ist und erhöhte Kosten oder Lieferverzögerungen verursacht. Grund hierfür ist, dass der ursprüngliche Produktionsplan mit seinen bereits realisierten und geplanten Vorgaben Zustände induziert hat, die Ausgangspunkt für den weiteren Produktionsablauf sind.

Zum anderen wird auch nach Ende des Planungshorizonts weiter produziert und für eine Fortsetzung der Produktion über den Horizont hinaus sollten, genauso wie für zusätzliche Kundenanfragen, optimierte Reserven - beispielsweise Lagerbestände und Maschinenkapazitäten - geschaffen werden.

Mit dem Ziel, die Wettbewerbsfähigkeit eines Unternehmens zu verbessern und unter Berücksichtigung der Ausgangslage, stellt sich damit das Problem, wie Planungsverfahren für verschiedene Planungsprobleme objektiv verglichen werden können. Zweitens wird das Problem fokussiert, wie das Produktionssystem so aufgestellt werden kann, dass die Leistung bei einem schwankenden Bedarf und gleichzeitig gesenkten Kosten verbessert wird.

2.2 Fachbegriffe

Produktion (vgl. Definition 1) ist ein Transformationsprozess, der Produkte aus materiellen und nicht materiellen Einsatzgütern erstellt.

Definition 1: Produktion

„Die sich in betrieblichen Systemen vollziehende Bildung von Faktorkombinationen im Sinne einer Anwendung technischer oder konzeptioneller Verfahren zur Transformation der dem Unternehmen zur Verfügung stehenden originären und derivativen Produktionsfaktoren in absetzbare Leistungen oder in derivative Produktionsfaktoren [...], die dann in weiteren Faktortransformationsprozessen unmittelbar genutzt oder in absetzbare Leistungen transformiert werden, um das Sachziel unter der Maßgabe der Formalziele zu erfüllen.“¹⁵

¹⁵ Vgl. [Cor98]

Das Output-Gut Laptop wird unter Verwendung von Tastatur, DVD-Laufwerk und Leiterplatten, aber ebenso von Patenten, Lizenzen und Software produziert. Bei Output-Gütern, im Folgenden Produkte genannt, wird zwischen End- und Zwischenprodukten klassifiziert.¹⁶ Ein Produkt wird in einem Produktionssystem (vgl. Definition 2) hergestellt.

Definition 2: Produktionssystem

Ein Produktionssystem ist die Summe aller Arbeitsmittel in einem festgelegten Bereich, dessen Aufgabe es ist, in das System eingehende Materialien in einen definierten Ausgangszustand zu transformieren und am Ausgang abzugeben.¹⁷

Neben Maschinen, Betriebsmitteln, Mitarbeitern, Produktionsstätten usw. beinhaltet dieses auch das Regelwerk für Prozesse in der Produktion und stellt damit den Ausgangspunkt einer jeden Optimierung dar. Die Definition der Außengrenzen eines Produktionssystems ist Voraussetzung für jede Form der Planung oder Optimierung des Systems.¹⁸

Die Produktion in einem Produktionssystem wird dabei durch unterschiedliche Produktionsformen stark beeinflusst, die nach folgenden Kriterien klassifiziert sind:¹⁹

- Mechanisierungsgrad²⁰.
- Stufigkeit²¹.
- Marktbezug²².
- Repetitionstyp²³.
- Anordnungstyp²⁴.

¹⁶ Vgl. [Tem04]

¹⁷ Vgl. VDI Richtlinie 3633

¹⁸ Vgl. [BKP05]

¹⁹ Vgl. [Tch07]

²⁰ Es wird zwischen manueller, mechanischer und automatisierter Produktion unterschieden.

²¹ Es existiert die einstufige und die mehrstufige Fertigung.

²² Es wird zwischen Kundenauftragsfertigung, Lagerfertigung und Eigenfertigung differenziert.

²³ Unterscheidung zwischen Massen-, Sorten-, Serien-, und Einzelfertigung.

Problemstellung der vorliegenden Arbeit ist, wie in der Einleitung abgegrenzt, die mehrstufige auftragsbezogene Serienfertigung (vgl. Definition 3) in der Automobilzulieferindustrie.

Definition 3: Serienfertigung

Serienfertigung bezeichnet die Produktion mehrerer gleichartig zusammengesetzter Produkte. Abhängig von der hergestellten Anzahl werden Klein-, Mittel- und Großserien unterschieden. Häufig vorzufindendes Produktionsprinzip ist die Fließfertigung bei Großserien.²⁵

Während der Produktion werden auf verschiedenen Stufen Zwischenprodukte gelagert. Dieses erfüllt mehrere Funktionen, die im Folgenden erläutert werden. Durch die Ausgleichsfunktion kann für die Produktion nicht benötigtes Material gelagert werden. Ein Lager kann damit als Puffer verwendet werden, der Schwankungen, z.B. der Produktion, ausgleicht. Lagerhaltung (vgl. Definition 4) führt darüber hinaus zu Kontinuität im Sortiment und gewährleistet die Bereitstellung von Waren. Hierdurch wird die Ausgleichsfunktion ergänzt, weil eine Diskrepanz zwischen Beschaffung und Absatz überbrückt werden kann.

Des Weiteren dient das Lager zur Sicherstellung der Produktion, falls unsichere Daten über zukünftige Mengenbedarfe, Liefer- und Bedarfszeitpunkte im Unternehmen vorhanden sind.²⁶ Durch die Bestellung großer Mengen kann ggf. ein Mengenvorteil erzielt werden, allerdings können bei einem Preisverfall der Ware auch hohe Verluste verursacht werden.²⁷

Definition 4: Lagerhaltung

Lagerhaltung bezeichnet das bewusste Bevorraten von Gütern. Funktionen dieser Bevorrattung sind Ausgleich, Bereitstellung, Sicherung, Veredelung und Spekulation.²⁸

²⁴ Differenzierung zwischen Fließ- und Werkstattfertigung.

²⁵ Vgl. [BKP05]

²⁶ Vgl. [Hop05]

²⁷ Vgl. [Wöh02]

²⁸ Vgl. [Dan99]

Weiterhin kann während der Produktion eine Störung auf Produktionsstufen auftreten. Diese schlägt sich ohne Puffermöglichkeit sofort auf nachgelagerte Produktionsstufen durch. Um die Unterbrechung der Produktion lokal zu begrenzen, wird deshalb oft mit einem Sicherheitsbestand gearbeitet.²⁹ Die angesprochene Lagerhaltung verursacht hohe Kosten, entsprechend groß ist das Potential an Kosteneinsparungen, das durch einen verringerten Lagerbestand erzielt werden kann. Um die Komplexität zu verringern, werden im Rahmen dieser Arbeit fixe Lagerkosten vernachlässigt und die variablen Lagerkosten (vgl. Definition 5) fokussiert.

Definition 5: Lagerkosten

„Lagerkosten sind alle Kosten, die im Zusammenhang mit der Bevorratung von Gütern anfallen“.³⁰ Hierbei wird zwischen fixen und variablen Kosten unterschieden. Zu den Fixkosten zählen neben Miete und Abschreibungen auch die Grundsteuer. Zu den variablen Lagerkosten zählen insbesondere die Kosten für im Lagergut gebundenes Kapital.

Die gesamte technische Auftragsabwicklung und eine effektive Produktion werden durch die PPS (vgl. Definition 6) unterstützt. Dabei müssen verschiedene Restriktionen, wie die maximal zur Verfügung stehenden Ressourcen, eingehalten werden. Allerdings werden die konkreten Aufgabeninhalte in der Literatur kontrovers diskutiert.³¹

Definition 6: Produktionsplanung und -steuerung

„Die PPS hat die Aufgabe, aufgrund erwarteter und/oder vorliegender Kundenaufträge den mengenmäßigen und zeitlichen Produktionsablauf unter Beachtung der verfügbaren Ressourcen durch Planvorgaben festzulegen, diese zu veranlassen sowie zu überwachen und bei Abweichungen Maßnahmen zu ergreifen, so dass bestimmte Ziele erreicht werden.“³²

²⁹ Vgl. [DW97]

³⁰ Vgl. [Sch02]

³¹ Vgl. [Hac84], [Ker93], [Kit82], [Kur98], [Sch80], [Spe92]

³² Vgl. [Zäp98]

In der PPS eingesetzte Systeme ermitteln Lösungen für Produktionsplanungsaufgaben (vgl. Definition 7). Die Ausgangsdaten werden dabei durch Ist- und Plandaten beschrieben. Während Istdaten den Zustand der Produktion bis Heute beschreiben, definieren Plandaten Zustandsgrößen, die in der Zukunft liegen. Ist- und Plandaten bilden damit die Grundlage für neue Plandaten und sind konsistent, wenn ein gültiger Produktionsablauf existiert, der auf den Istdaten basiert.³³

Definition 7: Produktionsplanungsaufgabe

„Die Produktionsplanungsaufgabe ist die Aufgabe für ein gegebenes Produktionssystem - ausgehend von gegebenen Ausgangsdaten - Plandaten, die in sich und mit den Ausgangsdaten konsistent sind, für einen definierten, zielgerichteten Produktionsprozess festzulegen, dem Produktionsprozess vorzugeben und auf Inkonsistenzen zu prüfen. Dabei sind die gegebenen und gesuchten Daten dem Modell des Produktionsprozesses zugeordnet.“³⁴

Eine bzw. mehrere Produktionsplanungsaufgaben werden durch Produktionsplanungsverfahren (vgl. Definition 8) gelöst. Die Lösung kann mittels drei verschiedener Klassen von Verfahren erfolgen. Ein erster Lösungsweg besteht darin, alle im Rahmen des Modells zulässigen Lösungen zu berechnen, entsprechende Werte der Zielfunktion zu bestimmen und anschließend die beste Lösung auszuwählen. In jedem Fall führt dieses Verfahren zu einem eindeutigen Optimum. Allerdings ist das Planungsverfahren in der Praxis aufgrund des hohen Rechenaufwands nicht anwendbar.

Ein zweiter Lösungsweg besteht in der Möglichkeit, mathematische Methoden anzuwenden. Diese liefern ein exaktes Optimum. Einen optimalen Plan mit den Methoden der linearen und dynamischen Programmierung zu ermitteln, ist jedoch nur bei einfachen Verhältnissen mit vertretbarem Aufwand möglich. Größere Probleme können mit diesen Verfahren nicht gelöst werden.

Die dritte Möglichkeit ist die Anwendung heuristischer Verfahren, dabei ist oft die Entfernung einer berechneten Lösung vom Optimum nicht feststellbar. Die Qualität des heuristischen Verfahrens kann dabei durch Abschätzung, Erprobung oder durch den Vergleich³⁵ mit exakten Verfahren bestimmt werden.

³³ Vgl. [Hei06]

³⁴ Vgl. [Rüt05]

³⁵ Beim Vergleich muss jedoch eine Obergrenze für die Problemgröße beachtet werden.

Definition 8: Produktionsplanungsverfahren

„Ein Produktionsplanungsverfahren ist ein festgelegter oder erzeugter Ablauf von Verfahrensschritten, der die Plandaten der Produktionsplanungsaufgabe so erzeugt, dass die gestellten Anforderungen der Aufgabe gelöst werden. Anforderungen beziehen sich dabei auf Sach- und Formalziele und die Konsistenz der Lösung.“³⁶

Externe und interne Einflüsse führen zu Störungen (vgl. Definition 9) des Produktionsablaufs. Ursache für eine interne Störung ist beispielsweise der Ausfall einer Maschine. Eine externe Störung wird z.B. durch eine unvorhergesehene Erhöhung oder Verringerung eines Kundenbedarfs verursacht. Eine Störung kann zufällig oder systematisch sowie einmalig oder mehrmalig auftreten und eine Klassifizierung ist durch Zeit, Qualität und Kosten möglich.

Definition 9: Störung

Störung bezeichnet die Abweichung eines Vorgangs von seinem festgelegten oder vorausgerechneten Verlauf aufgrund einer unvorhergesehenen endogenen oder exogenen Einwirkung.³⁷

Mit dem Ziel die Lieferfähigkeit zu erhöhen und die Kosten zu senken, ist es sinnvoll eine marktwirtschaftliche Robustheit (vgl. Definition 10) gegen Störungen anzuvisieren. Dies beinhaltet einerseits effizientes Reagieren auf Störungen und andererseits von Anfang an eine vorausschauende Planung, die zukünftige Störungen berücksichtigt. Des Weiteren werden so Wechselwirkungen der Planungsvorgänge untereinander begrenzt. Werden geplante Bewegungen und Bestände in einem Zeitabschnitt bzw. in einem Zeitpunkt eingehalten, so gilt ein Plan als erfüllt.

³⁶ Vgl. [Rüt05]

³⁷ Vgl. [Wik05-ol]

Definition 10: Robustheit

Robustheit bezeichnet die Eigenschaft, dass die Arbeitsleistung eines Systems akzeptabel bleibt, wenn sich die Umstände ändern. In der Produktionsplanung bezieht es sich auf das Ausmaß in dem sich die Arbeitsleistung verschlechtert wenn Störungen auftreten.³⁸ Eine Lösung muss auch bei „leicht“ geänderten Eingabedaten wenig an Qualität verlieren bzw. „billig“ in eine neue Lösung transformiert werden können. Des Weiteren darf die Durchführbarkeit durch abweichende Istwerte nicht gefährdet werden. Hierzu sind Kapazitätsbestands-/Materialpuffer erforderlich. Dabei müssen Änderungen ohne Unterbrechung des Produktionsprozesses bzw. ohne Verletzung der Grenzen möglich sein.

Tritt eine Störung auf und der ursprüngliche Produktionsplan ist nicht mehr durchführbar, so muss ein Planungsverfahren schnell und effizient reagieren. Ist eine exakte Lösung des Problems, z.B. der Ermittlung eines Produktionsablaufs unmöglich, werden Heuristiken (vgl. Definition 11) verwendet. Dabei werden in der Regel einzelne Nebenbedingungen nicht erfüllt oder das Problem umfasst Teilaspekte, die durch die Heuristik nicht berücksichtigt werden.

Definition 11: Heuristik

Ursprünglich bedeutet Heuristik „Kunst des Finden und Erfindens“³⁹ und demzufolge ist also eine Heuristik ein Verfahren, das zu neuen Erkenntnissen führt.

Metaheuristiken sind allgemeine, im Wesentlichen nicht problemspezifische und damit generische Prinzipien und Schemata zur Entwicklung und Steuerung heuristischer Verfahren. In der Literatur werden hierunter insbesondere Verfahren verstanden, die sukzessive versuchen, verbesserte Lösungen zu finden.

Der in dieser Arbeit verfolgte Ansatz für eine Heuristik zur Produktionsplanung und -steuerung unter Unsicherheit kommt aus der Spieltheorie und liegt im Schnittfeld von Entscheidungstheorie, Operations Research, Robuster Optimierung⁴⁰ und Künstlicher

³⁸ Vgl. [Bon98]

³⁹ Vgl. [Pec99]

⁴⁰ Vgl [Sch01]

Intelligenz⁴¹. Ein mehrstufiges ganzzahliges Entscheidungsproblem wird als Spiel aufgefasst und der entstehende Spielbaum anschließend ausgewertet.⁴²

⁴¹ Vgl [RN03]

⁴² Vgl [EGL04], [Pap83], [MWG95]

2.3 Zielsetzung der Arbeit

Erstes Ziel dieser Arbeit ist es, eine Möglichkeit zu finden, wie unterschiedliche Planungsverfahren untereinander objektiv verglichen werden können und das nicht nur für ein einzelnes Planungsproblem, sondern auch für unterschiedliche Probleme. Um die Objektivität eines Vergleichs zu gewährleisten, sollen Planungsproblem, Simulation des Produktionsablaufs und Planungsverfahren voneinander entkoppelt werden.

Das in der Einleitung abgegrenzte Problem soll dazu übersichtlich und strukturiert abgebildet werden und die Möglichkeit bestehen, die gefundene Lösung für weitere Probleme zu adaptieren. Weiterhin muss nach einer Möglichkeit gesucht werden, wie die Produktion in dem betrachteten Produktionssystem simuliert werden kann. Alle für den Produktionsablauf in dem Produktionssystem relevanten Daten sollen dabei strukturiert gespeichert werden. Aus einer Menge von Testinstanzen soll ein Planungsproblem gewählt und ein Planungsverfahren gestartet werden können. Dabei wird die Entwicklung eines Planungsverfahrens hier nicht fokussiert, sondern die Konzentration liegt darauf, die Möglichkeit zu schaffen ein Planungsverfahren an die Simulation des Produktionsablaufs anzukoppeln.

Um systematisch eine Lösung für das Problem zu erarbeiten, gilt es zwei Teilprobleme zu lösen, die in Kapitel 2.3.1 detailliert analysiert werden.

Zweites Ziel dieser Arbeit ist es, ein Produktionssystem so aufzustellen, dass die Leistung unter einem schwankenden Bedarf bei gleichzeitig gesenkten Kosten verbessert wird.

Neuartige Planungsansätze sollen einen Beitrag dazu leisten, das Produktionssystem so aufzustellen, dass das Unternehmen möglichst immer lieferfähig ist und möglichst geringe Gesamtkosten über alle Perioden entstehen. Die Planerstellung soll dabei schnell und zuverlässig erfolgen und robuste Produktionspläne erstellen. Weiterhin muss auf Unsicherheiten durch Bedarfsschwankungen und Schwankungen in der Produktionsmenge laufend reagiert werden. Dabei soll eine Verbesserung zu den heute in der PPS eingesetzten Ansätzen geboten werden.

Um systematisch eine Lösung für das Problem zu erarbeiten, gilt es drei Teilprobleme zu lösen, die in Kapitel 2.3.2 detailliert debattiert werden.

2.3.1 Zielsetzung für das Benchmarking von Planungsverfahren

2.3.1.1 Zielsetzung für die Modellierung des Planungsproblems

Für das Benchmarking von Planungsverfahren ist es zuerst erforderlich, eine Modellierungsmethode für das Planungsproblem zu finden. Die Aufgabe besteht darin eine solche Modellierungsmethode zu wählen, die das eingegrenzte Produktionssystem formalisiert und die Rahmenbedingungen des Produktionsablaufs festlegt. Ein Planungsproblem in der Produktion lässt sich in Form eines mathematischen Modells

abbilden.⁴³ Die Restriktionen sowie die Zielfunktion sollen deshalb in Form eines Mixed Integer Programms abgebildet werden.

Modelle in dieser Formalisierung können mit verschiedenen Algorithmen und Heuristiken gelöst werden, was hier jedoch nicht weiter verfolgt wird. Stattdessen werden die Optimierungsmodelle auf ihre Fähigkeit hin untersucht, das in dieser Arbeit betrachtete Planungsproblem abzubilden und gleichzeitig erweiterbar für weitere Planungsprobleme in der Produktion zu bleiben. Dabei stellt die gefundene Lösung den Ausgangspunkt für die Simulation des Produktionsablaufs dar, indem festgelegt wird, welche Kosten und Restriktionen in dem Planungsproblem fokussiert werden.

Der Stand der Technik wird deshalb nach vorhandenen Ansätzen zur Modellierung eines Planungsproblems durchsucht.

2.3.1.2 Zielsetzung für die Simulation des Produktionsablaufs

Für die Simulation des Produktionsablaufs soll nach einer Lösung gesucht werden, die das modellierte Planungsproblem implementiert und die Umsetzung eines Produktionsplans ermöglicht.

Über eine Schnittstelle soll ein Planungsverfahren an die Simulation des Produktionsablaufs angekoppelt werden können. Eine Lösung soll die Möglichkeit schaffen sich über eine grafische Oberfläche anzumelden und aus einer Liste von Planungsproblemen ein Problem zu wählen, für das die Produktion simuliert wird.

Stamm- und Bewegungsdaten müssen gespeichert werden, wobei es hierzu gilt, eine geeignete Datenbank zu finden, die es ermöglicht, alle Stamm- und Bewegungsdaten zu archivieren und dabei erweiterbar für weitere Problemstellungen zu bleiben.

Während der Umsetzung eines Produktionsplans müssen laufend die Einhaltung der Produktionsrestriktionen überprüft werden und am Ende der Simulation die entstandenen Ist-Kosten berechnet werden.

Vorhandene Lösungsansätze werden dazu im Stand der Technik analysiert.

⁴³ Vgl. [Alt06]

2.3.2 Zielsetzung für die Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie

2.3.2.1 Zielsetzung für Planungsstrategien

Um das Planungssystem bestmöglich aufzustellen ist es zuerst erforderlich Planungsstrategien aufzustellen, weil diese die Grundlage für einen Planungsansatz und damit auch die Grundlage für die Erstellung eines Produktionsplans bilden.

Vorhandene Ansätze zu Planungsstrategien werden dazu im Stand der Technik analysiert.

2.3.2.2 Zielsetzung für die Produktionsplanung und -steuerung

Ziel ist es, das Produktionssystem so aufzustellen, dass die Gesamtkosten über alle Perioden verringert werden und die Lieferfähigkeit verbessert wird. Im Gegensatz zum mathematischen Ansatz zur Problemlösung orientieren sich ingenieurwissenschaftliche Ansätze an der pragmatischen Lösungsfindung direkt in der Praxis. Dabei wird das gesamte Problem betrachtet und nicht nur ein Abbild, allerdings ist eine Aussage über die Optimalität der ermittelten Lösung oftmals schwierig. Für das in dieser Arbeit zu entwickelnde Planungsverfahren eignet sich damit eine Kombination aus mathematischem und ingenieurwissenschaftlichem Ansatz, das die Vorteile der beiden Ansätze vereint.

Gängige Vorgehensweisen versuchen den Materialbedarf nach Menge und Termin so genau wie möglich zu bestimmen, was es auch für das in dieser Arbeit betrachtete Problem zu erreichen gilt. Eine zu frühe Bereitstellung von Material verursacht dabei überflüssige Lagerkosten und andersherum führt eine zu späte Bereitstellung zu Lieferschwierigkeiten.⁴⁴ Ziel ist es deshalb, einen Auftrag so spät wie möglich zu fertigen, um keine unnötigen Lagerkosten zu verursachen, aber dabei zu beachten möglichst immer lieferfähig zu bleiben.

Für die PPS soll dabei eine Lösung gefunden werden, um aus den Stamm- und Bewegungsdaten den Materialbedarf abzuleiten, ggf. Lose zu bilden und die Aufträge einzulasten. Dabei soll das Planungsverfahren auch auf laufend eingehende Änderungen diversifiziert effizient reagieren und einen bestehenden Produktionsplan laufend verbessern.

Der Stand der Technik wird hierzu nach vorhandenen Ansätzen durchsucht.

⁴⁴ Vgl. [Tem04]

2.3.2.3 Zielsetzung für die Produktionsplanung und -steuerung unter Unsicherheit

Der Erfolg eines Verfahrens ist neben der gewählten Strategie auch von zufällig auftretenden Ereignissen abhängig. Deshalb muss der erstellte Produktionsplan nicht nur für statische Werte optimal sein, sondern berücksichtigen, dass der Horizont fortschreitet und sich die Ausgangsdaten, für die ein Produktionsplan erstellt wird, z.B. durch eine Störung, ändern können.

Stochastische Probleme sind nicht immer schwieriger zu lösen, als ihre deterministischen Varianten.⁴⁵ Der in dieser Arbeit verfolgte Ansatz führt dabei zu dem Forschungsgebiet „Mehrstufige Entscheidungen unter Unsicherheit“, das ein Teilgebiet des größeren Feldes von entscheidungstheoretischen Ansätzen ist.⁴⁶

Um möglichst immer lieferfähig zu sein und möglichst niedrige Gesamtkosten zu verursachen, gilt es deshalb einen robusten Produktionsplan zu erzeugen der Optionen offen hält, um auf zukünftige Änderungen zu reagieren. Die Erstellung des Produktionsplans verlangt deshalb auf potentielle Störungen abgestimmte Puffer in Form von Produktionskapazitäten und Lagerbeständen, sowie deren ständige Anpassung.

Vorhandene Ansätze zur PPS unter Unsicherheit werden dazu im Stand der Technik analysiert.

2.4 Strukturierung der Problemstellung

Die Arbeit wird in zwei Teile gegliedert. Teil Eins fokussiert auf das Benchmarking von Planungsverfahren mit dem Ziel, eine Möglichkeit dafür zu schaffen, Planungsverfahren zu vergleichen. Dabei gilt es erstens den Stand der Technik nach Ansätzen zur Modellierung eines Planungsproblems zu debattieren und zweitens eine Möglichkeit zu finden, den Produktionsablauf zu simulieren.

Teil Zwei der Arbeit fokussiert auf die PPS in der mehrstufigen auftragsbezogenen Serienfertigung für die Automobilzulieferindustrie mit dem Ziel, ein Produktionssystem bestmöglich aufzustellen. Hierzu gilt es erstens den Stand der Technik nach vorhandenen Planungsstrategien zu analysieren, zweitens den Stand der Technik zu Lösungen für die PPS zu debattieren und drittens, den Stand der Technik zur PPS unter Unsicherheit zu analysieren.

Die konzipierte Strukturierung dient als durchgängige Gliederung für die gesamte Arbeit.

⁴⁵ Vgl. [SS04], [MSU99]

⁴⁶ Vgl. [HBH88]

2.5 Anforderungen an die Problemlösung

Die Anforderungen an eine Problemlösung sind anhand der in Kapitel 2.4 vorgenommenen Strukturierung gegliedert und werden im Folgenden definiert, um den Stand der Technik gezielt im Hinblick auf die Lösung der Aufgaben zu analysieren.

2.5.1 Anforderungen an das Benchmarking von Planungsverfahren

2.5.1.1 Anforderungen an die Modellierung des Planungsproblems

Das anvisierte Ziel der Planung wird in der Zielfunktion definiert und die einzelnen Komponenten gewichtet. Für das in dieser Arbeit betrachtete Problem ist die Formulierung mit dem Fokus auf der Kostenminimierung geeignet. Kostenminimierung sollte in dem betrachteten Problem eine Gegenüberstellung von Lagerhaltungs-⁴⁷, Rüst-⁴⁸, Produktions-⁴⁹, Zurückstellungs-⁵⁰, Cancel-⁵¹ und Opportunitätskosten⁵² beinhalten.

Sinnvoll erscheint für die Berechnung der entstandenen Ist-Kosten eine produktweise Zuordnung der Lagerkosten. Für das betrachtete Problem sollten des Weiteren in der Zielfunktion die direkten Rüstkosten erfasst werden, aber nicht die Opportunitätskosten. Grund hierfür ist, wenn freie Kapazität ohnehin verfügbar ist, diese auch gut für einen Rüstvorgang genutzt werden kann und keine zusätzlichen Opportunitätskosten anfallen. Für die Modellierung des in dieser Arbeit betrachteten Problems sollten die Produktionskosten sowie die Kosten für Zurückstellen und Canceln produktweise zugeordnet werden. Außerdem sollte eine gefundene Lösung es ermöglichen, Opportunitätskosten für freie Kapazität in Abhängigkeit von der Maschine zu erfassen.

Eine einmal eingerichtete Maschine kann in der nächsten Periode im gleichen „Setup“ weiter verwendet werden, so dass keine Rüstkosten entstehen. Deshalb sollte das Optimierungsmodell für das in dieser Arbeit betrachtete Problem ein Übernehmen des

⁴⁷ Lagerkosten können fix, produktabhängig oder periodenabhängig modelliert werden.

⁴⁸ Rüstkosten entstehen, wenn ein Produktionsschritt die Einrichtung einer Maschine erfordert, wobei zwischen direkten Kosten und Opportunitätskosten klassifiziert wird. Dabei übertreffen die Opportunitätskosten oftmals die direkten Kosten [Alt06].

⁴⁹ Produktionskosten stellen dabei summiert die Material-, Personal- und Hilfsmittelkosten dar, die einem einzelnen Produkt zugeordnet werden können.

⁵⁰ Im Falle einer Nichterfüllbarkeit eines Bedarfs, weil z.B. die Zeit nicht ausreicht und Arbeitsrückstand entsteht, kann der Auftrag zurückgestellt werden.

⁵¹ Im absoluten Notfall kann ein Auftrag, um wieder lieferfähig zu sein, gecancelt werden.

⁵² Opportunitätskosten fallen, wie in der Einleitung abgegrenzt, für ungenutzte Maschinenkapazitäten an.

Rüstzustands in die nächste Periode ermöglichen. Reihenfolgeabhängige Rüstkosten sind vom vorherigen Zustand abhängig, erhöhen jedoch die Komplexität des Problems. Trotzdem sollte es das Modell unterstützen, auch reihenfolgeabhängige Rüstzeiten abzubilden.

In Bezug auf die Auslastung ist es wichtig, ob eine fraktionale Nutzung einer Periode unterstützt wird. In einfacheren Versionen von Rüstkosten wird lediglich durch eine binäre Zustandsvariable modelliert, dass Rüstkosten innerhalb einer Periode auftreten. Damit kann allerdings nicht zweimal innerhalb einer Periode gerüstet werden und ein Rüstvorgang beansprucht somit eine vollständige Periode.⁵³ Aus diesen Gründen sollte das Optimierungsmodell auch eine fraktionale Nutzung der freien Perioden für die Produktion unterstützen.

Die Nachfrage nach einem Zwischenprodukt wird in der Form eines Direktbedarfskoeffizienten a_{ki} beschrieben und gibt an, wie viele Produkte k in das übergeordnete Produkt i eingehen. Zum anderen kann noch ein Primärbedarf d_{kt} modelliert werden, welcher angibt, wie viel von einem Produkt direkt von einem Nachfrager bestellt wird. Dieser Wert ist bei Zwischenprodukten im Normalfall gleich 0, kann jedoch durchaus einen positiven Wert annehmen, wenn es sich dabei um ein Ersatzteil handelt. Eine Lösung muss es also ermöglichen, einen Primärbedarf für jedes Produkt abzubilden.

Der Horizont für die Planung kann endlich oder unendlich sein. Dabei kann die Modellzeit kontinuierlich oder diskret abgebildet werden.⁵⁴ Wird eine diskrete Zeitskala gewählt, können einzelne Zeitaggregationen unterschiedlich lang ausfallen und es gehen potentiell Informationen über die Reihenfolge des Produktionsablaufs innerhalb einer Gruppierung verloren. Für das in dieser Arbeit betrachtete Problem eignet sich aus Gründen der einfacheren Modellierbarkeit trotzdem ein diskretes Zeitmodell besser.

Optimierungsmodelle können Probleme mit deterministischem oder stochastischem Charakter abbilden. In deterministischen Modellen sind alle Daten bereits im Voraus bekannt, während in stochastischen Modellen die Daten auf Wahrscheinlichkeiten basieren.⁵⁵ Der Kundenbedarf kann statisch oder dynamisch modelliert werden, wobei statische Modelle voraussetzen, dass der Bedarf eines Kunden über einen Zeitraum gleich bleibt. Dynamische Modelle hingegen erlauben einen schwankenden Bedarf.⁵⁶ Aus der Einleitung ergibt sich, dass Kundenbedarfe und Produktionsgeschwindigkeiten

⁵³ Vgl. [Alt06]

⁵⁴ Eine kontinuierliche Betrachtung von Zeit ist realitätsnah, aber auch kompliziert. Hierbei ist die Größe einer Zeitspanne nicht festgelegt.

⁵⁵ Vgl. [VW06]

⁵⁶ Vgl. [VW06]

Schwankungen unterliegen und damit ist es erforderlich, ein stochastisches dynamisches Modell für die Abbildung des Problems zu nutzen.

Ein Produktionsprozess wird in Einprodukt- und Mehrproduktprobleme unterteilt. Dabei ergibt sich aus der Einleitung, dass Mehrproduktprobleme abbildet werden müssen. Im Fall einer Mehrprodukt-Betrachtung können die Produkte um verfügbare Maschinen konkurrieren.⁵⁷ Die Aufbaustruktur der Produkte bestimmt den Produktionsprozess erheblich, wobei zwischen einstufigen und mehrstufigen Produktionsprozessen unterschieden wird. Aus der Einleitung ergibt sich die Anforderung nach einer mehrstufigen Produktion.

Vorlaufzeiten entscheiden darüber, ob ein Zwischenprodukt nach seiner Bearbeitung an einer Maschine direkt zur Weiterverarbeitung verfügbar ist und sollten bei der Modellierung des Planungsproblems betrachtet werden.

Sind mehrere Maschinen vorhanden, wird differenziert, ob ein Produkt nur auf einer bestimmten Maschine produziert werden kann „Multi-Machines“ oder auf parallelen Maschinen. Bei parallelen Maschinen muss zusätzlich eine Maschine gewählt werden. Wie in der Einleitung abgegrenzt, muss das Optimierungsmodell potentiell auch parallele Maschinen abbilden.

Es wird weiterhin klassifiziert zwischen einem begrenzten und einem unbegrenzten Lager. Eine Begrenzung der Lagerkapazität hat signifikante Auswirkungen auf die Lösbarkeit des Modells. Aus der Einleitung ergibt sich, dass ein unbegrenztes Lager betrachtet wird, was die Lösbarkeit erleichtert. Zusätzlich zur Kapazitätsbegrenzung eines Lagers werden oft Sicherheitsbestände definiert.⁵⁸ Dieses ist aber Aufgabe des Planungsverfahrens und ist damit für die Modellierung des Problems irrelevant.

Die Reihenfolge in der die Produktion und die Rüstvorgänge ablaufen, beeinflussen diese stark. Fraktionierte Rüstzeiten ermöglichen dabei die Teilung von Rüstvorgängen und somit die Nutzung halber Perioden für die Produktion. Dies ist erstrebenswert für die Modellierung des betrachteten Planungsproblems, damit Planungsverfahren auch die gesamte Zeit vollständig nutzen können.

Kapazitäten von Maschinen können begrenzt oder unbegrenzt zur Verfügung stehen.⁵⁹ Aus der Einleitung ergibt sich, dass die Kapazitäten begrenzt sind und damit auch auf diese Weise modelliert werden müssen.

Eine gefundene Lösung muss weiterhin erweiterbar sein, um mit wenig Aufwand weitere Probleme abzubilden.

⁵⁷ Vgl. [Alt06]

⁵⁸ Vgl. [DW97]

⁵⁹ Die Betrachtung von Kapazitäten erhöht allerdings den Rechenaufwand [Alt06].

2.5.1.2 Anforderungen an die Simulation des Produktionsablaufs

Als Kernpunkte von Benchmarking werden in der Literatur vier Elemente diskutiert⁶⁰:

- Systematischer Leistungsvergleich auf Basis objektiver Leistungskriterien.
- Beurteilung der Stärken und Schwächen aufgrund des Benchmarks.
- Identifikation der Best Practices.
- Formulierung von Zielen, die zur nachhaltigen Leistungssteigerung führen.

Dieser Ansatz wird jedoch nicht weiter verfolgt und darauf fokussiert, eine Möglichkeit für den Vergleich von Planungsverfahren zu entwickeln ohne den Anspruch darauf zu erheben, aus den Ergebnissen zu lernen. Dabei sollen verschiedene Planungsverfahren objektiv in einer realistischen Umgebung für mehrere Probleminstanzen getestet werden können.

Um die Objektivität eines Benchmarks herzustellen, muss die Lösung eine Drei-Teilung in Planungsproblem, Simulation des Produktionsablaufs und Planungsverfahren ermöglichen.

Das Planungsproblem wird dabei von einem Unternehmen definiert und auf geeignete Weise formalisiert.⁶¹ Anschließend wird der Produktionsablauf für das spezifizierte Planungsproblem simuliert. Es soll möglich sein, sich über eine grafische Oberfläche anzumelden und eine Planungsproblem Instanz zu wählen, für das ein externes Planungsverfahren ausgewählt werden kann, das einen Produktionsplan berechnet. Dabei müssen parallel mehrere Produktionsläufe unabhängig voneinander gestartet werden können.

Während der Produktion sollen Störungen im Produktionsablauf auftreten und laufend neue Kundenbestellungen eingehen, auf die das Planungsverfahren reagieren kann.

Für die Problemstellung muss weiterhin ein Datenmodell spezifiziert werden, das den Sachverhalt zielorientiert abbildet.

Eine Problemlösung muss dabei potentiell für verschiedene Planungsprobleme und Planungsverfahren genutzt werden können und damit potentiell m:n Beziehungen⁶² abbilden.

⁶⁰ Vgl. [Leg99]

⁶¹ Vgl. Kapitel 2.3.1.1

⁶² Für m Probleme sollen n Verfahren analysiert werden können.

Weiterhin muss eine Lösung dafür geeignet sein, verschiedene Planungsvorschläge kostengünstig, schnell und praxisäquivalent unter fairen Bedingungen für alle Verfahren zu vergleichen.

2.5.2 Anforderungen an die Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie

2.5.2.1 Anforderungen an Planungsstrategien

Planungsstrategien müssen es ermöglichen, das Planungssystem so aufzustellen, dass das Unternehmen möglichst immer lieferfähig ist und möglichst geringe Gesamtkosten über alle Perioden entstehen. Dafür sollen minimale Bestände fokussiert werden und Kapazitäten geplant werden.

Eine Problemlösung muss dabei für die PPS in der mehrstufigen auftragsbezogenen Serienfertigung in der Automobilzulieferindustrie geeignet und hilfreich sein, um einen möglichst optimalen Produktionsplan zu erstellen.

2.5.2.2 Anforderungen an die Produktionsplanung und -steuerung

Eine Problemlösung muss hilfreich sein, das in der Einleitung abgegrenzte Produktionssystem bestmöglich aufzustellen. Klassische Planungskonzepte arbeiten dazu nach dem Push-Prinzip und zerlegen die PPS in Teilprobleme, die sie phasenbezogen sukzessiv lösen.⁶³ In derartigen Ansätzen wird mit einer Zweiteilung in Mengen- und Termin-/Kapazitätsplanung gearbeitet.⁶⁴ Die systemimmanente Schwäche besteht darin, dass die begrenzte Verfügbarkeit von Ressourcen bei der Erstellung eines Produktionsplans nicht berücksichtigt wird. Es wird erst nach der Planerstellung geprüft, ob der erstellte Produktionsplan auch gültig ist, oder sogar ganz auf diese Prüfung verzichtet. Hohe Bestände, Warteschlangen und Verspätungen sind die Folge.⁶⁵

Neue Konzepte verwenden eine Simultanplanung unter gleichzeitiger Berücksichtigung aller Restriktionen z.B. Kapazitätsrestriktionen von Maschinen. Damit können vollständige Material-, Finanz-, und Informationsflüsse betrachtet werden. Dieses Verfahren wird in dem Advanced Planner and Optimizer⁶⁶ (APO), der SAP AG seit 1998 sowie in dem am Lehrstuhl Dangelmaier entwickelten Planungsverfahren

⁶³ Vgl. [Tem04]

⁶⁴ Vgl. [Wöh02]

⁶⁵ Vgl. [Tem04]

⁶⁶ Vgl. [KMZ00]

OOPUS⁶⁷ (Objektorientierte Plattform für die Generierung und Integration individueller Fabrikplanungs- und -steuerungssysteme) seit 2000 eingesetzt. Das neue Planungsverfahren sollte aus diesen Gründen ebenfalls ein Simultanplanungskonzept nutzen, das die Anforderungen erfüllt und die Kapazitäten bei der Bedarfsplanung berücksichtigt.

Die Planerstellung muss dazu unter Berücksichtigung der verfügbaren Kapazitäten die Bedarfe auf die Maschinen einplanen. Dazu gilt es zu fokussieren, dass immer zuerst ein bereits angenommener Kundenauftrag erfüllt werden soll und anschließend die Kosten im Produktionsablauf minimiert werden.

Aus der Einleitung ergibt sich, dass die mehrstufige auftragsbezogene Serienfertigung betrachtet wird. Die Materialbedarfsplanung verbrauchsorientiert durchzuführen ist deshalb ungeeignet.⁶⁸ Sollen die Bestände minimiert werden, ist die bedarfsorientierte Vorgehensweise vorzuziehen. Hiermit können besser die Kapazitätskonkurrenz vermieden, Nichtverfügbarkeiten verringert und Verschrottungen verhindert werden.⁶⁹ Deshalb soll die Materialbedarfsplanung bedarfsorientiert durchgeführt werden.

Die Fertigungsmenge zwischen zwei Rüstvorgängen wird als Losgröße definiert.⁷⁰ Während die Logistik kleine Losgrößen favorisiert, werden in der Produktion Mindestlosgrößen bevorzugt. Bei der Entscheidung für große Lose sind die Lagerkosten höher, weil die Bedarfsmengen zukünftiger Perioden vorher produziert werden und anschließend gelagert werden, was die Lagerkosten potentiell unnötig erhöht.

Mit dem Ziel die Opportunitätskosten für ungenutzte Maschinen zu verringern, muss eine Methode dafür gefunden werden, die Auslastung nicht mehr anpassbarer Kapazitäten zu erhöhen.

Die Lagerkosten werden dabei auf der Grundlage der gesamten Kosten aller vorhergehenden Produktionsstufen berechnet.⁷¹ Parallel reduzieren sich Rüstvorgänge und Rüstzeiten. Umgekehrt verursachen kleine Lose einerseits hohe Rüstkosten, da die Anzahl der Rüstvorgänge steigt, andererseits aber geringere Lagerkosten. Die Losgröße wird weiterhin von der Beschaffung beeinflusst, indem z.B. Mindestabnah-

⁶⁷ Vgl. [DLM+98], [DHL+97], [DHH+97], [DHH+97a]

⁶⁸ Die verbrauchsorientierte Vorgehensweise ist zwar mit wenig Aufwand verbunden, aber eignet sich nur, wenn die Produkte einen niedrigen Wert besitzen, der Verbrauch kontinuierlich ist und Kapazitätskonflikte durch ausreichende Kapazität vermieden werden können.

⁶⁹ Vgl. [Gre88]

⁷⁰ Vgl. [Dan03]

⁷¹ Vgl. [Tem06]

memengen berücksichtigt werden.⁷² Die Beschaffung wird in dieser Arbeit jedoch nicht fokussiert. Bestände sollen dabei auf ein Minimum reduziert werden und Lose sollen deshalb nur gebildet werden, wenn eine knappe Kapazität bzw. fehlende Lagerbestände erfordern, dass Rüstzeit gespart wird.

Des Weiteren muss das Planungsverfahren eine geeignete Methode anbieten, um kritische Produkte proaktiv zu puffern und damit auf Störungen vorbereitet zu sein.

Das Planungsverfahren soll weiterhin berücksichtigen, dass Maschinen in Leerzeiten für Rüstvorgänge genutzt werden können und ein Rüstvorgang damit nur teuer ist, wenn die Kapazität der Maschinen ausgelastet ist oder in der Zukunft verringert werden kann. In einer simultanen Termin-/Kapazitätsplanung soll das genaue Zeitgerüst des Produktionsablaufs unter Berücksichtigung der Kapazität ermittelt werden.

Ist das Unternehmen durch eine extreme Situation nicht lieferfähig, muss das Planungsverfahren entscheiden, welche Aufträge zurückgestellt werden.

Änderungen des Produktionsplans können zu einer Fülle weiterer Änderungen führen und beeinflussen dabei in großem Ausmaß die Werker und Planer vor Ort sowie die Kunden und Lieferanten. Teilweise müssen einmal gemachte Vorgaben auch schon deshalb beibehalten werden, weil die Vorbereitungen zu ihrer Umsetzung, vor allem auf vorgelagerten Produktionsstufen, bereits schon so weit fortgeschritten sind, dass eine Änderung nicht mehr durchführbar ist. Ziel der Änderungsplanung soll es deshalb sein, den alten Produktionsplan nicht nur als Ausgangspunkt, sondern insbesondere auch als anzustrebenden Zielpunkt zu betrachten, weil die ursprüngliche Planung ggf. schon zu Vorleistungen im realen System geführt hat. Die Vernetzung des Produktionsablaufs soll dabei abgebildet werden und Möglichkeiten aufgezeigt werden, unter Berücksichtigung der Abhängigkeiten im Produktionsablauf, effektiv und diversifiziert zu reagieren.

Während der Ausführung eines Produktionsplans soll das Planungsverfahren laufend nach Verbesserungspotential suchen und ggf. mit einer Änderung reagieren.

2.5.2.3 Anforderungen an die Produktionsplanung und -steuerung unter Unsicherheit

Probleme in der PPS können oftmals selbst ohne die Berücksichtigung von Unsicherheiten in den Daten nur durch Heuristiken gelöst werden.⁷³ Dabei gelten

⁷² Vgl. [Hop05]

⁷³ Vgl. [VW06]

stochastische Probleme als PSPACE-schwer.⁷⁴ Grund hierfür ist, dass stochastische Probleme viele Varianten betrachten müssen und die Komplexität stark steigt.

Eine gefundene Lösung zur PPS unter Unsicherheit muss dazu beitragen, dass in der Einleitung abgegrenzte Produktionssystem bestmöglich aufzustellen. Dabei muss beachtet werden, dass ein gewählter Produktionsplan Optionen offen halten muss, um auf zukünftige Änderungen effektiv zu reagieren und deshalb verschiedene Produktionspläne analysiert werden sollten.

Opportunitätskosten für ungenutzte Maschinenkapazitäten sollen dabei verringert werden, indem Kapazitäten ggf. in der Zukunft verkleinert werden. Dafür ist es erforderlich, die Kapazitäten vorausschauend zu planen.

⁷⁴ Vgl. [Pap83]

3 Stand der Technik

„Die Industriegesellschaft ist einem stetigen Wandel unterworfen. Die Herausforderungen an sie werden komplexer. Das äußert sich im verstärkten internationalen Wettbewerb, aber auch in dem Bestreben der Gesellschaft, das Erreichte zu sichern. Adäquate Problemlösungen sind daher in zunehmender Weise nur fachübergreifend realisierbar.“

Wilhelm Dangelmaier (*1949)

Auf Grundlage der Strukturierung der Problemstellung aus Kapitel 2.4 wird der Stand der Technik nach geeigneten Lösungsansätzen durchsucht und diese anhand der in Kapitel 2.5 aufgestellten Anforderungen bewertet.

In Kapitel 3.1 wird der Stand der Technik zu dem Benchmarking von Planungsverfahren für die beiden Teilprobleme untersucht. Kapitel 3.2 diskutiert den Stand der Technik zur PPS für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie für die drei identifizierten Teilprobleme.

3.1 Ansätze für das Benchmarking von Planungsverfahren

3.1.1 Ansätze für die Modellierung des Planungsproblems

Altemeier klassifiziert auftretende Probleme hinsichtlich vier verschiedener Kriterien: Zielsetzung, Problemkontext, Produktion und Ressourcen.⁷⁵ In Tabelle 1 ist zu sehen, welche Art von Problemen z.B. das Single Level Uncapacitated Lot Sizing Problem (SLULSP) abbildet.

⁷⁵ Vgl. [Alt06]

Model	Objective								Env.	Production				Resources														
SLULSP	Objective (C/U/O)	Production Costs	Setup Costs	Setup Carry Over	Sequence Depend. SC	Stocking Costs	Echelon Stock	Backlog/Stockout	Terminated Horizon	BigBucket/SmallBucket	Static/Dynamic	Determin./Stochastic	Single-/Multiproduct	Multi-stage	Lead-Times	Handover batch-wise	Production Bounds	Multi Maschines	Parallel Maschines	Capacitated Stock	Production cap. C/CC/U	Setup Times	Sequence Depend. ST	Fract. Setup Times	Minimal On-/Off Times	Cleaning Times	Batchsize Restrictions	
	C	●	●	○	○	●	○	○	●	E	D	D	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Tabelle 1: Klassifikation des SLULSP

Ein häufig verwendetes Optimierungsmodell ist das Multi-Level Capacitated Lot Sizing (MLCLSP) Problem und dessen Varianten.⁷⁶ Es ist die mehrstufige Version des Capacitated Lot Sizing Problem (CLSP). Im Gegensatz zum CLSP wird die Struktur des Produktionsprozesses berücksichtigt. Es können mehrere Produkte produziert werden, die Nachfragemenge ist dynamisch und die Erzeugnisstruktur mehrstufig. Mehrere Maschinen produzieren dabei mit einer endlichen Produktionsgeschwindigkeit, wobei mehrere Produkte pro Periode produziert werden können.⁷⁷ Allerdings kann ein Rüstzustand nicht aus der Vorperiode übernommen werden. Hieraus folgt, dass, falls ein Produkt in einer Periode produziert wird, auch Rüstzeiten und Rüstkosten anfallen.

Im Folgenden wird das MLCLSP als mathematisches Modell formuliert und anschließend auf Grundlage der Anforderungen an eine Problemlösung debattiert.

Parameter

Symbol	Definition
a_{ki}	Direktbedarfskoeffizient bezüglich Produkt k und i .
b_{jt}	verfügbare Kapazität der Ressource j in Periode t .
d_{kt}	Primärbedarf für Produkt k in Periode t .
h_k	voller Lagerkostensatz des Produkts k (auch periodenabhängig definierbar).
J	Anzahl der Ressourcen ($j = 1, 2, \dots, J$).

⁷⁶ Vgl. [Tem06]

⁷⁷ In der Literatur auch als „Big-Bucket Modell“ bekannt.

K	Anzahl der Produkte bzw. Arbeitsgänge ($k = 1, 2, \dots, K$).
K_j	Indexmenge der Arbeitsgänge, die durch die Ressource j vollzogen werden.
M	Maximal mögliche Losgröße.
N_k	Indexmenge der Nachfolger des Produkts k (direkt übergeordnete Produkte bzw. nachfolgende Arbeitsgänge).
p_{kt}	variable Produktionskosten für Produkt k in Periode t .
q_{kt}	Losgröße für Arbeitsgang k in Periode t .
s_k	Rüstkostensatz des Produkts k (dieser kann auch periodenabhängig definiert werden).
T	Länge des Planungszeitraums in Perioden ($t = 1, 2, \dots, T$).
tb_k	Stückbearbeitungszeit für Arbeitsgang k .
tr_k	Rüstzeit für Arbeitsgang k .
y_{kt}	Lagerbestand für Produkt k am Ende der Periode t .
z_k	Mindestvorlaufzeit eines Auftrags für Produkt k .
γ_{kt}	binäre Rüstvariable für Arbeitsgang bzw. Produkt k in Periode t .

Tabelle 2: Parameter des MLCLSP

Zielfunktion

$$\text{Min } Z_{\text{MLCLSP}} = \sum_{m=1}^M \sum_{t=1}^T (s_k \cdot \gamma_{kt} + h_k \cdot y_{kt} + p_{kt} \cdot q_{kt}) \quad (1.1)$$

Nebenbedingungen

$$y_{k,t-1} + q_{k,t-z_k} - \sum_{i \in \mathcal{N}(k)} a_{ki} \cdot q_{it} - y_{kt} = d_{kt} \quad \begin{array}{l} k = 1, 2, \dots, K \\ t = 1, 2, \dots, T \end{array} \quad (1.2)$$

$$\sum_{k \in K_j} (tb_k \cdot q_{kt} + tr_k \cdot \gamma_{kt}) \leq b_{jt} \quad \begin{array}{l} j = 1, 2, \dots, J \\ t = 1, 2, \dots, T \end{array} \quad (1.3)$$

$$q_{kt} - M \cdot \gamma_{kt} \leq 0 \quad \begin{array}{l} k = 1, 2, \dots, K \\ t = 1, 2, \dots, T \end{array} \quad (1.4)$$

$$q_{kt} \geq 0 \quad \begin{array}{l} k = 1, 2, \dots, K \\ t = 1, 2, \dots, T \end{array} \quad (1.5)$$

$$y_{k0} = 0; y_{kT} = 0 \quad k = 1, 2, \dots, K \quad (1.6)$$

$$y_{kt} \geq 0 \quad \begin{array}{l} k = 1, 2, \dots, K \\ t = 1, 2, \dots, T \end{array} \quad (1.7)$$

$$\gamma_{kt} \in \{0,1\} \quad \begin{array}{l} k = 1, 2, \dots, K \\ t = 1, 2, \dots, T \end{array} \quad (1.8)$$

Durch die Zielfunktion (1.1) wird sichergestellt, dass die Summe aller Produktions- und Rüstkosten für die Maschinen sowie aller Lagerhaltungskosten für die Produkte minimiert werden. Dies stellt als Ausgangspunkt für das geforderte Modell eine gute Basis dar. Allerdings werden in der Zielfunktion keine Kosten für das Zurückstellen und das Canceln eines Bedarfs sowie Opportunitätskosten für ungenutzte Maschinenkapazitäten erfasst.

Mit der Lagerbilanzgleichung (1.2) werden die Bestände am Ende einer Periode mit den Produktionsmengen und den Bedarfen dieser Periode sowie den Beständen der vorangegangenen Periode verknüpft. Aus den Anforderungen an eine Lösung ergibt sich, dass Kundenbedarfe Schwankungen unterliegen. Dies wird in dem Modell jedoch nicht abgebildet und eine Erweiterung ist erforderlich.

Durch die Ungleichung (1.3) ist die maximale Kapazitätsnutzung der Maschinen in den einzelnen Perioden durch das jeweils entsprechende Kapazitätsangebot nach oben beschränkt. Aus den Anforderungen an eine Lösung ergibt sich, dass die Kapazitäten damit korrekt abgebildet werden.

Unter Berücksichtigung der Ungleichung (1.4) wird die maximale Losgröße vorgegeben. In den Anforderungen an eine Lösung wird eine maximale Losgröße nicht gefordert und sollte deshalb auch nicht modelliert werden.

Die Nebenbedingung (1.5) stellt sicher, dass die Losgröße für einen Arbeitsgang in einer Periode immer mindestens Null beträgt. Mit den Restriktionen (1.6) werden die Anfangsbestände eines Produktes in Periode Null und der Endbestand eines Produktes in Periode T auf Null gesetzt. Durch die Restriktion (1.7) wird sichergestellt, dass der Lagerbestand immer mindestens Null beträgt. Die Restriktion (1.8) definiert den zulässigen Wertebereich für die binäre Rüstvariable.

In Tabelle 3 wird das MLCLSP klassifiziert.⁷⁸

Model	Objective	Env.	Production	Ressources
MLCLSP	Objective (C/U/O)			
	├ Production Costs			
	├ Setup Costs			
	├ Setup Carry Over			
	├ Sequence Depend. SC			
	├ Stocking Costs			
	├ Echelon Stock			
	├ Backlog /Stockout			
	Terminated Horizon			
	├ Big-Bucket/SmallBucket	B		
	├ Static/Dynamic	D		
	Determ./Stochastic	D		
	Single-/Multiproduct		M	
	Multistage			●
	├ Lead-Times			●
	├ Handover batch-wise			○
	Production Bounds			○
	Multi Maschines			○
├ Parallel Maschines			○	
Capacitated Stock			○	
Production cap. C/CC/U			C	
Setup Times			●	
├ Sequence Depend. ST			○	
├ Fract. Setup Times			○	
Minimal On-/Off Times			○	
Cleaning Times			○	
Batchsize Restrictions			○	

Tabelle 3: Klassifikation des MLCLSP

⁷⁸ Vgl. [Alt06]

Dem MLCLSP liegt ein Big-Bucket Modell zugrunde, bei dem die Perioden so lang sind, dass auch mehrere Produkte in einer Periode produziert werden können. Allerdings kann dabei der Rüstzustand aus der Vorperiode nicht übernommen werden. Es werden weiterhin keine stochastischen Einflüsse berücksichtigt. Dies bewirkt unrealistische Umwelt- und Produktionsbedingungen. Grund hierfür ist, dass davon ausgegangen wird, dass Unsicherheiten in den Daten, wie Bedarfsschwankungen des Endkunden oder der Ausfall von Maschinen, zum Planungszeitpunkt nicht beherrschbar sind.

Die Begrenzung des Planungshorizonts mit der Vernachlässigung des Blicks über den Planungshorizont hinaus verursacht nicht optimale Reserven wie Lagerbestände und Maschinenkapazitäten.

Für das betrachtete Problem ist es nicht notwendig, das Modell zu lösen, sondern es reicht die Produktionsbedingungen zu formulieren und nach der Planerstellung die Einhaltung der Restriktionen zu überprüfen. Zusammenfassend kann gesagt werden, dass das MLCLSP die Anforderung an eine Lösung nicht erfüllt, allerdings Hinweise für eine Lösung gibt.

Drexl und Haase zeigen, dass die optimale Lösung des Discrete Lot Sizing and Scheduling Problem (DLSP) Modells niemals besser sein kann als die optimale Lösung des Continuous Setup Lot Sizing Problem (CSLP) Modells, sowie diese wiederum niemals besser ist als die optimale Lösung des Proportional Lot Sizing and Scheduling (PLSP) Modells.⁷⁹ Damit sind sowohl das DLSP als auch das CSLP ungeeignet, um das Planungsproblem abzubilden.

Das Proportional Lot Sizing and Scheduling with Parallel Machines (PLSP-PM) Optimierungsmodell ist eine Erweiterung des PLSP Modells. Dieses erweitert sowohl das DLSP als auch das CSLP, weil in beiden Modellen angenommen wird, dass höchstens ein Produkt pro Periode produziert werden kann.⁸⁰ Dem Modell liegt die Annahme zugrunde, dass mehrere Produkte für die Befriedigung einer dynamischen Nachfragemenge hergestellt werden müssen. Im Gegensatz zu anderen Varianten des PLSP kann ein Produkt im PLSP-PM Modell auf verschiedenen Maschinen produziert werden. Dadurch ist zu Beginn der Planung noch unklar, auf welcher Maschine ein Produkt tatsächlich produziert wird und das in der Einleitung abgegrenzte Planungsproblem wird diesbezüglich treffend abgebildet.

Es wird angenommen, dass mehrere Maschinen ein bestimmtes Produkt produzieren können, wobei dieses Produkt unter Umständen auf jeder Maschine eine andere Maschinenkapazität benötigt. Dem Modell liegt die Annahme zugrunde, dass die

⁷⁹ Vgl. [DH95]

⁸⁰ Vgl. [Fle90], [KKK87]

Produktionsgeschwindigkeit endlich ist, wobei je Periode höchstens ein Produktwechsel erfolgen kann.⁸¹

Im Folgenden wird das PLSP-PM als mathematisches Modell formuliert und auf Grundlage der Anforderungen an eine Problemlösung diskutiert.

Parameter

Symbol	Definition
a_{ji}	Gozinto Faktor. Der Wert ist Null falls Produkt i kein unmittelbarer Nachfolger von Produkt j ist. Ist Produkt i jedoch ein unmittelbarer Nachfolger von Produkt j so gibt a_{ji} die Menge von j an, die benötigt wird, um eine Einheit von Produkt j herzustellen.
C_{mt}	verfügbare Kapazität der Maschine m in Periode t.
d_{jt}	externer Bedarf nach Produkt j in Periode t.
h_j	nicht negative Lagerkosten für Produkt j pro Periode.
I_{j0}	Lagerbestand von Produkt j in Periode Null.
J	Anzahl der Produkte.
M	Anzahl der Maschinen.
m_j	Maschine auf der Produkt j produziert wird.
N_j	Menge aller Maschinen, die in der Lage sind, Produkt j zu produzieren.
P_m	Menge aller Produkte die Maschine m nutzen.
p_{jm}	benötigte Kapazität, um eine Einheit von Produkt j auf Maschine m zu produzieren; p_{jm} wird auf ∞ gesetzt falls Maschine m das Produkt j nicht produzieren kann.
q_{jmt}	Produktionsmenge von Produkt j auf Maschine m in Periode t.
S_{jm}	nicht negative Rüstkosten für Produkt j auf Maschine m.
S_j	Anzahl der unmittelbaren Nachfolger von Produkt j.
v_j	Vorlaufzeit von Produkt j.
x_{jmt}	Binärvariable, die anzeigt, ob für Produkt j eine Rüstung auf Maschine m in Periode t erfolgt ($x_{jmt} = 1$) oder nicht ($x_{jmt} = 0$).
y_{jm0}	Anfangsrüstzustand.
y_{jmt}	Binärvariable, die anzeigt, ob Maschine m für Produkt j am Ende der Periode t gerüstet ist ($y_{jmt} = 1$) oder nicht ($y_{jmt} = 0$).

Tabelle 4: Parameter des PLSP-PM

⁸¹ In der Literatur auch als „Small-Bucket Modell“ bekannt.

Zielfunktion

$$\text{Min } Z_{\text{PLSP-PM}} = \sum_{m=1}^M \sum_{j \in P_m} \sum_{t=1}^T (s_{jm} \cdot x_{jmt}) + \sum_{j=1}^J \sum_{t=1}^T (h_j \cdot I_{jt}) \quad (1.1)$$

Nebenbedingungen

$$I_{jt} = I_{j(t-1)} + \sum_{m \in N_j} q_{jmt} - d_{jt} - \sum_{i \in S_j} \sum_{m \in N_i} a_{ji} \cdot q_{imr} \quad \begin{array}{l} j = 1, \dots, J \\ t = 1, \dots, T \end{array} \quad (1.2)$$

$$I_{jt} \geq \sum_{i \in S_j} \sum_{m \in N_i} \sum_{r=t+1}^{\min\{t+v_j, T\}} a_{ji} \cdot q_{imr} \quad \begin{array}{l} j = 1, \dots, J \\ t = 0, \dots, T-1 \end{array} \quad (1.3)$$

$$\sum_{j \in P_m} y_{jmt} \leq 1 \quad \begin{array}{l} m = 1, \dots, M \\ t = 1, \dots, T \end{array} \quad (1.4)$$

$$x_{jmt} \geq y_{jmt} - y_{jm(t-1)} \quad \begin{array}{l} j = 1, \dots, J \\ m \in N_j \\ t = 1, \dots, T \end{array} \quad (1.5)$$

$$p_{jm} \cdot q_{jmt} \leq C_{mt} \cdot (y_{jm(t-1)} + y_{jmt}) \quad \begin{array}{l} j = 1, \dots, J \\ m \in N_j \\ t = 1, \dots, T \end{array} \quad (1.6)$$

$$\sum_{j \in P_m} p_{jm} \cdot q_{jmt} \leq C_{mt} \quad \begin{array}{l} m = 1, \dots, M \\ t = 1, \dots, T \end{array} \quad (1.7)$$

$$y_{jmt} \in \{0, 1\} \quad \begin{array}{l} j = 1, \dots, J \\ m \in N_j \\ t = 1, \dots, T \end{array} \quad (1.8)$$

$$I_{jt} \geq 0 \quad \begin{array}{l} j = 1, \dots, J \\ t = 1, \dots, T \end{array} \quad (1.9)$$

$$q_{jmt}, x_{jmt} \geq 0 \quad \begin{array}{l} j = 1, \dots, J \\ m \in N_j \\ t = 1, \dots, T \end{array} \quad (1.10)$$

Die Zielfunktion (1.1) minimiert die Summe der Lagerhaltungs- und Rüstkosten. Allerdings werden keine Produktionskosten, wie es in anderen Ansätzen⁸² üblich ist und in den Anforderungen an eine Lösung gefordert ist, modelliert.

⁸² Z.B. dem MLCLSP.

Mit der Lagerbilanzgleichung (1.2) werden die Bestände am Ende einer Periode mit den Produktionsmengen und den Bedarfen dieser Periode sowie den Beständen der vorangegangenen Periode verknüpft. Um die interne Nachfrage zu befriedigen, müssen positive Vorlaufzeiten beachtet werden, was die Ungleichung (1.3) garantiert.

Mit der Ungleichung (1.4) wird jeder Maschine ein eindeutiger Rüstzustand zugewiesen. Wird in einer Periode gerüstet, so wird dies durch Ungleichung (1.5) aufgezeigt. Ungleichung (1.6) garantiert, dass nur produziert werden kann, wenn der Rüstzustand gültig ist. Die Kapazitätsbedingung für jede Maschine wird in Ungleichung (1.7) erfasst. Bedingung (1.8) definiert eine Binärvariable. Nichtnegativität wird durch die Restriktionen (1.9) und (1.10) gewährleistet.

In Tabelle 5 wird das PLSP-PM klassifiziert.⁸³

Model	Objective								Env.			Production				Resources											
PLSP	Objective (C/U/O)	Production Costs	Setup Costs	Setup Carry Over	Sequence Depend. SC	Stocking Costs	Echelon Stock	Backlog / Stockout	Terminated Horizon	BigBucket / SmallBucket	Static / Dynamic	Determin./Stochastic	Single-/Multiproduct	Multistage	Lead-Times	Handover batch-wise	Production Bounds	Multi Machines	Parallel Machines	Capacitated Stock	Production cap. C/CC/U	Setup Times	Sequence Depend. ST	Fract. Setup Times	Minimal On-/Off Times	Cleaning Times	Batchsize Restrictions
	C	O	●	●	O	●	O	O	●	S	D	D	M	O	O	O	O	O	●	O	C	●	O	O	O	O	O

Tabelle 5: Klassifikation des PLSP-PM

Im PLSP-PM kann im Gegensatz zu einem Big-Bucket Modell der Rüstzustand aus der Vorperiode übernommen werden.⁸⁴ Des Weiteren bietet das PLSP-PM, wenn die Länge einer Periode angemessen klein gewählt wird, eine gute Annäherung an eine kontinuierliche Zeitachse.

Eine wesentliche Prämisse des PLSP-PM Modells ist, dass höchstens ein Rüstvorgang pro Maschine und Periode durchgeführt werden kann, woraus folgt, dass maximal zwei Produkte pro Maschine und Periode produziert werden können. Es werden weiterhin keine stochastischen Einflüsse wie Bedarfsschwankungen des Endkunden oder der Ausfall von Maschinen berücksichtigt, wie es die Anforderungen verlangen.

⁸³ Vgl. [Alt06]

⁸⁴ Vgl. [KD97], [DH95]

Damit zeigt sich, dass das PLSP-PM im Sinne der Anforderungen nicht als Ganzes für die Problemlösung verwendet werden kann, aber Ideen für eine Lösung liefert.

In der Literatur werden Versuche diskutiert, die Materialbedarfs- und Losgrößenplanung unter Unsicherheit abzubilden.⁸⁵ Gängige Optimierungsmodelle z.B. das MLCLSP setzen voraus, dass die Planungsdaten vollständig bekannt sind, was aber häufig nicht der Fall ist. Stochastischen Schwankungen unterliegen z.B. die Primärbedarfsmengen, die Durchlaufzeiten, die extern gelieferten Produkte und die Ausschussraten. Kundenbestellungen können dabei durch Wahrscheinlichkeitsverteilungen anstatt fester Kundenbestellungen abgebildet werden.

Sen formalisiert ein zweistufiges stochastisches Modell.⁸⁶ Die erste Stufe des zweistufigen Problems ist die Entscheidungsstufe, die zweite Stufe wird auch stochastische Stufe genannt. Die Berechnung der zweiten Stufe dominiert dabei den Lösungsvorgang.⁸⁷ Allerdings zielt die Modellierung des Planungsproblems noch nicht auf dessen Lösung ab, sondern das Problem soll abgebildet werden. Die anschließende Lösung erfolgt über Planungsverfahren, die selbst entscheiden müssen an welcher Stelle ggf. Vereinfachungen getroffen werden.

Bisher ist es nicht möglich, für ein Problem automatisiert ein Modell zu erstellen. Es wird für jedes Problem ein individuelles Modell entwickelt und deshalb entstand im Laufe der Jahre eine fast unüberschaubare Vielzahl von Modellen. Altemeier beschreibt in seiner Arbeit einen repräsentativen Ausschnitt an Modellen.⁸⁸ Allerdings erfüllt keines der Modelle die Anforderung an eine Lösung. Ein Grund hierfür ist, dass die Modelle darauf abzielen, sie später auch zu lösen und deshalb bereits bei der Erstellung des Modells Vereinfachungen getroffen werden. Hierdurch wird ein Modell zwar besser lösbar, jedoch entfernt es sich auch zunehmend von dem Problem in der Praxis. In dieser Arbeit soll jedoch ein Modell verwendet werden, um Bedingungen des Produktionsablaufs zu formalisieren und die entstandenen Ist-Kosten des Produktionsablaufs zu berechnen. Die Modellformulierung zielt dabei explizit nicht auf dessen Lösung hin, sondern darauf, das betrachtete Planungsproblem abzubilden.

Zusammenfassend ergibt sich damit folgendes Bild. Es existiert bereits eine Vielzahl von Ansätzen wie das MLCLSP und das PLSP die zusammengenommen in der Lage sind, das betrachtete Planungsproblem abzubilden. Allerdings existiert bisher kein Modell, das die Anforderungen an eine Lösung erfüllt und deshalb ist eine Neuentwicklung erforderlich, die die bestehenden Ansätze vereint und weiterentwickelt.

⁸⁵ Vgl. [Tem03]

⁸⁶ Vgl. [Sen00]

⁸⁷ Vgl. [SSV93]

⁸⁸ Vgl. [Alt06]

3.1.2 Ansätze für die Simulation des Produktionsablaufs

Produktion ist ein sich wiederholendes fortlaufendes Ereignis und es existieren für die Planung der Produktion spezifische Konzepte. Stadtler wendet ein Dekompositionsverfahren an, bei dem der Planungshorizont in mehrere kleine Planungsräume unterteilt wird und so für jedes kleine Planungsfenster eine exakte Lösungsfindung möglich wird.⁸⁹ Die Periode $t = 1, 2, \dots, 6$ wird zuerst geplant und darauf folgend wird die Planung für $6+1, \dots, 2T$ durchgeführt.⁹⁰ Hierbei wird zuerst von Periode 1 bis T geplant, anschließend eine bestimmte Anzahl Perioden fixiert und ein neuer Plan ab dem Zeitpunkt $\Delta T + 1$ erstellt „Rollierender Horizont“.⁹¹ Diesen Sachverhalt kann die Simulation des Produktionsablaufs nutzen, indem darauf fokussiert wird, von dem Planungsverfahren immer einen Produktionsplan in diskreten Zeitschritten just in time zu erhalten und die bereits entstandenen Ist-Kosten zu berechnen. Dabei kann das Planungsverfahren natürlich auch bereits einen Produktionsplan für eine Woche im Voraus erstellen.

Legner unterscheidet zwei Richtungen von Benchmarking:⁹²

- Quantitatives Benchmarking.
- Qualitatives Benchmarking.

Quantitatives Benchmarking misst Kennzahlen anhand objektiver Kriterien und ermöglicht einen Leistungsvergleich. Beim qualitativen Benchmarking werden Gestaltungsempfehlungen erarbeitet. Damit eignet sich für diese Arbeit quantitatives Benchmarking, indem Kennzahlen wie die von einem Verfahren verursachten Kosten miteinander verglichen werden.

Ein Benchmark ist eine vergleichende Analyse mit einem festgelegten Referenzwert und bedeutet, die wesentlichen Prozesse zu verstehen und mit den Prozessen anderer zu vergleichen.⁹³ Beim Benchmarking erfolgt damit ein direkter Vergleich der eigenen Position, in Bezug zu der Position der Konkurrenten. Dabei wird aus diesem Vergleich gelernt, um die eigenen Prozesse zu verbessern.⁹⁴ Für die Produktionsplanung kann

⁸⁹ Vgl. [Sta03]

⁹⁰ Diese Vorgehensweise bedeutet einen Planungsrhythmus alle T Perioden. Ein initialer Rüstzustand wird am Beginn der Periode gesetzt und die Zeiträume unabhängig geplant. Planungsperioden dürfen allerdings auch überlappen.

⁹¹ Vgl. [Sta03], [KD97]

⁹² Vgl. [Leg99]

⁹³ Vgl. [BGR94]

⁹⁴ Vgl. [ZK02]

ein Referenzwert festgelegt werden und am Ende der Simulation können die Planungsverfahren miteinander verglichen werden.

Eine Simulation des Produktionsablaufs ersetzt einen Wettbewerb, der aufgrund verschiedener Gründe z.B. der fehlenden Standardisierung von Planungsproblemen nicht existiert. Ein häufiger Fehler besteht darin, die Betrachtungsebene im Vorfeld nicht strikt abzugrenzen.⁹⁵ Des Weiteren sind für einen fairen objektiven Benchmark mit Chancengleichheit für alle Teilnehmer allgemeingültige Wettkampfgeregeln wichtig. Diese legen fest, was für die Teilnehmer in der Handlung vorgeschrieben, zulässig oder auch verboten ist.⁹⁶ Deshalb sollte das betrachtete Planungsproblem formal definiert werden. Hierfür wurde der Stand der Technik bereits in Kapitel 3.1.1 nach Möglichkeiten zur Modellierung des Planungsproblems analysiert. Eine dort gefundene Lösung kann als Grundlage der Betrachtungsebene verwendet werden, um einen fairen Wettbewerb der Verfahren zu ermöglichen.

Benchmarking im weiteren Sinne wird in verschiedenen Bereichen eingesetzt. Die Ursprünge von Hardware Benchmarks liegen in den 70er Jahren und wurden bis heute verbessert. Verlage wie die Computerzeitschrift C'T evaluieren z.B. Grafikkarten.⁹⁷ Dazu werden auf einem standardisierten Computer Programme gestartet und in mehreren Läufen mit verschiedenen Grafikkarten die Leistung gemessen. Aufgrund spezifizierter Kriterien wird ein Ranking erstellt. Die beste Grafikkarte wird dabei oft mit der vollen Punktzahl bewertet und alle anderen Grafikkarten entsprechend dieser Skala eingeordnet. Damit zeigt sich übertragen auf die Simulation des Produktionsablaufs, dass für den Vergleich von Planungsverfahren nicht zwingend der optimale Produktionsplan als Referenzwert gewählt werden muss, sondern es durchaus eine gute Möglichkeit ist, den besten bisher gefunden Produktionsplan als Referenzwert zu verwenden.

Für den Vergleich von Computern und Algorithmen wurde im Verlauf der letzten Jahre Schach das primäre Kampffeld. Im Schach finden dabei seit den fünfziger Jahren Turniere zwischen Computern und Menschen statt, wobei oft mehrere Matches gespielt werden, um den Sieger zu ermitteln.⁹⁸ Bei einem solchen Turnier findet ein Vergleich zwischen unterschiedlichen Spielern in einer Eins versus Eins Situation statt. Dabei vertrauen Großmeister oftmals auf ihre Intuition und verlieren so im Vergleich zu Computern, die einen Zug aufgrund programmierter Regeln ermitteln, immer mehr die Oberhand. Unterschiedliche Planungsstrategien werden damit in Form eines

⁹⁵ Vgl. [Wik07-o]

⁹⁶ Vgl. [TT99]

⁹⁷ Vgl. [Müs07]

⁹⁸ Vgl. [New97]

Wettkampfes zwischen zwei Teilnehmern unter Turnierbedingungen verglichen und die potentiellen Züge des Gegenspielers sind wie in der Produktionsplanung nur unter Unsicherheit planbar. Dieser Ansatz scheint geeignet zu sein, um den Zusammenhang zwischen einem Produktionsplaner und dem Ablauf in der Produktion abzubilden.

Gängige Algorithmen werden oftmals anhand künstlich erzeugter Testdaten verglichen. Dabei wird mit einem optimalen Verfahren das Gesamtoptimum für einen kleinen Testdatensatz ermittelt. Anschließend löst ein Verfahren das Problem und wird anhand der Kriterien Güte der Lösung und Laufzeit bewertet.⁹⁹ Es erfolgt ein Schluss von einem kleinen Problem auf große Probleme. Dem liegt die Annahme zu Grunde, dass Ergebnisse, die für kleine Probleminstanzen gelten, auch auf große Probleminstanzen übertragbar sind, für die die Ermittlung der optimalen Lösung nicht möglich ist. Übertragen auf die Produktionsplanung lassen sich mit entsprechender Vorsicht Aussagen zu Planungsverfahren, die für kleine Probleminstanzen gelten, auch auf große Planungsprobleme übertragen.

Eine weitere Möglichkeit für Benchmarking von Verfahren ist der Test in der Praxis. Hier sind die Bedingungen der Produktion realistischer als in jedem Modell, aber ein schlechtes Verfahren würde potentiell zur Insolvenz des Unternehmens führen. Damit erfüllt dieser Ansatz die Anforderungen an eine Lösung nicht.

Durch den Einsatz von PPS-Systemen lassen sich umfangreiche Datenmengen verwalten. Oftmals sind an die PPS-Systeme auch Business Intelligence Systeme angebunden, um graphische Auswertungen und Reportings zu erhalten.¹⁰⁰ Dies scheint auch für das in dieser Arbeit betrachtete Problem geeignet.

Das Planungsverfahren SAP APO ermöglicht es, einen Produktionsplan zu erstellen, wobei es aber unmöglich ist den Produktionsablauf zu simulieren und Planungsverfahren miteinander zu vergleichen. Damit werden die Anforderungen nicht erfüllt. Allerdings liegt SAP APO ein Datenmodell zugrunde, das ggf. genutzt werden kann. In der Stammdatenverwaltung von SAP APO werden Daten wie Lokation, Produkt, Ressource, Produktionsprozessmodell erstellt und gepflegt. Allerdings sollte das in dieser Arbeit verwendete Datenmodell keine unnötigen Datensätze abbilden und auf die mehrstufige auftragsbezogene Serienfertigung fokussieren. Dabei sollten zusätzlich zu den üblichen Stammdaten Schwankungen in der Produktionsmenge und im Kundenbedarf abgebildet werden, was gängige Datenmodelle bisher nicht ermöglichen. Damit werden die Anforderungen an eine Lösung nicht erfüllt.

Zusammenfassend kann gesagt werden, dass bereits eine Vielzahl von Ansätzen existiert, die in der Summe zur Lösung des Problems beitragen. Für sich allein

⁹⁹ Vgl. [PW05]

¹⁰⁰ Vgl. [BB01]

genommen existiert aber kein Lösungsansatz, der die Anforderungen an eine Lösung erfüllt.

3.2 Ansätze zur Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie

3.2.1 Ansätze für Planungsstrategien

Planungsstrategien lenken zielorientiert die langfristige Entwicklung eines Unternehmens und stellen die Grundlage eines Planungsverfahrens dar.¹⁰¹ Lödging entwickelte neun Leitstrategien, die sich an den von Wiendahl und Nyhuis formulierten produktionslogistischen Grundgesetzen orientieren.¹⁰²

1. Strategie: Regle den Bestand auf einem definierten Niveau.

Die Regelung des Bestandes auf einem definierten Niveau ermöglicht Unternehmen, die Zielerreichung bezüglich der logistischen Zielgrößen Bestand, Leistung und Durchlaufzeit aufeinander abzustimmen. Zum anderen erhöht eine Bestandsregelung die Vorhersagbarkeit der Durchlaufzeiten, ist aber mit dem Nachteil verbunden, dass Planungsvorgaben nicht exakt umgesetzt werden.

Aus den Anforderungen an eine Lösung ergibt sich, dass die Bestimmung des Bestandes nicht im Fokus stehen sollte, sondern vielmehr die Fähigkeit lieferfähig zu sein. Die geforderte Lieferfähigkeit kann natürlich durch Bestände erzielt werden, aber ebenso und potentiell effizienter durch freie Maschinenkapazitäten bzw. eine Kombination aus Beständen und freien Kapazitäten.

2. Strategie: Reduziere Bestandsschwankungen auf ein Minimum.

Die Reduzierung von Bestandsschwankungen durch weitestgehend feste Lagerbestände führt dazu, dass die Durchlaufzeiten weniger schwanken, wodurch sich die Terminplanung vereinfacht und die Termintreue verbessert. Je nach Auslastung der Maschinen ist es sinnvoll, Bestände vorzuhalten oder auf Bestände zu verzichten.

Auch wenn Bestandsschwankungen nachteilig auf die Terminplanung wirken, hebt dieser Nachteil den Vorteil niedrigerer Lagerkosten nicht auf. Deshalb sollten mit Blick auf die Anforderungen an eine Lösung die Bestände individuell ermittelt werden und nicht mit starren Werten für die Lagerbestände gerechnet werden.

3. Strategie: Verursache wenige blockierte Bestände.

Blockierte Bestände verursachen Reihenfolgevertauschungen und wirken nachteilig auf die Termintreue. Bestände sollten dabei immer dort verwendet

¹⁰¹ Vgl. [Tem04]

¹⁰² Vgl. [Löd05]

werden, wo sie am besten eingesetzt werden. Ex ante Bestände zu blockieren, zerstört deshalb Potential für die Optimierung des Produktionsablaufs. Diese Strategie kann damit einen Beitrag dazu leisten, das Optimierungspotential zu erhöhen.

4. Strategie: Trage zu einem guten zeitlichen Abgleich der Belastung mit der Kapazität bei.

Der Abgleich der Belastung mit der Kapazität beschleunigt bzw. verzögert den Durchlauf von Aufträgen mit dem Ziel, Materialflussabrisse zu vermeiden bzw. einen Bestandsaufbau zu verhindern.

Übertragen auf das Problem in dieser Arbeit können, indem die Kapazität an die Belastung angepasst wird, Lieferverzögerungen vermieden und Opportunitätskosten verringert werden. Damit erfüllt die Strategie die Anforderungen an eine Lösung.

5. Strategie: Verursache möglichst wenige Reihenfolgevertauschungen.

Arbeitssysteme sollten die Aufträge in der Reihenfolge ihrer Dringlichkeit abarbeiten. Dies reduziert die Streuung der Abgangstermineinhaltung und fördert eine hohe Termintreue. Aufträgen werden auf diese Weise Prioritäten zugewiesen, die sich aus der Dringlichkeit ergeben, um kritische Aufträge zu bevorzugen.

Wo es sinnvoll ist sollte damit ex ante eine Reihenfolge festgelegt werden und die Strategie erscheint für eine Lösung des Problems sinnvoll.

6. Strategie: Gleiche Planabweichungen im Rahmen einer Rückstandsregelung aus.

Droht ein positiver Rückstand, sollte die Kapazität kurzfristig erhöht werden. Ebenfalls sollte es möglich sein, die Kapazität zu verringern und Opportunitätskosten zu vermeiden. Die Kapazitäten anzupassen erscheint für die Problemlösung geeignet.

7. Strategie: Berücksichtige das Engpassprinzip.

An Engpassarbeitssystemen sollten keine Materialflussabrisse auftreten und damit sollten Kapazitätserhöhungen insbesondere hier durchgeführt werden.

Die Fokussierung auf Engpässe ermöglicht es, kritische Stellen im Produktionsablauf zu identifizieren und die Versorgung sicher zu stellen, indem Bestände eingeplant werden. Damit erfüllt die Strategie die Anforderungen an eine Lösung.

8. Strategie: Einfachheit.

Komplizierte Verfahren führen zu einem erhöhten Fehlerpotential und sind des Weiteren erläuterungsbedürftig, was wiederum die Akzeptanz verschlechtert.

Allerdings ist das kein Kriterium einen schlechteren Produktionsplan zu erstellen und sollte deshalb in dieser Arbeit nicht fokussiert werden.

9. Strategie: Kommuniziere Nachfrageinformationen möglichst schnell und unverzerrt.

Werden Nachfrageinformationen verspätet weitergeleitet, führt die verzögerte Kommunikation ggf. zu einer Überdimensionierung von Kapazitäten und Beständen.

Für das in dieser Arbeit betrachtete Problem ist dies aber nicht relevant und deshalb für eine Lösung des Problems auch ungeeignet, weil die Daten an das Planungsverfahren weitergegeben werden. Natürlich sollte das Planungsverfahren dabei aber immer mit den aktuellsten Planungsdaten arbeiten.

Bereits in den siebziger Jahren entwickelte Goldratt sein auf neun Strategien basierendes Produktionsplanungs- und Simulationsinstrument Optimized Production Technology¹⁰³ (OPT) und führte dieses 1980 auf dem amerikanischen Markt ein. Goldratt und Cox stellten folgende Strategien auf, die im Gegensatz zu den von Wiendahl und Nyhuis formulierten produktionslogistischen Grundgesetzen keine direkten Handlungsanleitungen darstellen, sondern vielmehr allgemeine Sachverhalte fokussieren, die individuell von einem Planungsverfahren berücksichtigt werden sollen.¹⁰⁴

1. Strategie: Harmonisiere den Fluss, nicht die Kapazitäten.

Wird der Fluss der Aufträge durch das Produktionssystem harmonisiert, können die Durchlaufzeit verringert und die Bestände verkleinert werden. Allerdings entstehen mit einer Harmonisierung des Flusses freie Kapazitäten, da auf eine maximale Auslastung der Kapazitäten verzichtet wird.

Kapazitäten sollten aber dennoch angepasst werden, um hohe Opportunitätskosten für ungenutzte Maschinen zu vermeiden. Eine Harmonisierung des

¹⁰³ Die Fertigungsziele Erhöhung des Fertigungs-Outputs, Senkung der Zwischen- und Endlagerbestände sowie die Senkung der Kosten für durchsatz erhöhende und bestandsverringende Maßnahmen sollen in OPT durch Beachtung von neun Strategien erreicht werden.

¹⁰⁴ Vgl. [GC87]

Flusses durch das Produktionssystem ist damit geeignet einen Beitrag zur Lösung des Problems zu leisten, wobei aber die Planung der Kapazitäten nicht vernachlässigt werden sollte.

2. Strategie: Der Nutzungsgrad einer Nicht-Engpassressource ist nicht durch ihre eigene Kapazität, sondern durch die Engpasskapazität gegeben. Eine Nicht-Engpassressource verursacht systemimmanent Opportunitätskosten weil sie nicht ausgelastet ist.

Fokus dieser Strategie liegt auf der laufenden Versorgung von Engpässen mit Aufträgen z.B. durch den Aufbau von Beständen und zielt auf eine Maximierung des Durchsatzes. Damit ist die Strategie im Sinne der Anforderungen geeignet, um Engpässe zu puffern, jedoch nicht um die Bestände zu verringern.

3. Strategie: Nutzung und Bereitstellung einer Ressource sind nicht synonym. Die Bereitstellung von Kapazitäten sollte geplant werden, um Engpässe zu vermeiden und keine unnötigen Opportunitätskosten zu verursachen. Kapazität die nicht mehr verringert werden kann, sollte bestmöglich genutzt werden, um in der Zukunft ggf. Kapazität auf dieser Ressource einzusparen.

Die Nutzung einer Ressource sollte deshalb im Sinne der Anforderungen an eine Lösung des Problems ggf. an einer anderen Stelle erfolgen, als die Bereitstellung einer Ressource.

4. Strategie: Eine an einem Engpass verlorene Stunde ist eine verlorene Stunde für das gesamte System. Engpässe sollten ständig mit Aufträgen versorgt werden, um einen reibungslosen Fluss zu gewährleisten.

Allerdings ist es nicht sinnvoll eine maximale Auslastung des Gesamtsystems durch eine maximale Auslastung der Engpass Ressourcen zu erreichen, nur um auf der anderen Seite hohe Lagerbestände aufzubauen und damit zusätzliche Lagerkosten zu verursachen. Deshalb ist es für das in dieser Arbeit betrachtete Problem nicht sinnvoll, das Produktionssystem immer auf Volllast zu fahren und auf diese Weise voll auszulasten, aber im Hinblick auf die Anforderungen an eine Lösung sinnvoll, Engpässe ggf. durch Bestände individuell zu puffern.

5. Strategie: Eine an einem Nicht-Engpass gesparte Stunde ist eine Illusion. Wird Kapazität an einem Nicht-Engpass gespart, erhöht dies die freie Kapazität, wirkt sich aber nicht auf den potentiellen Durchsatz aus.

Für das in dieser Arbeit betrachtete Problem sollte die Kapazität deshalb an einem Nicht-Engpass besser verkleinert werden oder wenn dies nicht möglich ist, eine Möglichkeit gesucht werden die Kapazitäten unter Beachtung der entstehenden Lagerkosten ggf. auszulasten, indem Aufträge aus der Zukunft vorgezogen werden und in der Zukunft verkleinert werden.

6. Strategie: Engpässe bestimmen sowohl den Durchsatz, als auch die Bestände.

Die Fokussierung auf Engpässe ermöglicht es, den Durchsatz zu erhöhen, liefert allerdings keine Lösung dafür, die Produktionskosten als Ganzes zu senken, sondern zielt insbesondere auf den maximal möglichen Durchsatz ab.

Für das betrachtete Problem erscheint die Strategie geeignet, die Lieferfähigkeit zu gewährleisten, indem Engpässe durch freie Kapazitäten oder Bestände gepuffert werden.

7. Strategie: Transportlose und Prozesslose sind in der Regel verschieden.

Der Transport von Produkten wird, wie in der Einleitung abgegrenzt, nicht fokussiert und die Strategie ist damit unter den gestellten Anforderungen ungeeignet.

8. Strategie: Das Prozesslos sollte variabel sein.

Wird das Prozesslos variabel anhand der Erfordernisse gebildet, können z.B. die Lagerkosten und die Rüstzeiten verringert werden. Die Strategie ist für die Lösung des Problems geeignet.

9. Strategie: Die Pläne sollten unter gleichzeitiger Beachtung aller Einschränkungen erstellt werden. Durchlaufzeiten sind Ergebnis des Plans bzw. der Steuerung und können nicht vorausbestimmt werden.

Nur wenn alle Einschränkungen bei der PPS betrachtet werden, kann vermieden werden, einen zwar in dem betrachteten Rahmen optimalen, aber im Ganzen ungültigen bzw. suboptimalen Produktionsplan zu erstellen. Die Strategie ist deshalb unter den gestellten Anforderungen geeignet, einen gültigen Produktionsplan zu erstellen.

Vorteil der OPT Strategien ist erstens die Konzentration auf Engpässe und die damit verbundene abgestufte Anforderung an die Qualität der Daten. Zweitens erreicht die Terminierung im nichtkritischen Teil des Produktnetzes verringerte Liegezeiten. Nachteil dieses Verfahrens ist, dass Probleme auftreten, falls sich Engpässe im Laufe

der Zeit ändern. Darüber hinaus ist der Aufwand für die Planung und Erstellung des Produktnetzwerkes bei einer Produktionsprogrammänderung hoch. Des Weiteren sind die OPT Strategien nicht für den Einsatz in dezentralen Strukturen geeignet.¹⁰⁵

Damit erfüllen die OPT Strategien nur in Teilen die Anforderungen an eine Lösung und sind in der bestehenden Form nur als Ausgangspunkt für eine Weiterentwicklung geeignet.

Andere Planungsverfahren beruhen strategisch auf einem Vergleich zwischen Lager- und Rüstkosten, um z.B. Lose zu bilden und die Gesamtkosten zu senken. Dabei werden je nach gewähltem Verfahren die Kosten entsprechend eines spezifischen Abgleichs gegenübergestellt und der kostengünstigste Schnittpunkt für die Losgröße gewählt.¹⁰⁶ Diese Strategien erfüllen die Anforderungen an eine Problemlösung jedoch nicht, weil sie einseitig auf den Abgleich von Lager- und Rüstkosten fokussieren und die Belegungssituation und die Auftragslage vernachlässigen.

Zusammenfassend ergibt sich damit, dass einige der Strategien unter Beachtung der Anforderungen für die Problemlösung geeignet sind und direkt verwendet werden können bzw. als Ausgangspunkt für eine Weiterentwicklung dienen. Für eine Problemlösung ist allerdings die Entwicklung neuer Planungsstrategien erforderlich, die unter Beachtung der Anforderungen bestehende Ansätze vereinen und die Lücken im Stand der Technik schließen.

¹⁰⁵ Vgl. [Gol88]

¹⁰⁶ Vgl. [Tem06], [ZR87], [Rob91]

3.2.2 Ansätze für die Produktionsplanung und -steuerung

Um einen Beitrag zur PPS zu leisten, wird zuerst der Stand der Technik analysiert. In Kapitel 3.2.2.1 wird die Materialbedarfsplanung im Hinblick auf die Anforderungen diskutiert und in Kapitel 3.2.2.2 der Stand der Technik zur Losgrößenplanung dahingehend untersucht, Rüstzeiten bei Bedarf zu verringern, aber nicht um im Vorfeld zwangsläufig eine vorgelagerte Losgrößenbildung durchzuführen¹⁰⁷. Die Termin- und Kapazitätsplanung wird in Kapitel 3.2.2.3 fokussiert und in Kapitel 3.2.2.4 Verfahren des Operations Research (OR) debattiert. Kapitel 3.2.2.5 analysiert den Stand der Technik zur Änderung eines Produktionsplans.

3.2.2.1 Materialbedarfsplanung

Die Materialbedarfsplanung bestimmt Art, Menge und Termin der Verbrauchsfaktoren, die für die Produktion eines Endproduktes nötig sind.¹⁰⁸ Die Bedarfe werden dazu aufgrund der Erzeugnisstruktur, die in Form einer Strukturstückliste dargestellt werden kann, festgelegt und alle Bestandteile eines Erzeugnisses einzeln aufgeführt. Durch explizites Aufführen und Kennzeichnen der Gliederungsstufe wird die Stellung in der Erzeugnishierarchie verdeutlicht. Folge davon ist, dass Produkte die an verschiedenen Stellen des Erzeugnisses eingehen auch mehrfach mit allen eingehenden Teilen aufgeführt werden. Dies wiegt den Nachteil einer unübersichtlichen Liste auf und scheint für eine Lösung geeignet zu sein.

Weiterhin gilt es für die Materialbedarfsplanung eine geeignete Datenstruktur zu nutzen, wobei dafür das für den Vergleich von Planungsverfahren zu entwickelnde Datenmodell geeignet scheint.

Die Nettobedarfsrechnung auf Basis von Fertigungsstufen bedingt, dass Vorräte die auf mehreren Stufen auftreten, mehrfach verrechnet werden. Dieses Problem ergibt sich bei einer Auflösung nach Dispositionsstufe nicht und der Rechenaufwand ist geringer.¹⁰⁹ Deshalb ist die Auflösung nach Dispositionsstufe für das in dieser Arbeit betrachtete Problem zu bevorzugen.

Beim Dispositionsstufenverfahren werden, ausgehend vom Primärbedarf, stufenweise die Sekundärbedarfe, bis hin zu den nicht zerlegbaren Einzelteilen, berechnet. Zunächst wird jedes Produkt einer Dispositionsstufe zugeordnet.¹¹⁰ Die einzelnen

¹⁰⁷ Vgl. die Ausführungen zu den Anforderungen an die PPS in Kapitel 2.5.2.2.

¹⁰⁸ Vgl. [Tem04]

¹⁰⁹ Vgl. [Sch04]

¹¹⁰ Die Dispositionsstufe entspricht dabei dem längsten Weg im Gozintographen von dem betrachteten Produkt zu den Endprodukten.

Stufen werden durchlaufen und für jedes untergeordnete Produkt die Sekundärbedarfsmenge aus den Nettobedarfsmengen der direkten Nachfolger abgeleitet.¹¹¹ Durch die Dispositionsstufenzuordnung und die sequenzielle Abarbeitung ist sichergestellt, dass die Nettobedarfsmenge der Nachfolgeprodukte bereits vollständig bekannt ist. Jedes Produkt der Erzeugnisstruktur kann genau einer Dispositionsstufe zugeordnet werden.¹¹² Dabei kann ein Zwischenprodukt auch in einer anderen Stufe selbst ein Vorprodukt sein. Diese Vorgehensweise ist gut geeignet für das in dieser Arbeit betrachtete Problem. Allerdings findet bisher kein simultaner Abgleich mit den aktuell verfügbaren Kapazitäten statt und eine Weiterentwicklung ist erforderlich.

Zusammenfassend lässt sich feststellen, dass bestehende Verfahren der Bedarfsermittlung die Konkurrenz der Erzeugnisse um begrenzte Ressourcen und die Mehrstufigkeit der Erzeugnisstruktur nur unzureichend berücksichtigen. Des Weiteren bedingt die Verwendung von Plan-Durchlaufzeiten, die auf Schätzwerten beruhen, dass oftmals von Anfang an durch eine Überschätzung der Pufferzeiten zu früh gefertigt wird und die Lagerbestände auf diese Weise unnötig erhöht werden. Die begrenzte Kapazität von Maschinen wird von Verfahren der Materialbedarfsplanung nicht berücksichtigt. Es werden somit aus systemimmanenten Gründen erhöhte Kosten und Lieferschwierigkeiten verursacht. Damit eignet sich keine der diskutierten Vorgehensweisen, um das Problem vollständig zu lösen. Allerdings bietet sich mit der Auflösung nach Dispositionsstufen ein Ansatz, der als Ausgangspunkt für die Entwicklung einer neuen Vorgehensweise für die Materialbedarfsermittlung geeignet zu sein scheint.

3.2.2.2 Losgrößenplanung

In der Losgrößenplanung wird bestimmt, welche in der Materialbedarfsplanung berechneten Nettobedarfe zusammengefasst werden. Zahlreiche Lösungsverfahren wurden mit unterschiedlichen Schwerpunkten entwickelt. Im Folgenden werden die für die Problemlösung vielversprechenden Ansätze anhand der Anforderungen an eine Lösung diskutiert.

Die einfachste Vorgehensweise ist es, einzelne Produkte nach Auftragseingang zu produzieren und die Maschine nach Fertigstellung des Auftrags umzurüsten, um das nächste Produkt zu produzieren. Allerdings werden durch den Stillstand der Maschine während der häufigen Umrüstungen hohe Kosten verursacht.¹¹³ Diese Vorgehensweise

¹¹¹ Vgl. [Tem04]

¹¹² Endprodukte gehören zur Dispositionsstufe null und Zwischen- und Vorprodukten beginnen ab Stufe eins.

¹¹³ Vgl. [Hop05]

ist damit geeignet, um möglichst geringe Lagerkosten zu verursachen, erfüllt aber insgesamt die Anforderungen an eine Lösung nicht.

Das Verfahren der exakten Losgröße ermittelt täglich die Unterdeckungsmenge über den Bedarf abzüglich des verfügbaren Lagerbestands. Ergibt sich eine Unterdeckung eines Materials, wird der Bedarf mit weiteren Bedarfsmengen, die am gleichen Tag aufgetreten sind, zu einem Bestellvorschlag zusammengefasst. Vorteile dieses Verfahrens sind die minimierten Lagerkosten und die Reduzierung der Bestände, weil zusätzlich zum Kundenbedarf keine weiteren Bestände aufgebaut werden. Ein Nachteil sind die erhöhten Losfixen Kosten.¹¹⁴ Das Verfahren ist deshalb ggf. geeignet, um einen Beitrag dazu zu leisten, die Lagerkosten zu senken.

Gewählt wird die feste Losgröße, wenn technische Gründe es erfordern, dass ein Material nur in bestimmten Mengen¹¹⁵ bezogen werden kann. Ebenso kann es möglich sein, dass ein Produkt nur in bestimmten Losgrößen gefertigt werden kann. Reicht die Menge einer festen Losgröße nicht aus, so werden in diesem Fall mehrere Lose zum gleichen Zeitpunkt eingeplant, bis die Menge der Unterdeckung ausgeglichen oder überschritten wird. Die feste Losgröße erzeugt also potentiell Bestände über den Bedarf hinaus.¹¹⁶ Daraus ergibt sich, dass die Vorgehensweise für eine Lösung des Problems ungeeignet ist.

Das Auffüllen bis zum Höchstbestand wird gewählt, wenn auf Grund von Lagerrestriktionen nur bestimmte Mengen im Bestand gehalten werden können. In diesem Fall füllt eine Bestellung das Lager komplett auf und verursacht dauerhaft hohe Lagerkosten. Der Ansatz erfüllt damit nicht die Anforderungen an die Lösung des Problems.

Statische Verfahren gehen davon aus, dass Bedarfe, die in der Zukunft liegen, unbekannt sind. Eine Vorplanung ist nicht möglich und infolgedessen müssen die Lagerbestände ständig kontrolliert werden.¹¹⁷ Vorteile sind die einfache Handhabung, Logik und die Beeinflussung der Losgröße. Nachteilig ist vor allem der hohe Pflegeaufwand, da für jedes Material Stammdaten gespeichert und gewartet werden müssen. Weiterhin müssen die Parameter anhand von Erfahrungswerten festgelegt werden.¹¹⁸ Damit sind sie ungeeignet, um die Bestände zu senken. Allerdings

¹¹⁴ Vgl. [Hop05], [HGH02]

¹¹⁵ Bestimmte Produkte können z.B. nur palettenweise bezogen werden.

¹¹⁶ Vgl. [Hop05], [HGH02]

¹¹⁷ Vgl. [Hop05]

¹¹⁸ Vgl. [BL06]

ermöglichen Mindestlosgrößen, die Rüstzeiten zu verringern. Dieser Sachverhalt kann für die Erzeugung von Pufferaufträgen¹¹⁹ genutzt werden.

Die periodischen Verfahren Tageslosgröße, Wochenlosgröße, Monatslosgröße¹²⁰, Losgröße nach PPS-Kalender¹²¹ und Schichtlosgröße¹²² basieren auf Erfahrungswerten und Prognosen. Es wird abgeschätzt, welche zukünftigen Bedarfe anfallen. Anschließend findet im Vorfeld eine Planung statt und die Losgrößen einer oder mehrerer Perioden werden zusammengefasst. Die Anzahl der Perioden kann dabei variabel gehalten werden.¹²³ Vorteile der periodischen Verfahren sind die einfache Anwendbarkeit, eine einfache zu Grunde liegende Logik sowie die Anpassbarkeit der Losgrößen. Erfahrungsgemäß werden die Verfahren in der Praxis jedoch nicht eingesetzt, weil das Wissen darüber fehlt und die Ermittlungsmethoden kompliziert sind.¹²⁴ Geeignet scheint die Vorgehensweise, eine bestimmte Anzahl von Perioden für die Zusammenfassung von Aufträgen zu Lösen zu betrachten. Allerdings sollte dies nicht zwangsläufig erfolgen, sondern nur wenn Kapazitätsengpässe es erfordern, zu Lasten höherer Lagerkosten, Rüstzeit zu sparen.

Optimierende Losgrößenverfahren basieren in den Grundzügen auf dem Andlerschen Losgrößenverfahren. Die für die Anwendung des Verfahrens erforderlichen Rahmenbedingungen sind in der Praxis jedoch selten zu finden und eine optimale Lösung kann durch dieses Verfahren oft nicht erreicht werden.¹²⁵ Dabei geht das Verfahren von einem konstanten Bedarf aus und ist damit unter den an eine Lösung gestellten Anforderungen ungeeignet.

¹¹⁹ Pufferaufträge bezeichnen Aufträge denen noch kein Kundenauftrag gegenübersteht, die aber dennoch gefertigt werden sollen.

¹²⁰ Bei der Tageslosgröße, der Wochenlosgröße oder der Monatslosgröße werden alle Bedarfe eines Tages, einer Woche oder eines Monats zu einem Los zusammengefasst.

¹²¹ Die Losgröße nach PPS-Planungskalender wird als Losgröße nach frei definierbaren Perioden beschrieben, wobei die Periode abhängig vom PPS-Planungskalender ist. Es können auch hier die Bedarfe nach einer oder mehreren frei wählbaren, flexiblen Perioden zusammengefasst werden.

¹²² Das Verfahren der Schichtlosgröße fasst während einer oder mehrerer Schichten, alle Bedarfe zu einer Losgröße zusammen. Bei stark schwankendem Bedarf kommt es zu höheren Kosten und einem höheren Pflegeaufwand.

¹²³ Vgl. [Hop05], [HGH02]

¹²⁴ Vgl. [Hop05], [HGH02], [BL06]

¹²⁵ Vgl. [And29], [Dan03], [Tem06]

Das Groff-Verfahren ist eine Heuristik die auf der klassischen Losgrößenformel nach Andler aufbaut und ausnutzt, dass beim Kostenminimum zusätzlich anfallende Lagerkosten gleich der Losfixkostensparnis sind.¹²⁶ Dabei sind die Grenzkosten von Lager- und Rüstkosten bei der optimalen Losgröße gleich hoch.¹²⁷ Bei Anwendung des Groff-Verfahrens werden die Bedarfsmengen der hintereinander liegenden Perioden schrittweise zu einem Los zusammengefasst.¹²⁸ Die Steigerung der Losgröße endet, sobald die durchschnittlichen Lagerkosten pro Periode die Verringerung der losfixen Kosten pro Periode erstmals übersteigen.¹²⁹ Die schrittweise Zusammenfassung von benachbarten Aufträgen zu einem Los stellt damit eine effektive Möglichkeit dar, um Rüstzeit zu sparen. Allerdings sollten die Lagerkosten den gesparten Opportunitätskosten gegenübergestellt werden, da, wenn freie Kapazität ohnehin verfügbar ist, auch keine Rüstkosten entstehen. Das Verfahren scheint damit einen Beitrag für die Problemlösung leisten zu können.

Wagner und Within entwickelten das dynamisch optimierende Wagner-Within-Verfahren. Dieses setzt allerdings bestimmte Bedingungen voraus, die oftmals nicht erfüllt sind.¹³⁰ Weiterhin ist das Verfahren, wie alle anderen analysierten Verfahren auch, durch seinen begrenzten Horizont eingeschränkt und führt somit nicht immer zu einem optimalen Ergebnis.¹³¹ Damit erfüllt das Verfahren nicht die Anforderung an eine Lösung.

Das Verfahren der gleitenden wirtschaftlichen Losgröße basiert darauf, dass die Funktion der durchschnittlichen Kosten, bezogen auf eine Mengeneinheit, an der Stelle der optimalen Losgröße ihr Minimum aufweist.¹³² Diese Eigenschaft wird auf eine dynamische Situation übertragen.¹³³ Ziel ist es, die Stückkosten des Loses zu minimieren. Dabei werden die Bedarfsmengen aufeinander folgender Perioden

¹²⁶ Diese Bedingung bedeutet, dass die marginale Verringerung der durchschnittlichen Rüstkosten pro Periode dem marginalen Anstieg der durchschnittlichen Lagerkosten pro Periode entspricht [Tem06], [Hop05].

¹²⁷ Vgl. [GT04]

¹²⁸ Nach jedem Schritt werden die durch die Erhöhung der Losgröße zusätzlich anfallenden Lagerkosten, mit den Kostenersparnissen durch Erhöhung der Losgröße verglichen.

¹²⁹ Vgl. [Hop05], [Gro79]

¹³⁰ Vgl. [Hop05], [HGH02], [And29], [WW58]

¹³¹ Vgl. [Tch07]

¹³² Vgl. [Tem06]

¹³³ Vgl. [Tem06], [GT04]

schrittweise zu einem Los zusammengefasst.¹³⁴ Das Verfahren wird wiederholt, bis die minimalen Stückkosten gefunden sind.¹³⁵ Bei Durchführung des Verfahrens der gleitenden wirtschaftlichen Losgröße wird keine optimale Losgröße gefunden, sondern lediglich eine angenähert optimale Lösung. Es ist das meistgenutzte Verfahren, obwohl Verfahren existieren, die bessere Ergebnisse liefern.¹³⁶ Wemmerlöv zeigte, dass das Verfahren schlechtere Ergebnisse liefert als das Groff-Verfahren.¹³⁷ Weiterhin wird durch die Fokussierung auf Stückkosten vernachlässigt, dass Rüstzeiten nicht zwingend auch Kosten verursachen. Das Verfahren erfüllt damit nicht die an eine Lösung gestellten Anforderungen.

Das Stückperiodenausgleichsverfahren ist eine Heuristik, die auf dem klassischen Losgrößenmodell basiert und ausnutzt, dass bei der optimalen Losgröße die Rüstkosten gleich den Lagerkosten sind und es zu einem Ausgleich der Kosten kommt.¹³⁸ Ausgehend von der Periode der Losgrößenplanung werden die Einzelbedarfe aufeinander folgender Perioden zu einem Los zusammengefasst. Dies erfolgt, bis die aus der Losgrößenerhöhung resultierenden ansteigenden Lagerkosten, die Rüstkosten übersteigen.¹³⁹ Die Rüstkosten fallen dabei unabhängig von der Losgröße bei jeder Bestellung an.¹⁴⁰ Wemmerlöv stellt allerdings fest, dass das Stückperiodenausgleichsverfahren schlechtere Ergebnisse als das Groff-Verfahren ermittelt.¹⁴¹ Damit erfüllt das Stückperiodenausgleichsverfahren nicht die Anforderungen an eine Lösung. Der Vergleich von Lager- und Rüstkosten stellt aber einen guten Ausgangspunkt für eine Weiterentwicklung dar.

Insgesamt zeigt sich, dass kein Verfahren geeignet ist, das Problem zu lösen. Die Verfahren fokussieren einseitig auf Rüst- und Lagerkosten und vernachlässigen, dass Rüstkosten eventuell gar nicht entstehen, wenn freie Kapazität verfügbar ist. Die Ansätze sind zusammengenommen jedoch gut geeignet, um Rüstzeiten zu sparen, wenn dies erforderlich ist, um Kapazität für die Produktion zu schaffen.

¹³⁴ Nach jedem Hinzufügen einer Bedarfsmenge werden die anfallenden Kosten pro Mengeneinheit berechnet und mit den Kosten der Vorperioden verglichen.

¹³⁵ Vgl. [Hop05]

¹³⁶ Vgl. [Hop05], [Tem06]

¹³⁷ Vgl. [Wem81], [Wem82], [WW84], [DV83], [LA86]

¹³⁸ Vgl. [Hop05], [Tem06]

¹³⁹ Vgl. [Hop05]

¹⁴⁰ Vgl. [Dem68]

¹⁴¹ Vgl. [Wem81], [Wem82], [WW84], [DV83], [LA86]

3.2.2.3 Termin- und Kapazitätsplanung

Wurden die zu produzierenden Mengen berechnet, wird die Termin- und Kapazitätsplanung durchgeführt.

Mit der Rückwärtsterminierung können die spätmöglichen Zwischen- und Starttermine der Fertigungsaufträge ermittelt werden, wobei die jeweiligen Endtermine vorgegeben sind. Um den spätesten Starttermin zu bestimmen, wird bei der Rückwärtsterminierung vom Endtermin des letzten Vorgangs ausgegangen. Auf Grundlage des Starttermins eines Nachfolgeknotens, wird anschließend mit der Durchführungszeit der Starttermin des Vorgängerknosens berechnet.¹⁴² Somit wird über dazwischen liegende Vorgänge der Starttermin des ersten Vorgangs bestimmt.

Wird ein Starttermin ermittelt, der in der Vergangenheit liegt, wird eine Kombination aus Vorwärts- und Rückwärtsterminierung verwendet. Hierbei wird eine Reduzierung der Auftragsdurchlaufzeit durchgeführt, z.B. durch Reduzierung der Transportzeiten. Anschließend wird eine Vorwärtsterminierung vom Gegenwartszeitpunkt ausgeführt.¹⁴³ Kann der geplante Endtermin nicht eingehalten werden, muss eine Lieferverzögerung hingenommen werden.¹⁴⁴

Vorteil der Rückwärtsterminierung ist, dass es ein einfaches und manuell durchführbares Verfahren ist, durch dessen Einsatz eine schnelle Terminaussage möglich ist. Die Produkte werden zum spätestmöglichen Zeitpunkt gefertigt und damit gegenüber der Vorwärtsterminierung geringere Lagerhaltungs- und Kapitalbindungskosten verursacht. Im Gegensatz zum einfachen Erstellen des Terminplans ist die Veränderung des Termingerüsts jedoch aufwändig. Bei einer Störung im Produktionsablauf kann es zu Lieferüberschreitungen kommen.¹⁴⁵ Die gestellten Anforderungen werden damit nicht erfüllt; das Verfahren ist für eine Lösung ungeeignet. Aber die Rückwärtsterminierung ist als Ausgangsbasis für die Konzeption eines neuen Planungsverfahrens gut geeignet, um die Bestände zu senken, wobei darauf geachtet werden muss, dass der Produktionsplan flexibel bleibt, um auf Störungen zu reagieren.

Hingegen berechnet die Vorwärtsterminierung für alle Aufträge den frühesten Anfangstermin. Die Zeitplanung beginnt auf der untersten Fertigungsstufe und verläuft in Fertigungsrichtung bis zum Enderzeugnis. Damit bestimmt die Vorwärtsterminierung

¹⁴² Vgl. [VS04]

¹⁴³ Liegt der errechnete Fertigungstermin nach der reduzierten Auftragsdurchlaufzeit vor dem geplanten Endtermin, so ist eine fristgerechte Lieferung möglich.

¹⁴⁴ Vgl. [DW97]

¹⁴⁵ Vgl. [DW97]

auch den frühesten Fertigstellungszeitpunkt eines Produktes.¹⁴⁶ Hiermit werden hohe Lagerbestände erzeugt und das Verfahren ist ungeeignet, um die Bestände dauerhaft zu senken.

Eine weitere Variante sind Prioritätsregeln, wobei einfache Varianten aus zwei Schritten bestehen, erstens dem Sortieren nach einer Regel und zweitens der Ausführung in der gewählten Reihenfolge. Aufträge in der Produktion können z.B. nach monoton wachsenden Bearbeitungszeiten sortiert werden.¹⁴⁷ Anschließend erfolgt die Einlastung der sortierten Aufträge anhand der festgelegten Reihenfolge auf die Maschinen.

Johnson entwickelte die auf Prioritätsregeln basierende Johnson-Heuristik.¹⁴⁸ Diese beschränkt sich allerdings auf zwei Maschinen und entspricht deshalb nicht den Anforderungen an eine Lösung.

Nawaz, Ensore und Ham entwickelten ein Einbeziehungsverfahren, das alle Aufträge nach monoton fallender Gesamtbearbeitungszeit sortiert.¹⁴⁹ Die Heuristik führt dabei zu guten Ergebnissen im Hinblick auf die Minimierung der Durchlaufzeit und der Minimierung der Leerzeiten.¹⁵⁰ Allerdings wird vernachlässigt, dass für Aufträge ein Liefertermin zugesagt wird und ggf. wird ein Auftrag zu spät oder zu früh erfüllt. Eine zu späte Erfüllung bedingt, dass das Unternehmen nicht lieferfähig ist; auf der anderen Seite verursacht eine zu frühe Fertigung Lagerkosten. Die Heuristik entspricht deshalb nicht den Anforderungen und ist für eine Lösung ungeeignet.

Zur Evaluierung von Cluster Algorithmen wurden drei Kategorien von Kriterien zur Gültigkeitsprüfung eines Clusters entwickelt: externe Kriterien, interne Kriterien und relative Kriterien. Externe und interne Kriterien beruhen auf statistischen Tests, die rechenintensiv durchgeführt werden müssen. Relative Kriterien hingegen wählen das beste resultierende Clusterergebnis anhand definierter Regeln. Dabei sind die drei Kategorien jedoch nicht gleichermaßen für alle Clustervarianten anwendbar.¹⁵¹ Aufgrund der gewählten Strategie ergeben sich spezifische Vor- und Nachteile, die untereinander abhängig und im Vorfeld bedingt unter Unsicherheit absehbar oder sogar nicht absehbar sind. Prioritätsregeln beschleunigen dabei die Entscheidungsfin-

¹⁴⁶ Vgl. [VS04]

¹⁴⁷ Vgl. [DSV97]

¹⁴⁸ Vgl. [Bet06]

¹⁴⁹ Vgl. [NEH83]

¹⁵⁰ Vgl. [FL03]

¹⁵¹ Vgl. [GMW07]

dung und sind für die Entwicklung eines Planungsverfahrens geeignet, um im Vorfeld eine Reihenfolge für die Auswahl von Aufträgen festzulegen.

Zusammenfassend ergibt sich folgendes Bild. Prioritätsregeln scheinen geeignet zu sein, um Aufträge nach festgelegten Kriterien auszuwählen. Die Rückwärtsterminierung scheint einen Beitrag zur kapazitätsorientierten Bedarfsplanung leisten zu können, weil auf diese Weise die Lagerkosten verringert werden können. Die Vorwärtsterminierung scheint einen Beitrag dazu leisten zu können, die vorgeplanten Bedarfe auf die Maschinen einzulasten.

3.2.2.4 Verfahren des Operations Research

Verfahren aus dem Bereich des OR werden im Folgenden im Hinblick darauf analysiert, ob sie in der Lage sind, einen Beitrag zu Ansätzen für die PPS zu leisten.

Optimierungsmodelle wie das MLCLSP¹⁵² und das PLSP-PM¹⁵³, die mit einem Solver wie CPLEX¹⁵⁴ oder MOPS¹⁵⁵ gelöst werden können, ermöglichen die Planung von Losgrößen und Terminen unter Beachtung von begrenzten Kapazitäten. Die Lösung gängiger Modelle braucht dabei nicht nur viel Zeit, sondern sie bilden auch nicht das in dieser Arbeit betrachtete Planungsproblem ab. Des Weiteren sind derartige Lösungen auf einen Single Best Solution Ansatz festgelegt, wobei sich die Rahmenbedingungen, unter denen eine Lösung ermittelt wurde, laufend ändern und ständige Neuplanungen hervorrufen. Außerdem wird der Planungshorizont begrenzt, weil das Scheduling-Problem sonst nicht in angemessener Zeit gelöst werden kann. Eine derartige künstliche Begrenzung des Planungshorizontes, mit der Vernachlässigung des Blickes über den Planungshorizont hinaus, verursacht suboptimale Reserven wie Lagerbestände und Maschinenkapazitäten. Optimierungsmodelle erfüllen damit nicht die an eine Lösung gestellten Anforderungen und werden nicht weiter für die PPS fokussiert.

Die Branch-and-Bound Heuristik versucht durch Backtracking eine optimale Lösung für ein Problem zu finden.¹⁵⁶ Eine erreichte Teillösung wird sukzessive zu einer Gesamtlösung aufgebaut. Der Suchraum wird partitioniert und das Ursprungsproblem

¹⁵² Vgl. [Tem06]

¹⁵³ Vgl. [KD97], [DH95]

¹⁵⁴ [Http://www.ilog.com/products/cplex](http://www.ilog.com/products/cplex)

¹⁵⁵ [Http://www.mops.fu-berlin.de](http://www.mops.fu-berlin.de).

¹⁵⁶ Vgl. [Hro04]

in Teilprobleme aufgeteilt, die jeweils einfacher gelöst werden können.¹⁵⁷ Das Verfahren wird in zwei Komponenten unterteilt:

- Verzweigen („Branching“)
Beim Verzweigen handelt es sich um das Unterteilen des Problems in zwei oder mehr Teilprobleme, die dann wiederum unterteilt werden können, so dass eine Baumstruktur entsteht.¹⁵⁸
- Beschränken („Bounding“)
Das Beschränken beschreibt das Abschneiden von Teilbäumen, die nicht mehr betrachtet werden müssen.¹⁵⁹

Das Branch-and-Bound Verfahren findet schnell eine optimale Lösung wenn viele Teilbäume abgeschnitten werden können. Im schlechtesten Fall muss dabei der gesamte Lösungsraum betrachtet werden. Die Leistungsfähigkeit des Verfahrens ist damit insbesondere von der Problemstruktur abhängig.¹⁶⁰ Für das Verzweigen des Problems existieren unterschiedliche Strategien, die die Performanz beeinflussen. Teilbäume können in bestimmten Fällen abgeschnitten werden, dabei existieren drei Möglichkeiten:¹⁶¹

- Optimalität.¹⁶²
- Schranke.¹⁶³
- Unlösbarkeit.¹⁶⁴

Übertragen auf die Produktionsplanung können Lösungen für Teilprobleme erzeugt werden, die später sukzessiv verfeinert werden. Diese Vorgehensweise scheint unter Beachtung der Anforderung geeignet zu sein, um einen Beitrag zur Lösung zu leisten.

¹⁵⁷ Vgl. [MF04]

¹⁵⁸ Vgl. [Hin75]

¹⁵⁹ Vgl. [Hin75]

¹⁶⁰ Vgl. [Wol98]

¹⁶¹ Vgl. [Wol98]

¹⁶² Ist die lineare Lösung eines Knotens ganzzahlig zulässig, muss dieser Knoten nicht weiter betrachtet werden „Abschneiden wegen Optimalität“.

¹⁶³ Wenn die untere Schranke eines Knotens größer ist, als die beste bisher gefundene zulässige Lösung, dann kann in diesem Knoten keine bessere Lösung mehr gefunden werden und muss nicht mehr betrachtet werden „Abschneiden wegen Schranke“.

¹⁶⁴ Ist das lineare Problem nicht lösbar, so ist auch das entsprechende ganzzahlige Problem nicht lösbar, daher wird ein unlösbarer Knoten nicht weiter betrachtet „Abschneiden wegen Unlösbarkeit“.

Weitere Verfahren des OR werden im Hinblick auf die PPS unter Unsicherheit in Kapitel 3.2.3 debattiert.

Zusammenfassend kann damit gesagt werden, dass Modelle des OR für eine Lösung des Problems ungeeignet sind. Anders hingegen scheint Branch and Bound einen Beitrag zur Lösung leisten zu können und es bieten sich Ansatzpunkte wie das „Beschränken“ und „Verzweigen“ für eine Weiterentwicklung im Hinblick auf die Anforderungen an eine Lösung.

3.2.2.5 Änderung eines Produktionsplans

Änderungsplanung beinhaltet, dass ein Produktionsplan existiert, der bereits teilweise ausgeführt wurde, wobei während der Ausführung Störungen auftreten, auf die ggf. reagiert werden muss.

Calhoun schlägt vor, um die Anzahl der Änderungen zu verringern, die Abweichungen vom ursprünglichen Masterplan zu minimieren. Dabei wird jede Abweichung mit Kosten bestraft und dem Nutzen gegenüber gestellt.¹⁶⁵ Allerdings wirken sich damit auch eigentlich kostenneutrale Änderungen auf die Kosten negativ aus. In der Produktionsplanung können die Kosten für eine Abweichung von einem Produktionsplan berechnet werden. Dabei kann die Änderung eines Produktionsplans durchaus auch kostenneutral erfolgen. Verbesserungspotential an einem bestehenden Produktionsplan geht aufgrund der festgelegten Strafkosten verloren. Damit scheint diese Vorgehensweise nicht für die Lösung des Problems geeignet zu sein.

Vergrößert ein Kunde seinen Bedarf, so besteht z.B. die Möglichkeit vorhandene Lose zu vergrößern. Abhängig von der gewählten Option wirken sich Auswirkungen auf Rüstzeitanteile oder Bestandsverläufe mehr oder weniger stark aus. Bei der operativen Änderung eines Produktionsplans existieren insbesondere vier Stellhebel¹⁶⁶:

- Alternative Maschinen.
- Losgrößen.
- Alternative Reihenfolgen.
- Kombination.

Sollten alternative Reihenfolgen für zwei oder mehr Bearbeitungsoperationen bestehen, kann so ein zeitweiliger Ausfall oder eine Belegung durch einen anderen Vorgang umgangen werden. Beispielsweise kann, falls hinsichtlich der Reihenfolge

¹⁶⁵ Vgl. [CDM+02]

¹⁶⁶ Vgl. [Wes05]

zweier Produktionsvorgänge keine Einschränkungen bestehen, für den einen Vorgang ein erst später vorgesehener anderer Vorgang vorgezogen werden. Die Strategien können dabei auch untereinander kombiniert werden. Die Ansätze eignen sich dabei gemäß den Anforderungen an eine Lösung als Ausgangspunkt für eine Rekombination mit dem Fokus auf der operativen Durchführung einer Änderung.

Busch analysiert in seiner Arbeit verschiedene Verfahren der Änderungsplanung für die PPS. Dabei stellt er fest, dass gängige Verfahren nicht auf der Grundlage einer monetären Zielfunktion optimieren und nur einen kleinen Lösungsraum betrachten.¹⁶⁷ Keines der diskutierten Verfahren erfüllt allerdings die an eine Lösung gestellten Anforderungen.

Ho verwendet einen Filtermechanismus, der unwichtige von wichtigen Datenänderungen trennt.¹⁶⁸ Carlson, Jucker und Kropp schlagen ebenfalls indirekte Filtermechanismen vor.¹⁶⁹ Somit wird nicht das Problem stochastisch auftretender Änderungen gelöst, aber die Auswirkungen bekannter Änderungen werden verringert. Diese Vorgehensweise scheint einen Beitrag zu einer Lösung leisten zu können und bietet sich als Ausgangspunkt für eine Weiterentwicklung an.

Zusammenfassend kann gesagt werden, dass bereits einige Ansätze für eine Änderungsplanung existieren, die allerdings einzeln die Anforderungen nicht erfüllen. Insbesondere fehlt eine klare Abgrenzung in welcher Situation mit welchen Methoden reagiert wird und eine Neuentwicklung bzw. die Rekombination bestehender Ansätze ist für eine Lösung des Problems erforderlich.

¹⁶⁷ Vgl. [Bus04]

¹⁶⁸ Vgl. [HCM+86], [Ho89]

¹⁶⁹ Vgl. [CJK79]

3.2.3 Ansätze für die Produktionsplanung und -steuerung unter Unsicherheit

Die Optimierung unter Unsicherheit gehört zu den schwierigsten Problemen.¹⁷⁰ Das Interesse fokussiert deshalb seit Jahren auf die Entwicklung heuristischer Lösungsansätze. Die wohl gierigste Heuristik „Hill Climbing“ akzeptiert keinen Zug, der nicht die beste bisher gefundene Lösung verbessert. Allerdings liefert diese Heuristik zumeist nur lokale Optima und kommt über diese auch nicht hinaus.¹⁷¹ Damit ist der Ansatz für das zu entwickelnde Planungsverfahren ungeeignet.

Eine weitere Lösungsmöglichkeit für stochastische Probleme stellt die Aggregation der Wahrscheinlichkeitsverteilungen zu einem konkreten Wert dar.¹⁷² Die Aggregation bedingt dabei, dass oft mit einem Wert gerechnet wird, der selbst niemals auftritt und eine Abweichung vom Optimum unausweichlich ist. Dennoch kann auf diese Weise eine Lösung bezüglich der erwarteten Eingabedaten ermittelt werden. Diese führt zwar keine konkrete Szenarioanalyse durch, aber dennoch werden Wahrscheinlichkeiten berücksichtigt. Damit ist dieser Ansatz im Sinne der Anforderungen bedingt geeignet.

Holland entwickelte die Genetischen Algorithmen¹⁷³, die neben der Evolutionsstrategie zu den populärsten Verfahren in der Gruppe der Evolutionären Algorithmen zählen und durch die Darwinistische Evolutionstheorie inspiriert sind.¹⁷⁴ Im Gegensatz zu anderen Optimierungsansätzen werden parallel mehrere initiale Lösungen erzeugt, die den Suchraum eingrenzen. Eine solche Lösungsmenge wird Population genannt.¹⁷⁵ Dieser Sachverhalt lässt sich auf die Produktionsplanung übertragen, indem mehrere Produktionspläne erzeugt werden. In weiteren Schritten verwendet der Algorithmus die ursprünglichen Lösungen, um neue untergeordnete Lösungen zu erzeugen. Hierbei unterstützen die genetischen Operatoren Mutation¹⁷⁶ und Rekombination.¹⁷⁷ Aus der

¹⁷⁰ Vgl. [Sen00]

¹⁷¹ Vgl. [ASN02]

¹⁷² Anschließend erfolgt die Lösung für den ermittelten deterministischen Wert.

¹⁷³ Der Genetische Algorithmus ist ein stochastisches paralleles Suchverfahren, das von den Prozessen in der Natur inspiriert ist [GMW07]. Genetische Algorithmen eignen sich insbesondere zur Lösung komplexer Optimierungsprobleme und wenden Strategien aus der Evolutionstheorie an, um die optimale Lösung zu ermitteln [GKK04], [SHF94].

¹⁷⁴ Vgl. [ZB05]

¹⁷⁵ Vgl. [VW06]

¹⁷⁶ Eine Mutation stellt dabei die lokale Transformation einer bereits gefundenen Lösung dar [Hro04].

¹⁷⁷ Vgl. [ASN02]

erzeugten Lösungspopulation werden die besten Lösungen nach der Strategie „Survival of the Fittest“ ausgewählt. Anschließend wird wieder eine neue Population erzeugt. Dieser Prozess setzt sich über mehrere Generationen fort, bis eine Population nur noch gleiche oder ähnliche Lösungen beinhaltet oder eine vorbestimmte Laufzeit erreicht ist.¹⁷⁸ Anschließend wird die beste ermittelte Lösung ausgewählt.¹⁷⁹

Mehrere Faktoren¹⁸⁰ beeinflussen die Laufzeit und die Lösungsgüte eines Genetischen Algorithmus. Der wahrscheinlich größte Vorteil von Genetischen Algorithmen ist, dass anstatt eine einzelne Lösung zu ermitteln, ein ganzes Lösungsbündel erstellt wird.¹⁸¹ Dies wäre auch für die Produktionsplanung ein gewaltiger Vorteil gegenüber bestehenden Lösungen, weil auf die Weise unter mehreren potentiellen Produktionsplänen ein Masterplan ausgewählt werden kann. Ein weiterer Vorteil ist die einfache Parallelisierbarkeit. Dabei besteht ein wesentlicher Ansatz in der Aufteilung einer Population in mehrere Subpopulationen, auf denen parallel der Genetische Algorithmus ausgeführt wird. Ein weiterer Vorteil ist die Robustheit. Wird die Schwankung der Ergebnisse bei wiederholter Ausführung betrachtet, weisen Genetische Algorithmen nur geringe Abweichungen auf. Damit scheinen Genetische Algorithmen einen Beitrag zur Lösung im Sinne der Anforderung leisten zu können.

Simulated Annealing¹⁸² ist ein heuristisches Optimierungsverfahren.¹⁸³ Es wird bei Problemen¹⁸⁴ angewendet, die durch eine hohe Komplexität das vollständige Ausprobieren aller Möglichkeiten oder die Anwendung mathematischer Verfahren ausschließen.¹⁸⁵ Die Grundidee besteht darin, dass eine Veränderung der Ausgangslösung auf jeden Fall akzeptiert wird, wenn sie zu niedrigeren Kosten führt.¹⁸⁶

¹⁷⁸ Vgl. [VW06]

¹⁷⁹ Vgl. [GKK04]

¹⁸⁰ Die Effizienz eines Genetischen Algorithmus beeinflussen: die Größe der Population, die Anzahl durchlaufener Generationen, die Repräsentation der Individuen, die Auswahl der Anfangspopulation und die Funktionen zur Bewertung, Selektion, Rekombination sowie Mutation [Hro04], [SHF94].

¹⁸¹ Vgl. [Hro04]

¹⁸² Simulated Annealing ist auch bekannt als Monte Carlo Annealing.

¹⁸³ Vgl. [ASN02]

¹⁸⁴ Zu den Einsatzgebieten gehören unter anderem die Terminplanung, die Maschinenbelegung und die Tourenplanung [ZB05].

¹⁸⁵ Vgl. [Wen95]

¹⁸⁶ Im Falle einer Kostensteigerung wird dagegen die Lösung nur mit einer vom Ausmaß der Kostenerhöhung abhängigen Wahrscheinlichkeit akzeptiert [Wen95].

Die Option der Akzeptanz auch von kurzfristig schlechteren Lösungen ermöglicht es, lokale Minima zu verlassen.¹⁸⁷ Simulated Annealing ist ein für verschiedene Probleme anwendbares Verfahren und liefert gute Lösungen bei einer relativ leichten Implementation.¹⁸⁸ Ein Vorteil ist, dass die Akzeptanzwahrscheinlichkeit es ermöglicht ausgehend von einem lokalen Optimum, das globale Optimum zu finden.¹⁸⁹ Von Nachteil ist der riesige Rechenaufwand. Deshalb ist es bisher für die Praxis zu langsam und wird vor allem zur Verbesserung bereits existierender Lösungen verwendet.¹⁹⁰ Um die Laufzeit zu verbessern, können zu Lasten der Lösungsqualität Abbruchbedingungen formuliert werden.¹⁹¹ Für die Produktionsplanung unter Unsicherheit scheint Simulated Annealing damit für die Verbesserung eines bereits gültigen Produktionsplans geeignet zu sein.

In letzter Zeit erweckten Samplingmethoden großes Interesse zur Lösung von Problemen¹⁹², insbesondere für solche Probleme, für deren deterministische Variante polynomielle Approximationsverfahren existieren.¹⁹³ Samplingmethoden sind jedoch keine Lösungsverfahren, sondern viel mehr werden durch Sampling nur wenige Beispiel-Szenarien in einem Lösungsverfahren betrachtet, anstatt aller Szenarien.¹⁹⁴ Ein Vorteil von Samplingmethoden ist dabei, dass sie auch angewandt werden können, wenn die Szenarien nicht ohne weiteres aufgezählt werden können. Es reicht aus, wenn es möglich ist, Beispielszenarien zu generieren.

Linderoth unterscheidet zwischen zwei Samplingansätzen.¹⁹⁵ Zum einen existiert der „interne“ Samplingansatz. Dieser wird von Algorithmen verwendet, die das ursprüngliche Problem lösen und dabei immer, wenn Informationen über die stochastischen Variablen benötigt werden, ein neues Beispielszenario generieren. Die Stochastische Dekomposition von Higle und Sen ist ein Schichtenverfahren und verwendet eine interne Samplingmethode.¹⁹⁶

¹⁸⁷ Vgl. [ZB05]

¹⁸⁸ Vgl. [Hro04]

¹⁸⁹ Vgl. [ZB05]

¹⁹⁰ Vgl. [Hro04]

¹⁹¹ Vgl. [VW06]

¹⁹² Vgl. [KSH01]

¹⁹³ Vgl. [SS04]

¹⁹⁴ Vgl. [HS96]

¹⁹⁵ Vgl. [LSW06]

¹⁹⁶ Vgl. [HS96]

Zum anderen generieren „externe“ Samplingmethoden im Vorfeld Beispielszenarien, wobei der Erwartungswert geschätzt wird. Das Problem kann entweder mit einem diskreten Algorithmus oder einem speziellen Algorithmus für stochastische Probleme gelöst werden. Damit scheint ein externer Samplingansatz geeignet zu sein, um Szenarien zu generieren.

Ein weiterer Ansatz zur PPS unter Unsicherheit ist die Spielbaumsuche¹⁹⁷, die ein Spiel aus der Perspektive eines Spielers betrachtet. Der Ausgang des Spiels ist deterministisch und hängt nur von den Zügen der Spieler ab.¹⁹⁸ Das Ziel ist es, eine Strategie zu finden, die mit hoher Wahrscheinlichkeit zum Gewinn führt. Ein Spieler weiß dabei nicht genau welche Züge sein Gegner in der Zukunft ausführt. Die optimale Strategie für ein solches Spiel kann z.B. mit dem Minimax Algorithmus bestimmt werden.¹⁹⁹ Der Suchraum kann entweder der vollständige Spielbaum des Spiels sein, oder aber nur ein Teilbaum. Für einfache Spiele wie Tic-Tac-Toe ist es möglich, den gesamten Spielbaum zu durchsuchen. Für komplexere Spiele wie Schach können dagegen nur Sequenzen untersucht werden.²⁰⁰ Eine Erweiterung des Minimax Algorithmus ist der Alpha-Beta Algorithmus.²⁰¹ Hier wird versucht, Teile des Suchraumes aufgrund bisheriger Erkenntnisse abzuschneiden und den Suchaufwand auf diese Weise zu verringern. Damit ist die Anwendung der Spielbaumsuche auf die PPS unter Unsicherheit potentiell geeignet, um Produktionspläne zu analysieren.

In Erweiterung zu traditionellen Lösungsverfahren wurde von Lorenz ein Ansatz zur mehrstufigen Entscheidung unter Unsicherheit²⁰² entwickelt.²⁰³ Dieser liegt im Schnittfeld von Entscheidungstheorie, Operations Research, Robuster Optimierung²⁰⁴ und Künstlicher Intelligenz²⁰⁵. Für die Modellierung und Berechnung stochastischer Planungsprobleme wird ein Reparaturspiel verwendet, in dem ein mehrstufiges

¹⁹⁷ Spielbäume können als einfache Bäume dargestellt werden. Die Knoten stellen dabei die Spielzustände dar und die Kanten die möglichen Züge. Alle Kanten einer Ebene stellen die Möglichkeiten eines Spielers da. Dabei können die Knoten als Zustände in der Produktion angesehen werden und die Kanten als mögliche Produktionspläne.

¹⁹⁸ Vgl. [Nil82]

¹⁹⁹ Vgl. [Nil82]

²⁰⁰ Vgl. [Nil82]

²⁰¹ Vgl. [KM75], [Nil 82]

²⁰² Vgl. [HBH88], [RS01]

²⁰³ Vgl. [EGH+03], [EGL05]

²⁰⁴ Vgl. [Sch01]

²⁰⁵ Vgl. [RN03]

ganzzahliges Entscheidungsproblem als Spiel²⁰⁶ aufgefasst wird. Dabei wird bei dem untersuchten Planungsproblem davon ausgegangen, dass die unsichere zukünftige Umgebung mit Hilfe von diskreten Szenarien modelliert werden kann. Die Szenarien können dann mit Hilfe einer Baumsuche, die simulativen Charakter hat, systematisch untersucht werden. Ein Spielbaum repräsentiert dabei sämtliche möglichen Spielverläufe, die sich aus der aktuellen Situation ergeben können. Die Auswertung erfolgt unter anderem mit Hilfe eines erweiterten Minimax-Prinzips. Die Algorithmische Grundlage für solche Spiele bilden der Alphabeta-Algorithmus²⁰⁷ und dessen Negascout-Variante²⁰⁸.

Sequentielle und parallele Baumsuchen wurden in der Praxis bereits in Programmen für verschiedene Problemstellungen unter Unsicherheit eingesetzt. Beispielsweise beschreiben Donninger, Kure und Lorenz ein hardware-unterstütztes paralleles Schachprogramm, das eine Alphabeta-Suche verwendet.²⁰⁹ Auf Conspiracy-Suche basiert ein von Lorenz beschriebenes Schachprogramm.²¹⁰ Für Problemstellungen in der Produktionsplanung ist die exakte Auswertung eines Reparaturspielbaums wegen seiner gigantischen Größe nicht praktikabel. Erfahrungen mit Spielbaumsuche zeigen, dass auch heuristisch beschränkte Spielbäume gute Lösungen liefern.

Ehrhoff, Grothklags und Lorenz untersuchten das stochastische Fleetassignment-Problem²¹¹, das in der Komplexität ähnlich zu dem in dieser Arbeit untersuchten Problem ist. Sie zeigen, dass eine heuristische Auswertung, bei der die Größe des Suchbaums in Tiefe und Breite stark eingeschränkt wird, gute Ergebnisse liefert. Mit wachsender Baumgröße verbessert sich auch die Qualität der Lösungen. Zwischen dem Fleetassignment-Problem und der Produktionsplanung existieren jedoch gravierende Unterschiede. Ein Produktionsvorgang kann im Notfall auch durch einen anderen Produktionsvorgang unterbrochen und später fortgesetzt werden, was im Fleetassignment unmöglich ist, weil ein Flugzeug erst landen muss. Des Weiteren müssen in der Produktionsplanung Planänderungen nicht pauschal mit einem festen Betrag bestraft werden, weil die Auswirkungen von Planänderungen berechnet werden können. Eine Anwendung dieses Konzepts auf die Produktionsplanung scheint es zu ermöglichen, einen Beitrag zur PPS unter Unsicherheit zu leisten.

²⁰⁶ Vgl. [EGL04], [Pap83], [MWG95]

²⁰⁷ Vgl. [KM75]

²⁰⁸ Vgl. [Rei83]

²⁰⁹ Vgl. [DKL04]

²¹⁰ Vgl. [Lor00]

²¹¹ Vgl. [EGL04]

Zusammenfassend ergibt die Analyse, dass kein Ansatz die Anforderung an eine Lösung vollständig erfüllt. Jedoch bieten sich mit den Genetischen Algorithmen, dem Reparaturspiel, Sampling und Simulated Annealing Ideen für eine Rekombination bestehender Ansätze und deren Weiterentwicklung mit dem Ziel, eine neuartige Vorgehensweise zur PPS unter Unsicherheit zu konzipieren.

4 Zu leistende Arbeiten

„Man kann die raffiniertesten Computer der Welt benutzen und alle Diagramme und Zahlen parat haben, aber am Ende muss man alle Informationen auf einen Nenner bringen, muss einen Zeitplan machen und muss handeln.“

Lee Iacocca (*1924)

Basierend auf der Strukturierung der Problemstellung in Kapitel 2.4 und den in Kapitel 2.5 formulierten Anforderungen wurde der Stand der Technik analysiert. In Kapitel 4 werden die zur Erreichung der Ziele noch verbleibenden Aufgaben abgeleitet.

4.1 Verbleibende Aufgaben für das Benchmarking von Planungsverfahren

Um die Objektivität zu erhöhen, erfolgt die strikte Aufteilung in Planungsproblem²¹², Simulation des Produktionsablaufs²¹³ und Planungsverfahren²¹⁴.

4.1.1 Verbleibende Aufgaben für die Modellierung des Planungsproblems

Ausgehend von der Analyse des Stands der Technik in Kapitel 3.1.1 wird deutlich, dass kein Modell das Planungsproblem vollständig abbildet, aber umgekehrt eine Kombination der Vielzahl von Ansätzen nahezu das Planungsproblem abbildet.

Aufgrund der diskutierten Schwächen bestehender Modelle ist deren Rekombination und Erweiterung nötig. Hierzu wird ein auf bestehenden Ansätzen basierendes adaptives Modell entwickelt, das das betrachtete Planungsproblem formalisiert. Dabei werden potentiell mehrere Planungsprobleme abgedeckt und nicht nur das in dieser Arbeit spezifizierte Problem. Dazu wird der Betrachtungshorizont definiert und ein deterministisches Optimierungsmodell, so allgemein wie möglich, aber so speziell wie nötig, formalisiert. Diesem soll ein Small-Bucket Modell zugrunde liegen und es muss

²¹² Vgl. Kapitel 4.1.1

²¹³ Vgl. Kapitel 4.1.2

²¹⁴ Ein Planungsverfahren wird an dieser Stelle nicht entwickelt, aber eine Möglichkeit dafür geschaffen, dass verschiedene Planungsverfahren sich für die Simulation des Produktionsablaufs über eine Schnittstelle anmelden können (vgl. Kapitel 4.1.2).

die für die Produktion im Stand der Technik debattierten wesentlichen Restriktionen umsetzen. So weit wie möglich wird das Modell linear formuliert.

Die Zielfunktion des Optimierungsmodells wird gegenüber bestehenden Modellen um drei Summanden erweitert. Die Verzögerung eines Auftrags wird bestraft, wobei hier insbesondere auch die Dauer der Verzögerung gewichtet wird. Ebenso wird der Totalausfall eines Auftrags erfasst. Des Weiteren fallen für leer stehende Maschinen Opportunitätskosten an.

Darüber hinaus gilt es, stochastische Störungen des Produktionsablaufs zu berücksichtigen. Das deterministische Optimierungsmodell wird hierzu zu einem stochastischen Modell erweitert, indem Parameter die stochastischen Einflüssen unterliegen, auch als stochastische Parameter abgebildet werden. Jede unsichere Eingabegröße des Optimierungsmodells wird deshalb um eine Wahrscheinlichkeitsverteilung erweitert. Es werden stochastische Kundenbedarfe, Störungen an Maschinen sowie schwankende Produktionsgeschwindigkeiten abgebildet. Dabei werden Störungen an Maschinen und schwankende Produktionsgeschwindigkeiten gebündelt erfasst.

4.1.2 Verbleibende Aufgaben für die Simulation des Produktionsablaufs

Die Analyse des Stands der Technik in Kapitel 3.1.2 macht deutlich, dass kaum Ansätze existieren, die die formulierten Anforderungen erfüllen. Aufgrund dieser Einschätzung wird eine Benchmarkplattform (vgl. Definition 12) entwickelt.

Definition 12: Benchmarkplattform

Unter Benchmarkplattform wird in dieser Arbeit eine Umgebung verstanden, die es ermöglicht Planungsverfahren der PPS untereinander objektiv zu vergleichen. Die Benchmarkplattform setzt sich dabei aus drei Komponenten zusammen: Produktionsdatenbank, Simulator und der Schnittstelle für Planungsverfahren²¹⁵.

Dazu wird das Planungsproblem in Form eines Optimierungsmodells formalisiert²¹⁶ und von dem Simulator implementiert. Damit erfolgt eine Trennung von Planungsproblem

²¹⁵ Die Schnittstelle für Planungsverfahren wird im Folgenden auch Optimierungseingabe genannt.

²¹⁶ Vgl. Kapitel 4.1.1

und Simulation des Produktionsablaufs und die Umgebung wird auf diese Weise so allgemein gehalten, dass sie für verschiedene Planungsprobleme einsetzbar ist.

Die Optimierungseingine soll es ermöglichen, dass parallel mehrere Planungsverfahren sich an dem Simulator anmelden können. Dabei wird ein Planungsproblem über die Optimierungseingine ausgewählt und die Simulation des Produktionsablaufs in dem Simulator gestartet. Der Simulator liest dazu die Kundenbedarfe, die Maschinendaten sowie den Arbeitsplan aus der zentralen Produktionsdatenbank, in der alle benötigten Daten gespeichert werden, und schickt diese über die Optimierungseingine an das Planungsverfahren. Auf die Datenbank kann dabei nur der Simulator direkt lesend und schreibend zugreifen. Die Daten werden so abgelegt, dass die Datenbank für verschiedene Planungsprobleme einsetzbar ist.

Über die Optimierungseingine schickt das Planungsverfahren einen erzeugten Produktionsplan an den Simulator. Der Produktionsplan wird im Simulator umgesetzt und der Produktionsablauf unterliegt in der Simulation stochastischen Störungen. Verfahren müssen zeigen, wie gut die erstellten Pläne von Anfang an waren und wie gut sich ein bestehender Produktionsplan an neue Situationen anpassen lässt.

Dabei wird die Produktion als ein rundenbasiertes Spiel abgebildet, an dem zwei Spieler partizipieren. Der erste Spieler „Optimierer“ versucht, einen optimalen Produktionsplan zu finden. Der zweite Spieler „Natur“ determiniert Auftragseingänge und Produktionsgeschwindigkeiten aus Wahrscheinlichkeitsverteilungen. Die in der Produktion angefallenen Ist-Kosten werden gemäß der entwickelten Zielfunktion berechnet. Ebenso wird die Einhaltung der aufgestellten Restriktionen geprüft. Anschließend wird ein Update mit den eingetretenen Änderungen an den Optimierer geschickt, der mit einer Änderung des Produktionsplans reagieren kann.

Wie im Stand der Technik analysiert, lassen sich die Verfahren am Ende der Simulation anhand des besten bisher gefundenen Verfahrens einordnen. Dies ist auch möglich ohne sicher zu sein, dass die optimale Lösung als Referenzwert gefunden wurde.

Abbildung 2 illustriert das Zusammenspiel der drei Komponenten der Benchmarkplattform Produktionsdatenbank, Simulator und Optimierungseingine.

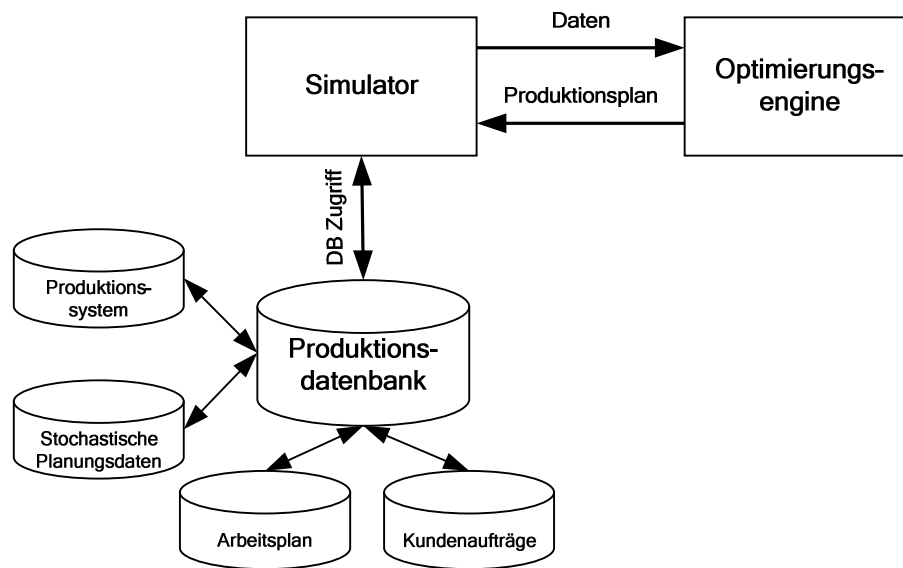


Abbildung 2: Benchmarkplattform

Stochastische Störungen des Produktionsablaufs bieten insbesondere zwei Vorteile gegenüber der Nutzung von Stördatenprofilen aus der Vergangenheit. Zum einen wird auf diese Weise die Qualität verschiedener Ansätze mit Hilfe statistischer Methoden verglichen. Dabei ist es das Ziel, durch die Modellierung der Störungen als Zufallsereignisse, die an echte Primärbedarfsprofile gekoppelt sind, realitätsnahe Aussagen zu erreichen. Zum anderen sind zufällige Störungen nicht vorhersagbar. Auf diese Weise wird verhindert, ein Planungsverfahren nur auf die gegebenen Daten zu optimieren, denn ein solches Verfahren wäre vermutlich nur für das spezielle Szenario geeignet.

4.2 Verbleibende Aufgaben zur Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie

4.2.1 Verbleibende Aufgaben für Planungsstrategien

Ausgehend von den in Kapitel 2 formulierten Anforderungen an eine Problemlösung und der Analyse des Stands der Technik in Kapitel 3.2.1 wird deutlich, dass bereits eine Vielzahl von Planungsstrategien existieren, die allerdings nur in Ansätzen für die Lösung des betrachteten Problems geeignet sind.

Deshalb werden neue Planungsstrategien konzipiert, die bestehende Ansätze vereinen und erweitern sowie einen Beitrag zur PPS und zur PPS unter Unsicherheit leisten.

4.2.2 Verbleibende Aufgaben für die Produktionsplanung und -steuerung

Die Analyse des Stands der Technik in Kapitel 3.2.2 macht deutlich, dass bereits viele Ansätze zur PPS existieren. Allerdings eignet sich allein genommen keiner der Ansätze, das Problem unter den in Kapitel 2 formulierten Anforderungen zu lösen. Die im Stand der Technik diskutierten bestehenden Ansätze werden deshalb rekombiniert und neue Planungsansätze entworfen.

Neue Planungsansätze sollen einen Beitrag dazu leisten, für deterministische Werte einen Produktionsplan zu erstellen und auf Störungen des Produktionsablaufs schnell und effizient zu reagieren. Die Planungsansätze basieren dabei auf den entwickelten Planungsstrategien²¹⁷ und werden im Folgenden klassifiziert.

Die Erstellung eines Produktionsplans soll für deterministische Planungsdaten schnell einen gültigen Produktionsplan berechnen. Die Analyse des Stands der Technik zeigt, dass die Vernachlässigung von Kapazitätsgrenzen bei der Nettobedarfsermittlung bereits vor der eigentlichen Einplanung zu absehbaren Konflikten führt. Das neue Planungsverfahren soll deshalb vorausschauend Konflikten vorbeugen und bereits bei der Bedarfsermittlung die Kapazitäten beachten. Die Verwendung von Prioritätsregeln, für die Auswahl von Aufträgen, spart Rechenzeit und deshalb werden im Vorfeld einer Planung Prioritätsregeln aufgestellt, die später Entscheidungen steuern. Um die Lagerkosten möglichst gering zu halten, werden Rüstzeiten vorerst nicht berücksichtigt und darauf geachtet, einen Bedarf in Bezug auf die Lagerkosten, zu einem gültigen Termin einzuplanen.

Tritt bei der Einplanung ein Konflikt auf, so wird versucht, diesen durch sukzessives Durchlaufen eines Regelwerks zu lösen. Misslingt die Konfliktbehebung trotz aller Bemühungen, wird ein Konzept entwickelt, nach dem in diesem Ausnahmefall Aufträge ausgewählt und ggf. zurückgestellt werden.

Alle Änderungen an einem bestehenden Produktionsplan, z.B. durch ein Fortschreiten des Horizonts oder eine Bedarfssenkung, werden durch eine vorhergehende Änderungsplanung eingeleitet. Bestehende Ansätze werden rekombiniert und Situationen in denen eine Änderungsplanung notwendig ist klassifiziert. Durch den Einsatz von Toleranzgrenzen soll dabei vermieden werden, dass jede Änderung eines Bedarfes zugleich eine Änderungsplanung auslöst und Planungsnervosität verursacht.

Um Konflikte prophylaktisch zu vermeiden, wird ein Beitrag dazu geleistet, kritische Engpässe proaktiv zu puffern, indem Bestände aufgebaut werden.

²¹⁷ Vgl. Kapitel 4.2.1

Ein bestehender Produktionsplan soll auch während der Ausführung verbessert werden. Dabei wird versucht, alle zur Verfügung stehenden Potentiale bestmöglich zu nutzen.

Des Weiteren soll ein Beitrag dazu geleistet werden, das Potential Kapazität bestmöglich zu nutzen.

4.2.3 Verbleibende Aufgaben für die Produktionsplanung und -steuerung unter Unsicherheit

Ausgehend von den in Kapitel 2 formulierten Anforderungen an eine Problemlösung und der Analyse des Stands der Technik in Kapitel 3.2.3 wird deutlich, dass keiner der bestehenden Ansätze zur PPS unter Unsicherheit das Problem vollständig lösen kann. Deshalb ist die Rekombination von bestehenden Ansätzen und deren Weiterentwicklung erforderlich.

Um das Produktionssystem langfristig bestmöglich aufzustellen, wird ein gültiger optimierter Produktionsplan erstellt. Dieser darf dabei nicht nur für ein konkretes Szenario optimiert sein, sondern muss vorher mehrere Szenarien betrachten und Optionen offen halten, um auf zukünftige Änderungen zu reagieren.

Dazu wird versucht, aus Kenntnissen der Umwelt bestmögliche Ergebnisse zu erzielen. Hierfür wird ein Spielbaum konzipiert, der mehrere Szenarien betrachtet. Dabei muss beachtet werden, dass in der Praxis die vollständige Auswertung eines Spielbaums, der sich aus einem praxisrelevanten stochastischen Planungsproblem ergibt, wegen seiner Größe nicht möglich ist. Der Spielbaum wird sowohl in der Tiefe und damit zeitlich, als auch in der Breite und damit auf ausgewählte Störungen, beschränkt. Die Analyse des Stands der Technik zeigt, dass eine Beschränkung des Lösungsraums in der Breite und in der Tiefe die Lösungsqualität wenig verringert.

Dabei erfolgen die Analyse von unterschiedlichen Kapazitätsniveaus und die Analyse unterschiedlicher Produktionspläne sowie die Gewichtung von Planungsperioden. Ergebnis der Analyse ist die Festlegung der zur Verfügung stehenden Kapazität für die übernächste Woche und die Auswahl eines Produktionsplans als Masterplan.

5 Umsetzung

„In der Wissenschaft gleichen wir alle nur den Kindern, die am Rande des Wissens hie und da einen Kiesel aufheben, während sich der weite Ozean des Unbekannten vor unseren Augen erstreckt.“

Isaac Newton (*1643)

Kapitel 5 führt die in Kapitel 4 herausgestellten zu leistenden Arbeiten aus, wobei sich die Gliederung an der Strukturierung des Problems aus Kapitel 2.4 orientiert.

In Kapitel 5.1 werden dazu die verbleibenden Aufgaben für das Benchmarking von Planungsverfahren gelöst und Kapitel 5.2 füllt die Lücken im Stand der Technik zur PPS in der mehrstufigen auftragsbezogenen Serienfertigung in der Automobilzulieferindustrie. Die Gliederung in den einzelnen Teilproblemen ist individuell an die spezifischen Erfordernisse angepasst und wird vor Beginn des jeweiligen Kapitels entwickelt.

5.1 Ein Beitrag für das Benchmarking von Planungsverfahren

5.1.1 Ein Beitrag für die Modellierung des Planungsproblems

Planungsproblem, Simulation des Produktionsablaufs und Planungsverfahren werden entkoppelt. Dazu wird im Folgenden das in dieser Arbeit betrachtete Planungsproblem für das Benchmarking von Planungsverfahren modelliert und in der Tradition mathematischer Modelle formalisiert. Dabei stellt das adaptive Modell kein allgemein auf jedes Unternehmen in exakt dieser Form übertragbares Modell dar, sondern bildet vielmehr den Ausgangspunkt für eine unternehmensspezifische Adaption.

In Kapitel 5.1.1.1 wird ein deterministisches Modell konzipiert, das in Kapitel 5.1.1.2 zu einem stochastischen Modell weiterentwickelt wird.

5.1.1.1 Entwicklung des deterministischen Modells

Für die Entwicklung eines Modells wird in Kapitel 5.1.1.1.1 zuerst der Betrachtungshorizont des Planungsproblems klassifiziert. Die Parameter und Variablen des Modells werden in Kapitel 5.1.1.1.2 formalisiert. In Kapitel 5.1.1.1.3 wird die Kostenfunktion entwickelt und in Kapitel 5.1.1.1.4 werden die Restriktionen des Modells aufgestellt.

5.1.1.1.1 Betrachtungshorizont

Für die Entwicklung eines Modells wird zuerst der Betrachtungshorizont klassifiziert, wobei T_0 die Startzeit und T_e die Endzeit im formalen Zeitmodell darstellt. T_i ist ein Zeitpunkt zwischen T_0 und T_e , wobei die Zeit in Perioden eingeteilt ist. Ereignisse, die vor dem heutigen Zeitpunkt liegen, gehören bereits der Vergangenheit an und unterliegen keinen Schwankungen mehr. Die entstandenen Ist-Kosten können exakt gemessen werden.

Ereignisse, die von heute ausgehend in der Zukunft liegen, werden in deterministische, stochastische und unbekannte Daten unterteilt. Deterministische Daten unterliegen keinen Schwankungen, sind sicher und ihr Wert ist fest. Hingegen unterliegen stochastische Daten Schwankungen, wobei Wahrscheinlichkeitsverteilungen bekannt sind. Ereignisse die weit in der Zukunft liegen, sind hingegen vollständig unbekannt. Abbildung 3 verdeutlicht den Zusammenhang.

Vergangenheit	Jetzt	Deterministische Daten	Stochastische Daten	Unbekannte Daten
---------------	-------	------------------------	---------------------	------------------

Abbildung 3: Klassifikation des Betrachtungshorizonts

5.1.1.1.2 Parameter und Variablen

Im Folgenden werden in Tabelle 6 die für die Entwicklung des Modells relevanten Parameter illustriert und in Tabelle 7 werden die Variablen des deterministischen Optimierungsmodells zusammengefasst. Die Parameter und Variablen bilden in Verbindung miteinander die Ausgangsbasis für eine systematische Entwicklung des Modells.

Parameter

Symbol	Definition
a_{pq}	Direktbedarfskoeffizient bezüglich Produkt p und q. Der Wert ist Null falls Produkt q kein unmittelbarer Nachfolger von Produkt p ist, andernfalls gibt a_{pq} die Menge von Produkt p an, die benötigt wird, um Produkt q herzustellen.
B_p	Kosten für das Zurückstellen (Backlogging) einer Einheit des Produktes p für eine Periode. Der Bedarf in einer bestimmten Periode kann auch noch durch Produktion in einer späteren Periode gedeckt werden.
C_{it}	maximale Produktionsmenge von Produkt P_i mit Aufgabe i während Periode t.

D_{pt}	externer Bedarf nach Produkt p in Periode t .
I	Menge der Aufgaben.
I_i	Menge der Aufgaben, die die gleiche Maschine wie Aufgabe i benutzen.
M	Menge der Maschinen.
M_i	Maschine, die von Aufgabe i benötigt wird.
O_p	Kosten für das Nichterfüllen eines externen Bedarfs (Stockout) für eine Einheit von Produkt p .
P	Menge der Produkte.
P_i	Produkt, das von Aufgabe i produziert wird.
Q_i	Produktionskosten für eine Einheit von Produkt P_i mit Aufgabe i .
R_{ij}	Anzahl der Perioden, die ausschließlich für das Rüsten von Aufgabe i zu Aufgabe j benötigt werden.
R'_{ij}	$= T_{ij} - R_{ij}$; fraktionaler Anteil der ($R_{ij}+1$ sten)-Periode nach dem Beginn des Umrüstens von Aufgabe i auf Aufgabe j .
S_p	Lagerhaltungskosten, die anfallen, wenn Produkt p eine Periode im Lager gehalten wird.
T	Anzahl der Perioden.
T_{ij}	Rüstzeit um von Aufgabe i auf Aufgabe j zu rüsten.
V_m	Opportunitätskosten für ungenutzte Kapazität der Maschine m .
X_{ij}	Rüstkosten die anfallen wenn von Aufgabe i zu Aufgabe j gerüstet wird (auf Maschine $M_i = M_j$).
Δ_{pi}	ganzzahlige Anzahl von Perioden für den Transport von Produkt p zu Maschine M_i (für Aufgabe i).

Tabelle 6: Entwicklung der Parameter des deterministischen Optimierungsmodells

Variablen

Symbol	Definition
b_{pt}	Zurückstellen (Backlogging) eines Produktes p am Ende der Periode t .
o_{pt}	Nichterfüllen des Bedarfes nach Produkt p am Ende der Periode t .
q_{it}	Produktionsmenge von Aufgabe i in Periode t .
s_{pt}	Lagerbestand von Produkt p am Ende der Periode t .
v_{mt}	Binärvariable, die anzeigt, ob die Maschine m in Periode t leer steht.
x_{ijt}	Binärvariable, die anzeigt, ob eine Umrüstung von Aufgabe i zu Aufgabe j am Anfang der Periode t erfolgt ($x_{ijt} = 1$) oder nicht ($x_{ijt} = 0$).
y_{it}	Binärvariable, die anzeigt, ob eine Maschine M_i für Aufgabe i am Ende der Periode t gerüstet ist ($y_{it} = 1$) oder nicht ($y_{it} = 0$).

Tabelle 7: Entwicklung der Variablen des deterministischen Optimierungsmodells

5.1.1.1.3 Kostenfunktion

Im Ausnahmefall sind die verspätete Fertigstellung und der Totalausfall eines Auftrages erlaubt, werden aber mit hohen Kosten bestraft.

Den Rüstzustand einer Maschine zu verändern, verursacht Rüstzeiten und Rüstkosten. Diese hängen dabei jeweils vom neuen und alten Zustand ab und werden entsprechend in der Kostenfunktion erfasst.

Erzeugnisse, die in einer Periode produziert werden, um einen zukünftigen internen oder externen Bedarf zu erfüllen, müssen gelagert werden und verursachen produktspezifische Lagerkosten, die periodenweise berechnet werden.

Für ungenutzte Maschinenkapazitäten fallen Opportunitätskosten an und die Produktion eines Produkts verursacht Produktionskosten, die es jeweils in der Kostenfunktion zu erfassen gilt.

Dabei wird die Höhe der Kosten in der speziellen Produktionsdatenbank²¹⁸ festgelegt. Auf dieser Grundlage und den in Kapitel 5.1.1.1.2 eingeführten Parametern und Variablen wird die Kostenfunktion des deterministischen Modells definiert:

$$\begin{aligned} \text{Min } Z_{D\text{-}MLCLS\text{-}PM} = & \sum_{t=1}^T \left(\sum_{p \in P} (S_p \cdot s_{pt} + B_p \cdot b_{pt} + O_p \cdot o_{pt}) + \sum_{\substack{i \in I \\ j \in I_i}} X_{ij} \cdot x_{ijt} + \right. \\ & \left. \sum_{i \in I} Q_i \cdot q_{it} + \sum_{m \in M} V_m \cdot v_{mt} \right) \end{aligned} \quad (1.1)$$

Durch die Kostenfunktion (1.1) wird sichergestellt, dass über alle Perioden die Lagerhaltungskosten ($S_p \cdot s_{pt}$) für alle Produkte, plus die Kosten für den Totalausfall ($O_p \cdot o_{pt}$) eines Produktes, plus die Kosten für die verspätete Fertigstellung ($B_p \cdot b_{pt}$) eines Produktes, plus die Rüstkosten ($X_{ij} \cdot x_{ijt}$), plus die Produktionskosten ($Q_i \cdot q_{it}$) sowie die Kosten für ungenutzte Kapazitäten ($V_m \cdot v_{mt}$) minimiert werden.

5.1.1.1.4 Restriktionen

Die Produktion eines Erzeugnisses erfordert den exklusiven Zugriff auf eine Maschine, deshalb konkurrieren Erzeugnisse um die zur Verfügung stehenden Maschinen. Ein Erzeugnis kann auch alternativ auf parallelen Maschinen produziert werden und die Produktionsgeschwindigkeit einzelner Maschinen unterliegt Schwankungen. Die

²¹⁸ Vgl. Kapitel 5.1.2.1

Produktion eines Erzeugnisses kann nur stattfinden, wenn sich die Maschine im richtigen Rüstzustand befindet.

Verschiedene Erzeugnisse müssen produziert werden, um eine dynamische Nachfrage zu befriedigen. Der Planungshorizont ist dabei in endlich viele Perioden aufgeteilt. Darüber hinaus werden positive Vorlaufzeiten berücksichtigt.

Sobald eine Rüstaktion durchgeführt wurde, wird der Rüstzustand beibehalten bis eine Rüstzustandsänderung auftritt. Auf diese Weise sind Leerlaufzeiten bei der Produktion eines Erzeugnisses auf einer bestimmten Maschine erlaubt. Es fallen keine zusätzlichen Rüstaktionen an. Eine Änderung des Rüstzustands kann nur zu Periodenbeginn stattfinden. Dabei kann jede Maschine nur eine Erzeugnisart in einer Periode herstellen.

Basierend auf diesen Überlegungen und den in Kapitel 5.1.1.1.2 definierten Parametern und Variablen werden die Restriktionen des deterministischen Modells aufgestellt:

$$s_{p(t-1)} + \sum_{i:P_i=p} q_{it} - D_{pt} - \sum_{\substack{q \in P \\ j:P_j=q}} a_{pq} \cdot q_{j(t+\Delta_{p,j})} + b_{pt} + o_{pt} - b_{p(t-1)} = s_{pt} \quad \forall p \in P, t = 1, \dots, T \quad (1.2)$$

$$b_{pt} + o_{pt} - b_{p(t-1)} \leq D_{pt} \quad \forall p \in P, t = 1, \dots, T \quad (1.3)$$

$$\sum_{i:M_i=m} y_{it} = 1 \quad \forall m \in M, t = 1, \dots, T \quad (1.4)$$

$$y_{i(t-1)} + y_{jt} - 1 \leq x_{ijt} \quad \forall i \in I, \forall j \in I_i, t = 1, \dots, T \quad (1.5)$$

$$C_{it} \cdot (y_{it} - \sum_{\substack{j \in I_i \\ t-R_{j,i} < t' \leq t}} x_{jit'} - \sum_{j \in I_i} R'_{ji} \cdot x_{ji(t-R_{j,i})}) \geq q_{it} \quad \forall i \in I, t = 1, \dots, T \quad (1.6)$$

$$b_{pt} \geq 0 \quad \forall p \in P, t = 1, \dots, T \quad (1.7)$$

$$o_{pt} \geq 0 \quad \forall p \in P, t = 1, \dots, T \quad (1.8)$$

$$q_{it} \geq 0 \quad \forall i \in I, t = 1, \dots, T \quad (1.9)$$

$$s_{pt} \geq 0 \quad \forall p \in P, t = 1, \dots, T \quad (1.10)$$

$$x_{ijt} \in \{0, 1\} \quad \forall i \in I, j \in I_i, t = 1, \dots, T \quad (1.11)$$

$$y_{it} \in \{0, 1\} \quad \forall i \in I, t = 1, \dots, T \quad (1.12)$$

Die Gleichungen (1.2) bilden die Lagerbilanzgleichungen. Der Lagerbestand von Erzeugnis p zum Ende der Periode t kann formalisiert werden als: Lagerbestand zum Ende der Periode $t-1$ plus die produzierte Menge während der Periode t minus dem internen und externen Bedarf. Darüber hinaus müssen die Erzeugnisse, die in Periode t zurückgestellt wurden und die Erzeugnisse deren Auftrag in Periode t nicht erfüllt und gelöscht wurde, addiert werden. Diese Erzeugnisse reduzieren den internen und externen Bedarf in der aktuellen Periode. Subtrahiert werden die Anzahl der zurückgestellten Erzeugnisse in Periode $t-1$. Diese Erzeugnisse müssen noch produziert werden, jetzt oder in einer zukünftigen Periode. Dabei ist zu beachten, dass für den internen Bedarf positive Vorlaufzeiten Δ_{pj} berücksichtigt werden, indem Erzeugnis p Δ_{pj} Perioden vor Beginn von Aufgabe j aus dem Lager genommen wird. Während dieser Δ_{pj} Perioden fallen für den internen Bedarf des Erzeugnisses p keine Lagerkosten an, aber diese Kosten können zu den Produktionskosten von Aufgabe j addiert werden.

Durch die Ungleichungen (1.3) wird sichergestellt, dass niemals mehr Produkte während einer Periode als zurückgestellt und damit als nicht erfüllt gekennzeichnet werden, als externer Bedarf besteht.

Mit den Gleichungen (1.4) und (1.10) wird sichergestellt, dass jede Maschine einen eindeutigen Rüstzustand am Ende einer Periode besitzt.

Wird in einer Periode gerüstet, so wird dies durch die Ungleichungen (1.5) gezeigt. Falls eine Maschine am Ende der Periode $t-1$ ($y_{i(t-1)} = 1$) in Zustand i war und der Zustand am Ende der Periode t gleich j ist ($y_{jt} = 1$) wurde in der Periode t von i auf j gerüstet und dieses erzwingt, dass in Ungleichung (1.5) x_{ijt} auf 1 gesetzt wird. Für alle

anderen Rüstzustände kann x_{ijt} Null werden und die Rüstkosten X_{ij} werden dieses für die optimale Lösung erzwingen.

Die Produktionskapazität wird durch die Ungleichungen (1.6) festgelegt. Dabei kann Aufgabe i in Periode t nur produzieren, wenn die Maschine M_i im richtigen Rüstzustand während dieser Periode ist ($y_{it} = 1$). Sind alle auftretenden x_{ijt} Null, können bis zu C_{it} Produkte in Periode t produziert werden. Da Rüstzeiten berücksichtigt werden, muss sichergestellt werden, dass die Umrüstung von einer Aufgabe j zu i vor oder während Periode t fertig gestellt wird. Die erste Summe reduziert die verfügbare Produktionskapazität auf Null für Perioden die vollständig für Umrüstungen reserviert sind. Die zweite Summe reduziert die Produktionskapazität um die für die Umrüstung erforderliche Zeit.

Nichtnegativität für die Produktionsmenge, den Lagerbestand, das Zurückstellungen und das Nichterfüllen eines Bedarfes wird durch die Restriktionen (1.7), (1.8), (1.9) und (1.10) gewährleistet.

Für die Rüstzustandsänderungen und Rüstzustandsvariablen werden in den Bedingungen (1.11) und (1.12) Binärvariablen definiert.

5.1.1.2 Entwicklung des stochastischen Modells

Das deterministische Modell wird zu einem stochastischen Modell erweitert, um die Anforderung an eine Lösung des Problems zu erfüllen. Der Betrachtungshorizont wird in Kapitel 5.1.1.2.1 klassifiziert und die Parameter und Variablen in Kapitel 5.1.1.2.2 konzipiert. In Kapitel 5.1.1.2.3 wird die Kostenfunktion entwickelt und in Kapitel 5.1.1.2.4 werden die Restriktionen des Modells aufgestellt.

5.1.1.2.1 Betrachtungshorizont

Der Betrachtungshorizont ist identisch mit der für das deterministische Modell entwickelten Klassifikation des Betrachtungshorizonts und wird an dieser Stelle nicht erneut entwickelt. Stattdessen wird auf Kapitel 5.1.1.1.1 verwiesen.

5.1.1.2.2 Parameter und Variablen

In der Produktion unterliegen verschiedene Aspekte des Produktionsablaufs stochastischen Einflüssen. Dabei ist eine Berücksichtigung von stochastischen Einflüssen mit einer Erhöhung der Komplexität verbunden. Allerdings werden somit das

Produktionssystem und der Produktionsablauf realistisch dargestellt. Aus Komplexitätstheoretischer Sicht liegt das Problem im Bereich PSPACE²¹⁹.

Die Auftragsmenge kann von einem Kunden nach Eingang der Bestellung erhöht oder verringert werden und neue Bestellungen können kurz-, mittel- und langfristig eingehen. Damit unterliegt die Auftragsmenge Unsicherheit.

Des Weiteren unterliegen die Produktionsgeschwindigkeit, also die Zeit, die eine bestimmte Maschine für die Durchführung einer Teilaufgabe benötigt, und damit die Produktionsmenge, stochastischen Schwankungen. Weiterhin kann eine Maschine ausfallen und so erhebliche Verzögerungen verursachen, was ebenfalls durch eine verringerte Produktionsmenge abgebildet wird.

In dieser Arbeit werden die Auftragsmenge und die Produktionsgeschwindigkeit als stochastische Größen erfasst. Beide Parameter werden im Optimierungsmodell durch die Funktion Stochastik $S()$ ergänzt. Auftragsmenge und Produktionsgeschwindigkeit können somit mehrere diskrete Werte mit einer bestimmten Wahrscheinlichkeit annehmen.

Durch folgenden Ausdruck wird in dem Modell die Auftragsmenge erfasst:

$S(D_{p,t})$	Externer stochastischer Bedarf nach Produkt p in Periode t
--------------	--

Die Produktionsmenge wird durch folgenden Ausdruck modelliert:

$S(q_{i,t})$	Stochastische Produktionsmenge von Aufgabe i in Periode t
--------------	---

Die Auftragsmengen und Produktionsgeschwindigkeiten werden in der Produktionsdatenbank²²⁰ gespeichert.

5.1.1.2.3 Kostenfunktion

Auf Grundlage der in Kapitel 5.1.1.1.1 vorgenommenen Klassifikation des Betrachtungshorizonts, der in Kapitel 5.1.1.1.2 definierten Parameter und Variablen

²¹⁹ In der Komplexitätstheorie bezeichnet PSPACE die Klasse der Entscheidungsprobleme, die von deterministischen Turingmaschinen mit polynomialem Platz entschieden werden können. Die Klassen NC, P und NP sind jeweils Untermengen der Klasse PSPACE [Wik08-ol].

²²⁰ Vgl. Kapitel 5.1.2.1

und der Erweiterungen in Kapitel 5.1.1.2.2, sowie der in Kapitel 5.1.1.2.3 formalisierten Kostenfunktion des deterministischen Modells, wird die Kostenfunktion des stochastischen Modells definiert:

$$\begin{aligned} \text{Min } Z_{\text{SPLSP-PM}} = & \sum_{t=1}^T \left(\sum_{p \in P} S_p \cdot s_{pt} + B_p \cdot b_{pt} + O_p \cdot o_{pt} \right) + \sum_{\substack{i \in I \\ j \in I_i}} X_{ij} \cdot x_{ijt} + \\ & \sum_{i \in I} Q_i \cdot S(q_{it}) + \sum_{m \in M} V_m \cdot v_{mt} \end{aligned} \quad (1.1)$$

Durch die Kostenfunktion (1.1) wird sichergestellt, dass über alle Perioden die Lagerhaltungskosten ($S_p \cdot s_{pt}$) für alle Produkte, plus die Kosten für den Totalausfall ($O_p \cdot o_{pt}$) eines Produktes, plus die Kosten für die verspätete Fertigstellung ($B_p \cdot b_{pt}$) eines Produktes, plus die Rüstkosten ($X_{ij} \cdot x_{ijt}$), plus die Produktionskosten ($Q_i \cdot S(q_{it})$) sowie die Kosten für ungenutzte Kapazitäten ($V_m \cdot v_{mt}$) minimiert werden.

5.1.1.2.4 Restriktionen

Basierend auf der in Kapitel 5.1.1.1.1 vorgenommenen Klassifikation des Betrachtungshorizonts, der in Kapitel 5.1.1.1.2 definierten Parameter und Variablen, sowie den in Kapitel 5.1.1.2.2 formalisierten Erweiterungen und den in Kapitel 5.1.1.1.4 entwickelten Restriktionen des deterministischen Modells, werden die Restriktionen des stochastischen Modells aufgestellt:

$$\begin{aligned} s_{p(t-1)} + \sum_{i: P_i=p} S(q_{it}) - S(D_{pt}) - \sum_{\substack{q \in P \\ j: P_j=q}} a_{pq} \cdot S(q_{j(t+\Delta_{p,j})}) \\ + b_{pt} + o_{pt} - b_{p(t-1)} = s_{pt} \end{aligned} \quad \begin{array}{l} \forall p \in P \\ t = 1, \dots, T \end{array} \quad (1.2)$$

$$b_{pt} + o_{pt} - b_{p(t-1)} \leq S(D_{pt}) \quad \begin{array}{l} \forall p \in P \\ t = 1, \dots, T \end{array} \quad (1.3)$$

$$\sum_{i: M_i=m} y_{it} = 1 \quad \begin{array}{l} \forall m \in M \\ t = 1, \dots, T \end{array} \quad (1.4)$$

$$y_{i(t-1)} + y_{jt} - 1 \leq x_{ijt} \quad \begin{array}{l} \forall i \in I \\ \forall j \in I_i \\ t = 1, \dots, T \end{array} \quad (1.5)$$

$$C_{it} \cdot \left(y_{it} - \sum_{\substack{j \in I_i \\ t-R_{j,i} < t' \leq t}} x_{j it'} - \sum_{j \in I_i} R'_{ji} \cdot x_{j i(t-R_{j,i})} \right) \geq S(q_{it}) \quad \begin{array}{l} \forall i \in I \\ t = 1, \dots, T \end{array} \quad (1.6)$$

$$b_{pt} \geq 0 \quad \forall p \in P, t = 1, \dots, T \quad (1.7)$$

$$o_{pt} \geq 0 \quad \forall p \in P, t = 1, \dots, T \quad (1.8)$$

$$S(q_{it}) \geq 0 \quad \forall i \in I, t = 1, \dots, T \quad (1.9)$$

$$s_{pt} \geq 0 \quad \forall p \in P, t = 1, \dots, T \quad (1.10)$$

$$x_{ijt} \in \{0, 1\} \quad \forall i \in I, j \in I_i, t = 1, \dots, T \quad (1.11)$$

$$y_{it} \in \{0, 1\} \quad \forall i \in I, t = 1, \dots, T \quad (1.12)$$

Mit den Lagerbilanzgleichungen (1.2) werden die Bestände am Ende einer Periode mit den Produktionsmengen und den Bedarfen dieser Periode sowie den Beständen der vorangegangenen Periode verknüpft. Darüber hinaus wird nicht erfüllter Bedarf aus der Vorperiode zu dem Bedarf in dieser Periode hinzugefügt. Bedarf, für den Strafkosten für Nicht-Erfüllung in Kauf genommen werden, wird ebenso abgezogen wie Bedarf, der unter Beachtung der Verzugskosten erst in der nächsten Periode erfüllt wird.

Die Ungleichungen (1.3) sichern, dass niemals mehr Produkte während einer Periode als zurückgestellt und nicht erfüllt gekennzeichnet werden, als externer Bedarf besteht.

Mit den Ungleichungen (1.4) wird jeder Maschine ein eindeutiger Rüstzustand zugewiesen. Wird in einer Periode gerüstet, so wird dies durch die Ungleichungen (1.5) aufgezeigt.

Durch die Ungleichungen (1.6) werden die Produktionskapazitäten individuell festgelegt.

Nichtnegativität wird für die Variablen durch die Restriktionen (1.7), (1.8), (1.9) und (1.10) gewährleistet und die Bedingungen (1.11) und (1.12) definieren die Binärvariablen.

5.1.2 Ein Beitrag für die Simulation des Produktionsablaufs

Mit dem Ziel, das Planungsproblem, die Simulation des Produktionsablaufs und die Planungsverfahren voneinander zu entkoppeln, wird eine Benchmarkplattform konzipiert, die die Schnittstelle zwischen verschiedenen Planungsverfahren und Planungsproblemen bildet und gleichzeitig die Simulation des Produktionsablaufs durchführt. Die Benchmarkplattform besteht aus drei Komponenten, wobei die spezielle Produktionsdatenbank in Kapitel 5.1.2.1 entwickelt wird, der Simulator in Kapitel 5.1.2.2 konzipiert wird und eine Schnittstelle für Planungsverfahren in Kapitel 5.1.2.3 entworfen wird.

5.1.2.1 Entwicklung der Produktionsdatenbank

Alle für die Aufstellung des Produktionssystems relevanten Datensätze werden in der Produktionsdatenbank zentral gespeichert. Die Determinanten der Produktionsdatenbank werden in Kapitel 5.1.2.1.1 formalisiert und das Funktionsprinzip in Kapitel 5.1.2.1.2 entwickelt.

5.1.2.1.1 Determinanten

In Erweiterung zu bestehenden Ansätzen werden Maschinenausfälle, geänderte oder neue Kundenbestellungen und Produktionsschwankungen sowie Opportunitätskosten für ungenutzte Maschinenkapazitäten abgebildet. Weiterhin werden ausschließlich die für das betrachtete Planungsproblem relevanten Zusammenhänge fokussiert. Der Zugriff auf die Datenbank erfolgt dabei exklusiv durch den Simulator (vgl. Abbildung 4).

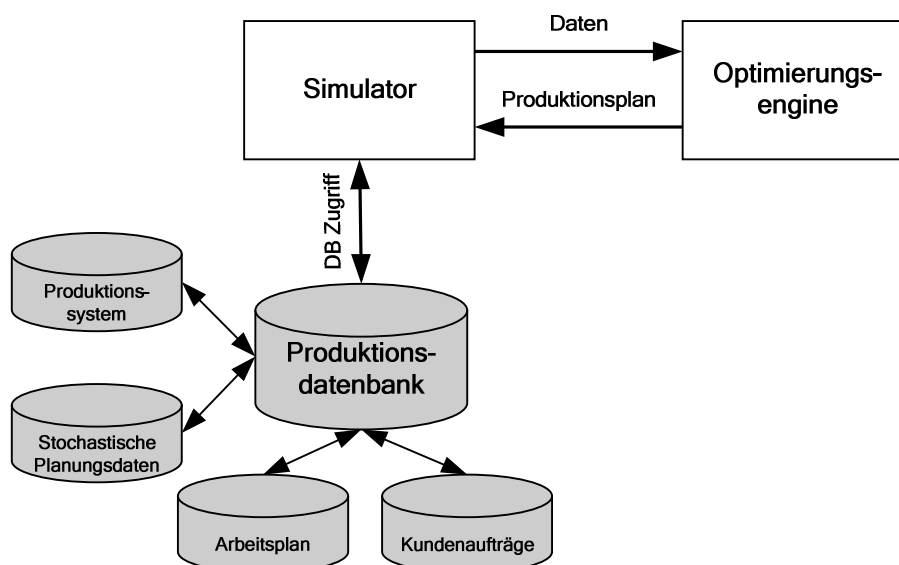


Abbildung 4: Komponente Produktionsdatenbank

Für die einfache Verwaltung werden die benötigten Daten in einer relationalen Datenbank erfasst, die sich in MySQL²²¹ erstellen lässt. Innerhalb der relationalen Datenbank werden die einzelnen Daten in Tabellen gespeichert. Primärschlüssel²²² (PK) ermöglichen innerhalb der Tabellen die eindeutige Identifizierung der Datensätze und Fremdschlüssel²²³ (FK) stellen die Verbindung zwischen den Tabellen her.

5.1.2.1.2 Funktionsprinzip

Die Produktionsdatenbank wird in vier Komponenten entwickelt: Kundenaufträge²²⁴, Arbeitsplan²²⁵, Produktionssystem²²⁶ und Stochastische Planungsdaten²²⁷.

5.1.2.1.2.1 Kundenaufträge

Ausgangspunkt für jede Produktionsplanung ist ein bestehender Primärbedarf, der durch einen Kundenauftrag hervorgerufen wird. Ein Auftrag bietet dem Kunden die Möglichkeit, das zu fertigende Produkt, den Fertigstellungs-/Auslieferungstermin und die zu produzierende Menge zu bestimmen. Jedem Kundenauftrag wird dabei eine eindeutige Auftragsnummer zugewiesen.

Diese Informationen werden in der Tabelle „Auftragslage“ gespeichert und über den Primärschlüssel „auftrag_id“ eindeutig identifiziert. Das Attribut „zufallsverteilung_id“ ordnet dem Auftrag durch eine diskrete Wahrscheinlichkeitsverteilung²²⁸ eine Produktionsmenge zu. Die Spalte „produkt_id“ bezeichnet innerhalb der Tabelle

²²¹ MySQL ist kostenlos verfügbar, läuft stabil und lässt sich gut in eine webbasierte Oberfläche einbinden. Letzteres wiederum hat den Vorteil, dass die Eingabe und Bearbeitung der Daten, unabhängig vom Betriebssystem, auf einer Vielzahl von verschiedenen Eingabegeräten möglich ist.

²²² Ein PK kann dabei auch aus mehreren Attributen bestehen. Besteht der PK aus mehreren Attributen werden diese iterativ nummeriert.

²²³ FK sind Attribute, welche auf Primärschlüssel anderer Tabellen verweisen und somit eine Verbindung zwischen den Relationen herstellen. Ein FK kann, muss aber nicht, PK seiner Relation sein.

²²⁴ Vgl. Kapitel 5.1.2.1.2.1

²²⁵ Vgl. Kapitel 5.1.2.1.2.2

²²⁶ Vgl. Kapitel 5.1.2.1.2.3

²²⁷ Vgl. Kapitel 5.1.2.1.2.4

²²⁸ Vgl. Kapitel 5.1.2.1.2.4

„Auftragslage“ das zu fertigende Produkt. Die Attribute „zufallsverteilung_id“ und „produkt_id“ sind Fremdschlüssel der Tabelle Auftragslage (vgl. Abbildung 5). Das Attribut „endtermin“ spezifiziert den Liefertermin des Auftrags.

Kundenaufträge	
Auftragslage	
PK	<u>auftrag_id</u>
FK1	produkt_id
FK2	zufallsverteilung_id endtermin

Abbildung 5: Datenbankkomponente Kundenaufträge

5.1.2.1.2.2 Arbeitsplan

Im Rahmen der Fertigung ist zu berücksichtigen, dass ein Produkt²²⁹ das Ergebnis aus verschiedenen Teilprodukten sein kann. Dabei ist im Rahmen der Fertigung zu beachten, dass ein Folgeprodukt aus verschiedenen Vorprodukten zusammensetzt werden kann. Ein Folgeprodukt kann innerhalb einer nachgelagerten Produktionsstufe im Fertigungsverlauf selbst wieder ein Vorprodukt sein. Sämtliche Bauteile sowie das Fertigerzeugnis selbst, werden als Produkte in der gleichnamigen Tabelle „produkt“ gespeichert.²³⁰

Neben dem Produktnamen, der maximalen Lagerkapazität, dem Lagerbestand und den Lagerkosten des entsprechenden Produktes in Euro wird jedem Produkt auch die Produktionsmenge zugeordnet, die auf einmal gefertigt werden muss. Die entsprechenden Attribute in der Tabelle sind „name“, „lagerkapazität“, „lagerbestand“, „wert“ und „produktionsmenge“. Ein weiteres wichtiges Attribut dieser Tabelle ist „teilaufgabe_id“. Hierbei handelt es sich um einen Fremdschlüssel der Tabelle, welcher einen Verweis auf die zweite der drei Tabellen in diesem Bereich herstellt. Die Tabelle „Produkt“ verfügt über zwei Attribute zur Definition von Konventionalstrafen. Dabei gibt „penalty_stockout“ die Kosten an, die bei nicht termingerechter Auslieferung entstehen und „penalty_backlog“ die entstehenden Kosten pro Minute bei der verspäteten

²²⁹ Steht an dieser Stelle stellvertretend für Einzelteil, Baugruppe und Erzeugnis.

²³⁰ Die Tabelle „Produkt“ beinhaltet deshalb sämtliche Vorprodukte der herzustellenden Endprodukte sowie die Endprodukte selbst.

Auslieferung eines Auftrages. Der Primärschlüssel der Tabelle „Produkt“ ist „produkt_id“ (vgl. Abbildung 6).

Produkt	
PK	<u>produkt_id</u>
FK1	teilaufgabe_id name lagerkapazität wert penalty_stockout penalty_backlog lagerbestand produktionsmenge

Abbildung 6: Datenbanktabelle Produkt

Der Arbeitsplan legt die einzelnen Arbeitsvorgänge²³¹, die zur Herstellung einer Baugruppe oder des Fertigerzeugnisses notwendig sind, fest. Die einzelnen Teilaufgaben bzw. Vorgänge werden in der Tabelle „teilaufgabe“ mit Namen abgelegt, die alle Werte des Attributes „teilaufgabe_id“ verwaltet und ihnen einen bezeichnenden Namen zuweist. Das Attribut „teilaufgabe_id“ definiert, mit welcher Teilaufgabe, also mit welchem Produktionsschritt, das entsprechende Produkt hergestellt wird (vgl. Abbildung 7).

Teilaufgabe	
PK	<u>teilaufgabe_id</u>
	name

Abbildung 7: Datenbanktabelle Teilaufgabe

Zu den Teilaufgaben jeweils notwendige Ressourcen in Form von Bauteilen oder Zwischenerzeugnissen werden mit Hilfe der Tabelle „stueckliste“ festgelegt. In der Tabelle „Stueckliste“ wird der Zusammenhang zwischen den einzelnen Bauteilen

²³¹ In der Datenbank als Teilaufgabe bezeichnet.

definiert. Sämtliche Teile, die zur Herstellung als Vor- oder Folgeprodukte in den Produktionsprozess eingehen, werden als Datensatz gespeichert. Infolge der Festlegung der Fertigungsstruktur wird auch die entsprechende Einsatzmenge zur Herstellung einer Einheit des entsprechenden Produktes in der Stückliste hinterlegt. Dazu verfügt die Tabelle „stueckliste“ über die Attribute „produkt_id“ und „teilaufgabe_id“, die sowohl Primär- als auch Fremdschlüssel sind, und über die Spalte „menge“ (vgl. Abbildung 8).

Stueckliste	
PK,FK2	<u>produkt_id</u>
PK,FK1	<u>teilaufgabe_id</u>
	menge

Abbildung 8: Datenbanktabelle Stueckliste

Der Arbeitsplan wird damit durch drei Tabellen abgebildet und legt die Herstellungsschritte der zu fertigenden Produkte fest. Die drei Tabellen Produkt, Teilaufgabe und Stueckliste werden in dem Bereich Arbeitsplan zusammengefasst (vgl. Abbildung 9).

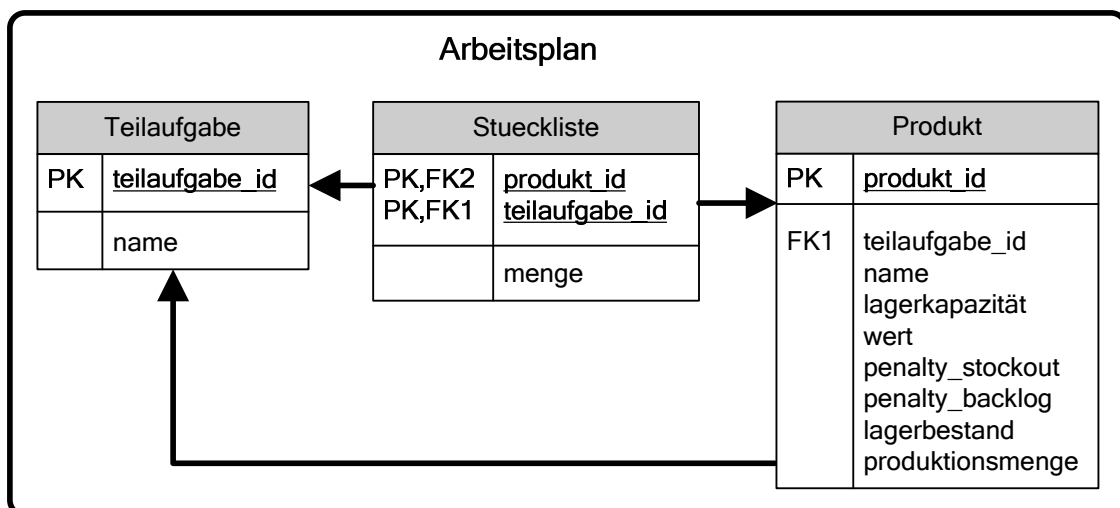


Abbildung 9: Datenbankkomponente Arbeitsplan

5.1.2.1.2.3 Produktionssystem

Im Datenbankmodell umfasst die Tabelle „maschine“ sämtliche Betriebsmittel²³², die die Produktion durchführen, wobei das betrachtete Produktionssystem aus mehreren durchaus auch gleichartigen Maschinen besteht. Die Tabelle „maschine“ speichert die zur Produktion bereitstehenden Maschinentypen und deren verfügbare Anzahl. Diese Informationen werden in den Spalten „maschine_id“, „name“ und „anzahl“ abgelegt. Neben dem vorhandenen Maschinenbestand werden auch die Lieferanten von Bauteilen als Potentialfaktoren betrachtet und ebenfalls in der Tabelle „maschine“ abgelegt. Das Attribut „maschine_id“ wird in dieser Tabelle als Primärschlüssel verwendet und jeder Maschinentyp wird eindeutig identifiziert (vgl. Abbildung 10).

Maschine	
PK	<u>maschine_id</u>
	name anzahl opportunitätskosten

Abbildung 10: Datenbanktabelle Maschine

Einer Teilaufgabe werden mehrere Maschinen und umgekehrt einer Maschine mehrere Teilaufgaben durch die Tabelle „Teilaufgabe_Maschine“ zugeordnet. Fertigungszeiten²³³ sind den Teilaufgaben in der Tabelle „teilaufgabe_maschine“ zugeordnet. Das Attribut „teilaufgabe_maschine_id“ stellt den Primärschlüssel dar. Auf diese Weise kann, über die Fremdschlüssel „zufallsverteilung_id“, „maschine_id“ und „teilaufgabe_id“, einer oder mehreren Maschinen eine oder mehrere Teilaufgaben zugewiesen werden. Das Attribut „zufallsverteilung_id“ ist Fremdschlüssel der Tabelle.

Kosten, die durch die Durchführung einer Teilaufgabe auf einem Betriebsmittel anfallen, werden in dem Attribut „kosten_pro_periode“ abgelegt. Zusätzlich besteht die Möglichkeit, über die Spalte „dauer_min“, die zugehörige Dauer festzulegen (vgl. Abbildung 11).

²³² Betriebsmittel umfasst hier sämtliche Anlagen, Einrichtungen, Maschinen, Werkzeuge, Vorrichtungen oder sonstige gegenständliche Produktionsmittel, die im Arbeitssystem an der Ausführung der Arbeitsaufgabe beteiligt sind.

²³³ Bezeichnet die einzelnen Vorgangszeiten.

Teilaufgabe_Maschine	
PK	<u>teilaufgabe_maschine_id</u>
FK3	zufallsverteilung_id
FK2	maschine_id
FK1	teilaufgabe_id
	kosten_pro_periode
	dauer_min

Abbildung 11: Datenbanktabelle Teilaufgabe_Maschine

In der Tabelle „ruestkosten“ sind für jeden vorkommenden Rüstvorgang die dafür anfallenden Kosten und der notwendige Zeitaufwand²³⁴ enthalten. Die Tabelle fokussiert auf die Abbildung des Rüstvorgangs von einer Teilaufgabe zu einer folgenden Teilaufgabe. Die eindeutige Zuordnung erfolgt durch den Primärschlüssel „ruestkosten_id“. Fremdschlüssel ist die Spalte „maschine_id“.

Dabei beschreiben die Attribute „teilaufgabe_von_id“ und „teilaufgabe_nach_id“ die entsprechende Teilaufgabe, von der bzw. zu der gewechselt wird. Die Spalten „zeitaufwand“ und „kosten_pro_periode“ legen die Zeitdauer der Umrüstung und die damit verbundenen Kosten fest (vgl. Abbildung 12).

Ruestkosten	
PK	<u>ruestkosten_id</u>
FK1	maschine_id
	teilaufgabe_von_id
	teilaufgabe_nach_id
	zeitaufwand
	kosten_pro_periode

Abbildung 12: Datenbanktabelle Ruestkosten

Wie in der Einleitung abgegrenzt, ist es möglich, Kapazität für die übernächste Woche zu verringern. In der Tabelle Nutzungsgrad wird für jede Maschine die nicht produziert

²³⁴ Bedingt durch das Umstellen der Maschinen.

ein Eintrag eingelegt.²³⁵ In der Spalte „faktor“ wird der Grad angegeben, um den die Opportunitätskosten und die Produktionsmenge gesenkt werden. Dabei kann dieser Faktor zwischen 100 % und 0 % liegen. In Abbildung 13 wird die Tabelle illustriert.

Nutzungsgrad	
PK	<u>maschine_id</u>
PK	<u>periode_von</u>
	periode_bis faktor

Abbildung 13: Datenbanktabelle Nutzungsgrad

In dem Bereich Produktionssystem werden die vier Tabellen Maschine, Teilaufgabe_Maschine, Ruestkosten und Nutzungsgrad gebündelt und die Abhängigkeiten untereinander dargestellt (vgl. Abbildung 14).

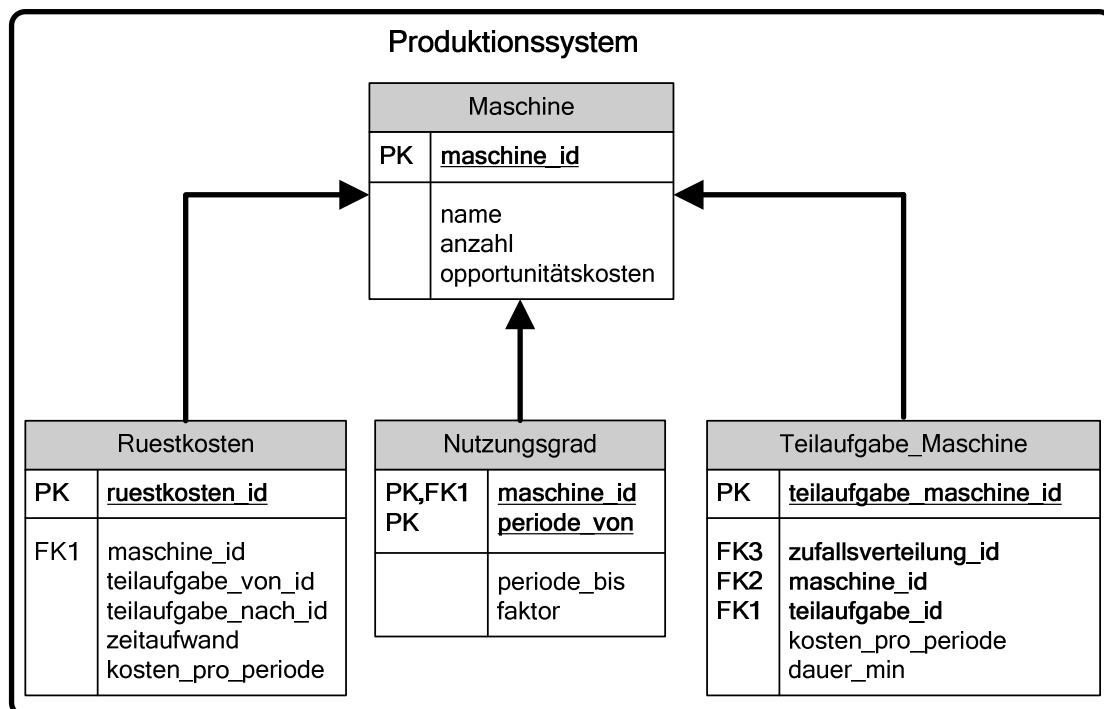


Abbildung 14: Datenbankkomponente Produktionssystem

²³⁵ Für in der Tabelle rechtzeitig für die übernächste Woche gespeicherte Maschinen, fallen in dem festgelegten Zeitraum geringere Opportunitätskosten an.

5.1.2.1.2.4 Stochastische Planungsdaten

Mit dem Ziel Störungen abzubilden, werden die Schwankungen bei Auftragsgrößen, Maschinengeschwindigkeiten oder auch Lieferzeiten der Teilezulieferer in den Tabellen „zufallsverteilung“ und „verteilungstabelle“ gespeichert. In der Tabelle „zufallsverteilung“ werden dazu die Einflussfaktoren benannt, während in der „verteilungstabelle“ die jeweils möglichen Parameter mit einem Wahrscheinlichkeitswert gewichtet werden.

Dazu enthält die Tabelle „zufallsverteilung“ die Spalten „zufallsverteilung_id“, „name“ und „diskret“, um den Verteilungen einen identifizierenden eindeutigen Namen zuzuweisen. Des Weiteren wird spezifiziert, ob es sich um eine diskrete Verteilung handelt (vgl. Abbildung 15).

Zufallsverteilung	
PK	<u>zufallsverteilung_id</u>
	name diskret

Abbildung 15: Datenbanktabelle Zufallsverteilung

Die Tabelle „Verteilungstabelle“ ordnet den festgelegten Verteilungen in einem weiteren Schritt die Wahrscheinlichkeiten mit den entsprechenden Werten zu. Auf diese Weise können zum Beispiel der Verteilung „nachfrage“ verschiedene Mengenwerte zugeteilt werden. Verteilungen können auf diese Weise so detailliert wie gefordert abgebildet werden und an die Bedürfnisse unternehmensspezifisch variabel adaptiert werden (vgl. Abbildung 16).

Verteilungstabelle	
PK	<u>verteilungstabelle_id</u>
FK1	<u>zufallsverteilung_id</u> wert wahrscheinlichkeit

Abbildung 16: Datenbanktabelle Verteilungstabelle

Verwendet werden die Verteilungen innerhalb der Tabellen „Auftragslage“ und „Teilaufgabe_Maschine“, um die Produktionsmengen bzw. Produktionsgeschwindigkeiten und die Auftragsmengen abzubilden.²³⁶ Der Zusammenhang der Tabellen Verteilungstabelle und Zufallsverteilung wird in Abbildung 17 verdeutlicht.

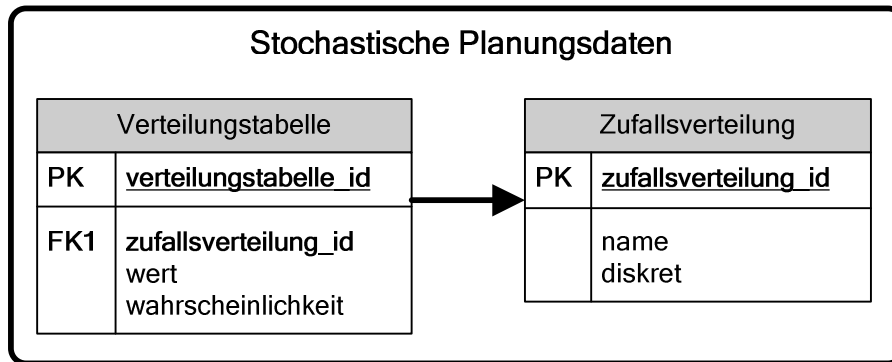


Abbildung 17: Datenbankkomponente Stochastische Planungsdaten

5.1.2.2 Entwicklung des Simulators

Die Determinanten des Simulators werden in Kapitel 5.1.2.2.1 aufgestellt und auf dieser Grundlage das Funktionsprinzip des Simulators in Kapitel 5.1.2.2.2 entwickelt.

5.1.2.2.1 Determinanten

Im Simulator wird die Produktion in dem fokussierten Produktionssystem auf Grundlage eines Produktionsplans über den gesamten Betrachtungshorizont abgebildet. Dabei werden verschiedene Restriktionen eingehalten. Kunden geben Bestellungen auf und ändern bzw. stornieren sie. Die Produktion wird für das in Kapitel 5.1.1 formalisierte Planungsproblem simuliert und die dort aufgestellten Restriktionen²³⁷ müssen bei der Produktion eingehalten werden.

Gewinn und Kosten des Produktionsablaufs werden basierend auf der entwickelten Zielfunktion²³⁸ berechnet. Dabei können auch mehrere Simulationsläufe parallel durchgeführt werden.

²³⁶ Die Kapazität einer Maschine ergibt sich aus der Produktionsgeschwindigkeit für eine Teilaufgabe.

²³⁷ Vgl. Kapitel 5.1.1.2.4

²³⁸ Vgl. Kapitel 5.1.1.2.3

Auf Grundlage eines von einem externen Planungsverfahren erstellten Produktionsplans wird die Produktion durchgeführt und die Zeit schreitet in der Simulation fort. Die Simulation wird dazu verwendet, Verfahren zu vergleichen, aber kann auch dazu verwendet werden, den Produktionsprozess zu verbessern, indem Anordnung und Anzahl von Maschinen optimiert werden.

Der Simulator bildet dabei die zentrale Einheit zwischen Produktionsdatenbank²³⁹ und der Schnittstelle für Planungsverfahren²⁴⁰. Des Weiteren verwaltet der Simulator die Datenbank, steuert den Austausch von Informationen mit der Optimierungseingine und simuliert den Produktionsablauf (vgl. Abbildung 18).

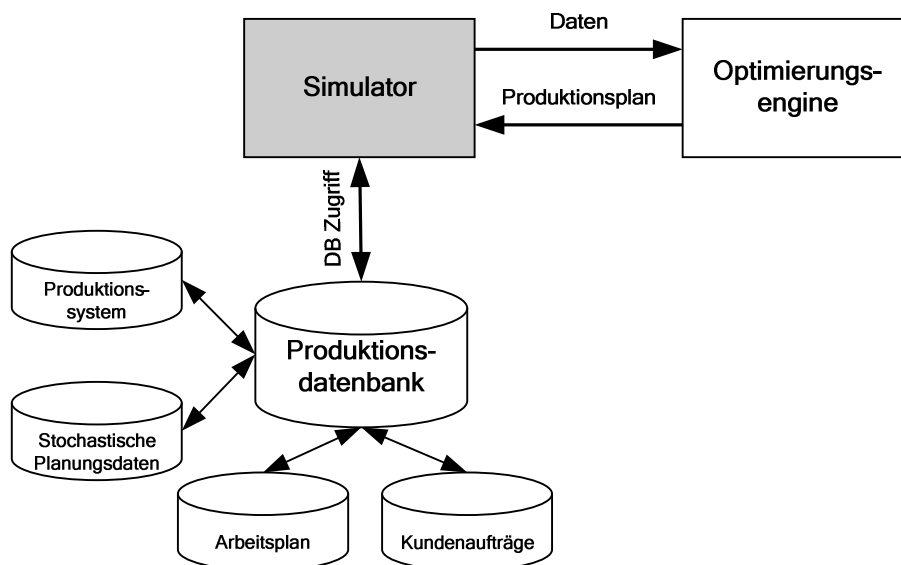


Abbildung 18: Komponente Simulator

Algorithmen werden in dieser Arbeit in Pseudocode²⁴¹ formalisiert. Die hier verwendete Notation basiert auf Standardprogrammkonstrukten, die fast allen Programmiersprachen gemein sind. Ziel der Notation ist es, eine schnelle Übersetzung der Algorithmen in verschiedene Programmiersprachen zu ermöglichen. Die Algorithmen starten mit **begin** und schließen mit **end**. Die Inhalte sind eingerückt und eine Reihung von Befehlen erfolgt über zeilenweise Befehlsangaben. Dabei wird jede Zeile iterativ nummeriert und die erste Zeile beginnt mit „S₁“ (Step 1).

²³⁹ Vgl. Kapitel 5.1.2.1

²⁴⁰ Auch als Optimierungseingine bezeichnet. Vgl. Kapitel 5.1.2.3

²⁴¹ Diese halbformale nicht standardisierte Programmiersprache ermöglicht es, einen Algorithmus durch Text und Schlüsselwörter zu beschreiben.

Schleifen werden durch **for** begonnen und Verzweigungen in der Form **if** ($i > n$) **then** [...] **else then** [...] angegeben. Die Anweisungen innerhalb des mit **begin** startenden und mit **end** aufhörenden if-Blockes werden dabei ausgeführt falls die Bedingung i größer n erfüllt ist. Andernfalls springt das Programm in den **else** Block, wobei der Anweisungsteil mit wiederum mit **begin** eingeleitet wird und mit **end** schließt. Kommentare werden durch **/*** eingeleitet und durch ***/** abgeschlossen. Die Anweisung **do parallel** zeigt an, dass der folgende Teil parallel ausgeführt wird. Der Operator **AND** verbindet zwei Ausdrücke miteinander. **Program** kennzeichnet den Beginn eines Programms. Programme können von anderen Programmen aus gestartet werden. Die Zuweisung einer Variable erfolgt über den Ausdruck „Variable := Wert“. **GoTo** bewirkt den Start an einer beliebigen Stelle des Programms, wobei von dort aus die Befehle weiterhin sukzessiv ausgeführt werden. Dabei wird z.B. die Programmzeile 1 durch den Befehl **GoTo** S_1 gestartet.

5.1.2.2 Funktionsprinzip

Im Simulator wird das stochastische Planungsproblem als ein rundenbasiertes Spiel, an dem zwei Spieler teilnehmen und jeweils abwechselnd ziehen, simuliert. Beide Spieler wissen zu jedem Zeitpunkt was beide Spieler bereits getan haben. Der eine Spieler ist ein Unternehmen, das versucht seinen Gewinn zu maximieren und der andere ist die Natur, die Kundenbedarfe und produzierte Produktionsmengen realisiert. In Abbildung 19 werden die Spieler zusammengefasst.

<u>Optimierer</u>	Der Optimierer versucht, einen langfristig möglichst optimalen Produktionsplan zu finden.
<u>Natur</u>	Die Natur determiniert laufend Auftragseingänge und Produktionsmengen.

Abbildung 19: Determinanten des Simulators

Ein Simulationslauf besteht aus einer Sequenz von Zügen der Spieler Natur und Optimierer, die jeweils abwechselnd ziehen und wird von der Optimierungseingine gestartet. Der Simulator schickt einmalig, um die Menge der zu übertragenden Daten zu verringern, alle Stammdaten an die Optimierungseingine. Anschließend werden für jeden folgenden Zeitabschnitt nur die Änderungen bzw. neuen relevanten Planungsdaten übertragen.

Wurde ein Produktionsplan an den Simulator über die Optimierungseingabe gesendet, startet die Produktion. Der Simulator führt dazu basierend auf diesem Produktionsplan die Produktion durch und setzt die simulierte Zeit einen Schritt vor. Anschließend bestimmt der Simulator die tatsächlichen Produktionsgeschwindigkeiten und Kundenbedarfe sukzessiv für jeden Zeitschritt T_i , der zwischen T_0 und T_e liegt und überträgt diese an die Optimierungseingabe. Auf diese Weise werden aus den bisher stochastisch vorliegenden Daten deterministische Werte.

Das Planungsverfahren kann jetzt ggf. auf die neue Situation reagieren, wobei die Möglichkeit besteht, die neuen Ereignisse zu ignorieren oder aber zu reagieren, indem der Produktionsplan ggf. durch eine Änderungsplanung anpasst wird. Die Zeit wird fortgeschrieben und das Programm springt zurück zu dem Punkt, an dem die durch die Natur bestimmten Änderungen übertragen werden.

Nach jedem Schritt werden vom Simulator Gewinn und Kosten des Produktionsablaufs berechnet. Der formulierte Ablauf wiederholt sich, bis das Ende des Simulationshorizonts erreicht wurde.

Im Folgenden wird in der entwickelten Pseudocode-Schreibweise der Algorithmus des Programms für den Simulator konzipiert (vgl. Abbildung 20).

Program Simulator()

S₁ **begin**

/ Ein Simulationslauf wird von einem Client unter Angabe der Start- und Endzeit des Intervalls, also der Anzahl von Minuten nach denen der Simulator eine Rückmeldung an den Optimierer liefert, des Ports auf dem eine Verbindung aufgebaut wird sowie des verwendeten Testdatensatzes, gestartet. */*

S₂ **StarteSimulation**(Benutzer, Passwort, Startzeit, Endzeit, Port, Intervall, Datensatz);

/ Sende die Stammdaten und die stochastischen Daten an den Optimierer. */*

S₃ **Sende**(„Plandaten“);

/ Für jeden Zeitabschnitt. */*

S₄ **for all** ($t \in T_i$)

S₅ **begin**

/ Der Simulator ist bereit, einen Aufruf zu empfangen und wartet auf den Optimierer. */*

S₆ **WarteAufOptimierer**();

/ Empfange den Produktionsplan. */*

S₇ **Empfange**(„Produktionsplan“);

/ Der Spieler Natur macht seinen Zug und ermittelt den neuen und geänderten Bedarf der Kunden. */*

S₈ **ZugDerNatur**(„Kundenbedarf“);

/ Der Spieler Natur macht seinen Zug und ermittelt die Produktionsgeschwindigkeit der Maschinen. */*

S₉ **ZugDerNatur**(„Produktionsgeschwindigkeit“);

/ Sende den von der Natur ermittelten Kundenbedarf an den Optimierer. */*

S₁₀ **Sende**(„Kundenbedarf“);

/ Sende die von der Natur ermittelte Produktionsgeschwindigkeit an den*

```

Optimierer. */
S11 Sende(„Produktionsmenge“);
/* Ermittle die bis heute im Produktionsablauf entstandenen Ist-Kosten. */
S12 ErmittleIstKosten();
S13 t := t + 1;
S14 end
S15 end.

```

Abbildung 20: Algorithmus des Simulators

5.1.2.3 Entwicklung der Schnittstelle für Planungsverfahren

Die Determinanten der Schnittstelle für Planungsverfahren werden in Kapitel 5.1.2.3.1 definiert und das Funktionsprinzip in Kapitel 5.1.2.3.2 entwickelt.

5.1.2.3.1 Determinanten

Die Schnittstelle für Planungsverfahren wird wie vorher bereits erwähnt auch als Optimierungseingine bezeichnet und ist die Verbindungseinheit zwischen dem Simulator²⁴² und mehreren Planungsverfahren (vgl. Abbildung 21).

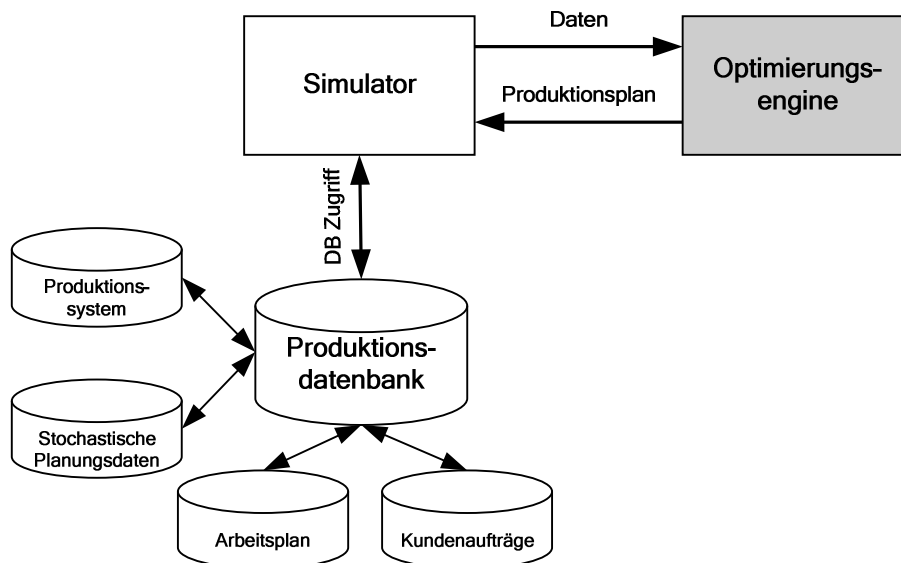


Abbildung 21: Komponente Optimierungseingine

Durch die Optimierungseingine die Kommunikation zwischen dem Simulator und verschiedenen Planungsverfahren gesteuert und weiterhin eine Oberfläche für den

²⁴² Vgl. Kapitel 5.1.2.2

Austausch der Planungsdaten geboten. Darüber hinaus ermöglicht die Optimierungseingine die Visualisierung des Produktionsablaufs.

5.1.2.3.2 Funktionsprinzip

Ein Optimierungsverfahren erstellt aus den Stammdaten und Bewegungsdaten einen Produktionsplan und sendet diesen an die Optimierungseingine, die den Produktionsplan an den Simulator weiterleitet. Abbildung 22 illustriert die Funktionsweise.

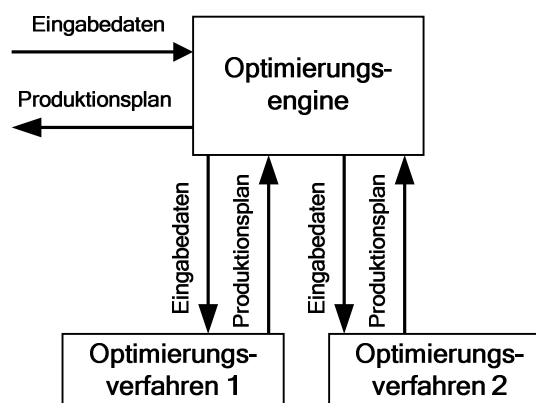


Abbildung 22: Funktionsprinzip der Optimierungseingine

Durch die Optimierungseingine ist es möglich, sich von jedem Rechner mit Zugang zum Internet durch Angabe eines Ports sowie von Benutzernamen und Passwort an einem Simulator anzumelden. Über das Attribut Datensatz kann eine Instanz gewählt werden, für die ein Produktionsplan erstellt und die Produktion simuliert werden soll. Startzeit und Endzeit definieren, wie lange die Simulation durchgeführt wird.

Ein Planungsverfahren, das einen Produktionsplan erstellt, wird mit dem Funktionsaufruf „Promise“ gestartet. Die Produktion wird nach der Übermittlung an den Simulator²⁴³ entsprechend des Produktionsplans durchgeführt und eine Rückmeldung mit den aktualisierten Planungsdaten vom Simulator empfangen, auf die das Planungsverfahren entsprechend reagieren kann.

In Abbildung 23 wird der Algorithmus in der in Kapitel 5.1.2.2.1 definierten Pseudocode Notation konzipiert.

²⁴³ Die Übermittlung erfolgt über die Optimierungseingine.

```
Program StarteSimulation()  
S1 begin  
    /* Baue eine Verbindung mit dem Simulator auf und starte eine Simulation */  
S2 Connect(Benutzer, Passwort, Startzeit, Endzeit, Port, Intervall, Datensatz);  
    /* Empfange die Stammdaten vom Simulator. */  
S3 Empfange(„Stammdaten“);  
    /* Empfange die stochastischen Daten vom Simulator. */  
S4 Empfange(„StochastischeDaten“);  
    /* Für jeden Zeitabschnitt. */  
S5 for all Zeitabschnitte do  
S6 begin  
    /* Empfange den neuen und geänderten Kundenbedarf vom Simulator. */  
S7 Empfange(„Kundenbedarf“);  
    /* Empfange die Produktionsmenge vom Simulator. */  
S8 Empfange(„Produktionsmenge“);  
    /* Starte das Planungsverfahren „Promise“ und sende die Planungsdaten. */  
S9 Produktionsplan := Promise();  
    /* Sende den erstellten Produktionsplan zur Umsetzung an den Simulator. */  
S10 Sende(Produktionsplan);  
S11 end  
S12 end.
```

Abbildung 23: Algorithmus der Optimierungseingine

5.2 Ein Beitrag zur Produktionsplanung und -steuerung für die mehrstufige auftragsbezogene Serienfertigung in der Automobilzulieferindustrie

Die im Stand der Technik analysierten Planungsstrategien werden in Kapitel 5.2.1 im Hinblick auf die Anforderungen an eine Problemlösung weiterentwickelt. Durch eine Rekombination bestehender Ansätze und stellenweise Neuentwicklungen wird in Kapitel 5.2.2 ein Beitrag zur PPS und in Kapitel 5.2.3 ein Beitrag zur PPS unter Unsicherheit geleistet. Die entwickelten Konzepte werden dabei in der definierten Pseudocode Notation²⁴⁴ formalisiert.

5.2.1 Ein Beitrag zu Planungsstrategien

Ziel der Planungsstrategien ist es, einen Beitrag dazu zu leisten, das betrachtete Produktionssystem bestmöglich aufzustellen. Dazu wurden in Kapitel 5.2.1.1 bis Kapitel 5.2.1.14 speziell für das betrachtete Problem den Anforderungen an eine Lösung entsprechende Planungsstrategien entwickelt.

5.2.1.1 Strategie 1: Erstelle Lösungspopulationen

Anstatt wie gängige Verfahren der Losgrößen- und Termin-/Kapazitätsplanung immer nur eine einzige Lösung zu ermitteln, sollten besser Lösungspopulationen erstellt und miteinander verglichen werden. Hierzu müssen verschiedene Produktionspläne aufgestellt und anschließend analysiert werden. Dabei können die verschiedenen Produktionspläne z.B. durch feste Regeln, Mutation oder Rekombination berechnet werden. Der beste Produktionsplan sollte dabei für den Produktionsablauf ausgewählt und die anderen gemäß des Prinzips „Survival of the fittest“ verworfen werden.

Vorteil dieser Strategie ist, dass ein Multi-Solution Ansatz verfolgt wird und damit die Analyse mehrerer vollständiger Produktionspläne ermöglicht wird. Auf dieser Grundlage und anhand festgelegter Bedingungen, z.B. den niedrigsten erwarteten Kosten und/oder der besten Anpassbarkeit, wird ein Produktionsplan ausgewählt. Nachteilig wirkt sich die hohe Laufzeit aus, weil mehrere vollständige Produktionspläne erstellt werden müssen.

5.2.1.2 Strategie 2: Führe Änderungen lokal durch

Ein bestehender Produktionsplan kann jederzeit an eine neue Situation angepasst werden. Der alte Produktionsplan sollte dabei allerdings möglichst umfangreich erhalten werden, um die Planungssicherheit zu gewährleisten. Hierzu sollten

²⁴⁴ Vgl. Kapitel 5.1.2.2.1

Änderungen des bestehenden Produktionsplans ausgelöst werden, die möglichst wenig weitere Änderungen nach sich ziehen. Damit ist es sinnvoll, die Änderung eines Produktionsplans so lokal wie möglich durchzuführen, aber so global wie nötig.

Vorteil der Strategie ist die Reduzierung der Planungsnervosität, weil ein bestehender Produktionsplan weitgehend beibehalten wird. Sind allerdings viele Änderungen an einem bestehenden Produktionsplan nötig, kann es aufwändiger sein, den bestehenden Produktionsplan zu ändern, als einen neuen Produktionsplan zu berechnen.

5.2.1.3 Strategie 3: Plane und nutze die Kapazitäten

Kapazitäten sollten geplant und falls nötig verkleinert werden, um Opportunitätskosten für ungenutzte Kapazitäten zu vermeiden und so die Kosten zu verringern. Dabei muss berücksichtigt werden, dass möglichst keine Engpässe in der Produktion entstehen, die zu einem Produktionsdefizit und schlussendlich einem Lieferverzug führen. Weiterhin sollten vorhandene Kapazitäten, die nicht mehr verkleinert werden können, ausgelastet werden.

Eine Erhöhung der Auslastung führt zu einer Erhöhung der Lagerkosten, wenn den produzierten Produkten kein Kundenauftrag gegenübersteht. Sind die gesparten Opportunitätskosten dabei höher, als die neu verursachten Kosten, sollte ein Bedarf aus der Zukunft in die Gegenwart gezogen werden. In der Zukunft kann die Kapazität dann entsprechend verkleinert werden und die Kosten auf diese Weise eingespart werden.

Die Verringerung der Kapazität kann die Kosten senken, dabei entsteht jedoch der Nachteil, dass ggf. zu wenig Pufferkapazität für Störungen zur Verfügung steht. Die Ausnutzung bestehender Kapazitäten bedingt weiterhin, dass ggf. vorzeitig Lagerbestände aufgebaut werden, die Lagerkosten verursachen. Zusammenfassend lässt sich festhalten, dass die Kapazität kurzfristig bestmöglich genutzt, aber mittelfristig geplant und ggf. angepasst werden sollte.

5.2.1.4 Strategie 4: Vergrößere den Planungshorizont

Wird der Planungshorizont vergrößert, können die Verfügbarkeit verbessert und die Kosten im Produktionsablauf gesenkt werden. Grund hierfür ist, dass auch potentiell in der Zukunft entstehende Engpässe bereits heute in der Planung berücksichtigt werden.

Vorteil der Strategie ist, dass Kapazitäten und Bestände langfristig geplant werden können und damit die Kosten gesenkt werden und die Lieferfähigkeit verbessert wird. Nachteilig wirkt sich ein ggf. erhöhter Änderungsumfang aus, weil je weiter in die Zukunft geplant wird, die Planungsdaten unsicherer werden. Ferner wirkt sich

nachteilig aus, dass sich die Laufzeit mit einer Vergrößerung des Planungshorizonts verschlechtert.

5.2.1.5 Strategie 5: Plane nicht zu fest und halte Optionen offen

Kein Produktionsplan ist unveränderbar, sondern wird bei Ereignissen, die noch nicht in der Planung berücksichtigt sind, potentiell geändert. Deshalb sollte ein Plan immer Optionen offen halten, um auf neue Situationen möglichst optimal reagieren zu können.

Diese Strategie verbessert die Anpassungsfähigkeit eines Produktionsplans auf Änderungen und ermöglicht somit eine kostengünstige Reaktion auf Störungen. Von Nachteil ist, dass ggf. Kosten für reservierte Pufferkapazität und aufgebaute Lagerbestände verursacht werden.

5.2.1.6 Strategie 6: Verwende Jobprioritäten und Prioritätsreihenfolgen

Ein vorab erstelltes Regelwerk, um z.B. Konflikte direkt zu lösen, spart Rechenzeit und ermöglicht eine effiziente Reaktion. Gleiches gilt bei der Erstellung eines Produktionsplans für vorab festgelegte Reihenfolgen bei der Auswahl eines Auftrages.

Die Laufzeit wird mit dieser Strategie verbessert, wobei es jedoch erforderlich ist, die Regeln laufend zu prüfen und ggf. an neue Situationen anzupassen.

5.2.1.7 Strategie 7: Erstelle einen Masterplan und beachte alle Restriktionen

Ein Masterplan stellt einen guten Ausgangspunkt für eingehende Änderung an den Planungsdaten dar und berücksichtigt nach Möglichkeit bereits im Vorfeld, dass Störungen auftreten können. Weiterhin verringert ein Masterplan, auf den ggf. wieder eingeschwenkt wird, die Planungsnervosität.

Werden bei der Planerstellung alle Restriktionen gleichzeitig beachtet, muss nicht sukzessive aus einem für eine Submenge der Restriktionen optimalen, aber global betrachtet ungültigen Produktionsplan, ein gültiger Produktionsplan berechnet werden.

Vorteil der Strategie ist, dass die Planungsnervosität durch einen Masterplan gesenkt wird und durch die Einhaltung aller Restriktionen keine ungültigen Produktionspläne erstellt werden.

5.2.1.8 Strategie 8: Gewichte die Perioden und verwende variable Auftragspakete

Werden die Perioden gewichtet, können Schwerpunkte auf Perioden gesetzt werden. Z.B. kann immer das gemacht werden, was jetzt besonders günstig ist. Zukünftige

Perioden können in der Planung weniger stark berücksichtigt werden, weil weit in der Zukunft liegende Abläufe durch Umplanungen ggf. nie durchgeführt werden.

In Abhängigkeit vom Bedarf sollten Auftragspakete (vgl. Definition 13) variabel gebildet werden.

Definition 13: Auftragspaket

Unter einem Auftragspaket wird die Zusammenfassung mehrerer gleichartiger Aufträge zu einem Paket verstanden. Ziel ist es, freie Kapazität durch die Reduzierung der Rüstzeit zu schaffen.

Insbesondere an Engpässen werden Pakete gebildet, um die Rüstzeiten zu verringern. Hingegen stehen an Nichtengpässen ungenutzte Kapazitäten zur Verfügung, die das Potential für häufigeres Umrüsten schaffen. Auf diese Weise werden die Aufträge insgesamt schneller durch die Produktion geschleust und die Lagerbestände verringert. Dabei ist es nicht sinnvoll, eine feste Auftragsgröße zu verwenden, sondern die Auftragspakete individuell unter Beachtung aller entscheidungsrelevanten Größen zu ermitteln.

Der Fokus dieser Strategien liegt dabei auf den sicheren Planungsdaten und zielt darauf hin, die Kosten langfristig zu senken und die Planungsnervosität zu verringern. Außerdem ermöglichen flexible Auftragspakete, die Lagerkosten zu verringern und erst bei Bedarf die Lagermengen zu Gunsten kürzerer Rüstzeiten zu erhöhen.

5.2.1.9 Strategie 9: Verringere die Bestände durch ein build-to-order Szenario

Bestände können auf allen Stufen aufgebaut werden und puffern Engpässe, verursachen aber andererseits auch Kapitalbindungskosten. Durch ein build-to-order Szenario werden die Liegezeiten des Materials reduziert, weil nur dann produziert wird, wenn für die zu produzierende Menge auch ein Kundenauftrag vorliegt. Hieraus resultiert, dass ggf. bewusst Leerzeiten an Maschinen in Kauf genommen werden, weil auf eine maximale Auslastung der Ressourcen verzichtet wird.

Durch die Verringerung der Lagerbestände werden Lagerkosten gespart, wodurch sich ggf. die Fähigkeit auf Störungen zu reagieren verschlechtert, weil keine Lagerbestände aufgebaut werden. Puffer gegen Störungen können aber auch in Form von freien Kapazitäten bestehen.

5.2.1.10 Strategie 10: Puffere Engpässe durch flexible Zwischenprodukte proaktiv

Eine an einem Engpass verlorene Stunde, ist eine verlorene Stunde für das gesamte System, denn durch Engpässe wird der maximal mögliche Output begrenzt. Engpässe bestimmen damit sowohl den Durchsatz, als auch die Bestände. Ein Engpass muss deshalb ständig mit Materialien versorgt werden, um eine maximale Auslastung des Engpasses zu erreichen. Deshalb ist es sinnvoll, vor Engpässen Bestände einzuplanen, um auf diese Weise Störungen auf vorgelagerten Stufen proaktiv zu puffern.

Mit dieser Strategie können damit Puffer gegen potentielle Störungen an Engpässen aufgebaut werden, wobei jedoch die Lagerkosten systemimmanent steigen.

5.2.1.11 Strategie 11: Reagiere sofort auf Ereignisse und repariere den Masterplan

Tritt eine Störung an einer Maschine auf oder verändert ein Kunde seinen Bedarf muss das Planungsverfahren sofort eine Entscheidung treffen und ggf. mit einer Änderung des Produktionsplans reagieren. Für eine Reaktion sollten die verfügbaren Handlungsoptionen so angewendet werden, das das Ziel verfolgt wird, den bestehenden Masterplan zu reparieren und somit möglichst wenige weitere Umplanungen zu verursachen.

Eine schnelle Reaktion auf Störungen und die Orientierung an einem Masterplan leistet damit einen Beitrag, die Lieferfähigkeit zu verbessern und die Kosten zu verringern.

5.2.1.12 Strategie 12: Nutze bestmöglich die Potentiale: Bestand, Kapazität und Zeit

Lagerbestand kann für jedes Produkt auf- oder abgebaut werden und verursacht für jede Lagereinheit periodenweise Kosten. Allerdings ermöglichen Bestände auch einen reibungslosen Fluss, indem Engpässe gepuffert werden. Bereits vorhandene Bestände müssen dabei bestmöglich verwendet werden und die Zuweisungen zur Nutzung eines Bestandes sollten laufend überprüft werden.

Liegen kapazitive Engpässe vor, verursachen Rüstzeiten hohe Kosten durch nicht für die Produktion nutzbare Kapazität. Rüstzeiten können dabei verringert werden, indem auf der gleichen Maschine mehrmals hintereinander der gleiche Rüstzustand genutzt wird. Dieses bedingt jedoch, dass bei der Planerstellung bekannt ist, welches Werkzeug in der Vorperiode verwendet wurde.

Das Potential Kapazität ist ein knappes Gut, das bestmöglich geplant und auch genutzt werden sollte (vgl. hierzu Strategie 3).

Zeit ist ein Potential, dessen Nutzung oft eine Verschiebung des Auftragstermins bedeutet. Hierdurch wird das Produktionssystem kurzfristig entlastet, weil bestehende Anforderungen in die Zukunft verschoben werden. Allerdings entstehen ggf. hohe direkte und indirekte Kosten durch Konventionalstrafen und einem Imageschaden gegenüber den Kunden.

Vorteil dieser Strategie ist, dass die Lager- und Opportunitätskosten verringert werden können und die Lieferfähigkeit verbessert werden kann. Allerdings ist die bestmögliche Wahl der Potentiale mit vielen Schwierigkeiten verbunden und eine differenzierte Abwägung erforderlich.

5.2.1.13 Strategie 13: Optimiere fortlaufend und verknüpfe Vorgänger und Nachfolger

Ein bestehender Produktionsplan sollte laufend auf Verbesserungspotential im Hinblick auf Auslastung, Rüstzeiten und Lagerkosten überprüft werden. Hierzu müssen die Beziehungen im Produktionsablauf eines Bedarfs abgebildet und beachtet werden, dass ein Produkt erst gefertigt werden kann, wenn seine Vorgänger fertig gestellt wurden und ausreichend Kapazität verfügbar ist. Spätestens muss das Produkt bis zum Bedarfstermin des Nachfolgers fertig gestellt sein.

Die Startzeit eines Auftrages ist optimal, wenn ein Produkt und alle seine Vorgänger nicht auf Lager liegen und auch keine Rüstzeiten anfallen, weil der Rüstzustand des Vorgängers auf einer Maschine direkt genutzt werden kann und kein Rüstvorgang notwendig ist.

Durch die Verknüpfung von Vorgänger und Nachfolger ermöglicht diese Strategie eine effiziente Reaktion auf Änderungen. Weiterhin führt die laufende Verbesserung eines Produktionsplans dazu, dass die Lagerkosten gesenkt werden, aber führt auch zu einer Erhöhung der Planungsnervosität.

5.2.1.14 Strategie 14: Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans

Wie sinnvoll es ist, einen Ablauf zu parallelisieren hängt stark davon ab, wie hoch der Rechenaufwand für das Problem ist und wie aufwändig die Kommunikation zwischen den parallelen Bestandteilen ist.²⁴⁵

²⁴⁵ Die meisten Algorithmen werden standardmäßig seriell implementiert. Dabei ist es durch eine Parallelisierung möglich, einen Laufzeitvorteil oder eine Verbesserung der Lösungsqualität zu erreichen. Hier existiert die interne Parallelität, bei der ein Programm selbst intern parallel abläuft, und die externe Parallelität, bei der zwei Programme zueinander parallel ausgeführt werden.

Für jeden Masterplan sollte eine Worst-Case-Analyse durchgeführt werden, die berechnet, wie groß der Mindestnutzen eines Produktionsplans ist. Hierdurch kann durch erneute Berechnung des Mindestnutzens für einen alternativen Plan entschieden werden, ob die Änderung eines bestehenden Produktionsplans auf jeden Fall zu einer Verbesserung der Lösung führt.

Vorteil dieser Strategie ist, dass die Planerstellung durch die Parallelisierung beschleunigt wird. Des Weiteren unterstützt ein Mindestnutzen die diversifizierte Reaktion auf Änderungen und bietet eine Entscheidungshilfe für die PPS.

5.2.2 Ein Beitrag zur Produktionsplanung und -steuerung

Um die Anforderungen an eine Lösung zu erfüllen, müssen verschiedene Aufgaben gelöst werden, die im Folgenden strukturiert werden. Kapitel 5.2.2.1 leistet einen Beitrag zur Erstellung eines gültigen Produktionsplans und in Kapitel 5.2.2.2 werden bestehende Lücken im Stand der Technik zur Änderung eines Produktionsplans geschlossen. Bestehende Ansätze der PPS werden in Kapitel 5.2.2.3 dahingehend erweitert, dass ggf. Pufferaufträge für Engpässe erstellt werden. Kapitel 5.2.2.4 fokussiert darauf vorhandene Ansätze der PPS zu vereinen und zu erweitern, um einen gültigen Produktionsplan mit dem Ziel niedrigerer Gesamtkosten zu optimieren. Ein Ansatz dazu einen Ausnahmefall zu beheben, wird in Kapitel 5.2.2.5 konzipiert und in Kapitel 5.2.2.6 wird ein Beitrag dazu geleistet, die Auslastung eines Produktionssystems zu verbessern.

Die entwickelten Konzepte werden in Algorithmen²⁴⁶ formalisiert und die Lösung einer Teilaufgabe in einer Komponente (vgl. Definition 14) bzw. einem Komponentencluster (vgl. Definition 15) gebündelt.

Definition 14: Komponente

Eine Komponente löst eine spezifizierte Aufgabe, wobei der Lösungsvorgang selbst als Algorithmus formuliert wird. Komponenten können von anderen Komponenten und Komponentenclustern aufgerufen werden und selbst ebenfalls Komponenten und Komponentencluster starten.

Definition 15: Komponentencluster

Ein Komponentencluster koordiniert das Zusammenspiel von zwei oder mehr Komponenten und besitzt die gleiche Funktionalität wie eine Komponente.

5.2.2.1 Erstellung eines gültigen Produktionsplans

Für die Erstellung eines Produktionsplans auf Grundlage deterministischer Planungsdaten werden die Strategien „Verwende Jobprioritäten und Prioritätsreihenfolgen“ (Strategie 6), „Erstelle einen Masterplan und beachte alle Restriktionen“ (Strategie 7), „Gewichte die Perioden und verwende variable Auftragspakete“ (Strategie 8), „Verringere die Bestände durch ein build-to-order Szenario“ (Strategie 9),

²⁴⁶ Die Formalisierung der Algorithmen erfolgt in der entwickelten Pseudocode Notation (vgl. Kapitel 5.1.2.2.1)

„Optimiere fortlaufend und verknüpfe Vorgänger und Nachfolger“ (Strategie 13) und „Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans“ (Strategie 14) auf operativer Ebene umgesetzt.²⁴⁷

Die Erstellung eines Produktionsplans erfolgt in folgendem Ablauf: In der kapazitätsorientierten Bedarfsermittlung²⁴⁸ werden zuerst die Bedarfe und die zur Verfügung stehenden Kapazitäten abgeglichen. Anschließend erfolgt die Einlastung²⁴⁹ der Aufträge, wobei eine konkrete Zuweisung von einem Auftrag zu einer Maschine vorgenommen wird.

Tritt dabei ein Planungskonflikt (vgl. Definition 16) auf, so wird versucht diesen durch einen Katalog von Handlungsoptionen²⁵⁰ sukzessiv zu beseitigen z.B. durch die Verringerung der Rüstzeit oder das Vorziehen bereits eingeplanter Lose.

Definition 16: Planungskonflikt

Ein Planungskonflikt stellt eine Situation dar, in welcher ein Kundenauftrag nicht eingelastet werden kann.

Das Komponentencluster *Planerstellung* wird dazu in Abbildung 24 entwickelt. Dabei wird es von dem Komponentenclustern *Änderungsplanung*²⁵¹ und den Komponenten *Kapazitätsanalysealgorithmus*²⁵² und *Plananalysealgorithmus*²⁵³ gestartet und verwendet selbst die Komponenten *Kapazitätsorientierte Bedarfsermittlung*²⁵⁴, *Einlastungsalgorithmus*²⁵⁵ und *Optimierungsalgorithmus*²⁵⁶. Das Komponentencluster *Planerstellung* terminiert mit der Rückgabe des erstellten Produktionsplans.

²⁴⁷ Vgl. Kapitel 5.2.1

²⁴⁸ Vgl. Kapitel 5.2.2.1.1

²⁴⁹ Vgl. Kapitel 5.2.2.1.2

²⁵⁰ Vgl. Kapitel 5.2.2.1.3

²⁵¹ Vgl. Kapitel 5.2.2.2

²⁵² Vgl. Kapitel 5.2.3.1

²⁵³ Vgl. Kapitel 5.2.3.2

²⁵⁴ Vgl. Kapitel 5.2.2.1.1

²⁵⁵ Vgl. Kapitel 5.2.2.1.2

²⁵⁶ Vgl. Kapitel 5.2.2.4

```

Program Planerstellung()
S1 begin
    /* Empfange die deterministischen Stamm- und Bewegungsdaten. */
S2 EmpfangePlanungsdaten();
    /* Starte die Komponente Kapazitätsorientierte Bedarfsermittlung. */
S3 ModifizierteBedarfsmatrix [ ] := KapazitätsorientierteBedarfsermittlung();
    /* Starte die Komponente Einlastungsalgorithmus. */
S4 Produktionsplan [ ] := Einlastungsalgorithmus();
    /* Starte die Komponente Optimierungsalgorithmus. */
S5 Produktionsplan [ ] := Optimierungsalgorithmus();
    /* Rückgabe des erstellten Produktionsplans. */
S6 return Produktionsplan [ ];
S7 end.

```

Abbildung 24: Algorithmus des Komponentenclusters Planerstellung

5.2.2.1.1 Kapazitätsorientierte Bedarfsplanung

In der kapazitätsorientierten Bedarfsermittlung wird jeder Bedarf in seine Vorprodukte aufgelöst und unter Beachtung der Kapazitäten ein Starttermin vorgeplant. Dabei erfolgt die kapazitätsorientierte Bedarfsplanung für deterministische Bedarfe und Kapazitäten in mehreren Stufen. Zuerst werden alle Maschinen auf ihre verfügbare Kapazität überprüft. Dabei wird die Kapazität paralleler Maschinen summiert und die Gesamtkapazität zusätzlich gespeichert. Ergebnis ist das Kapazitätsangebot aller Maschinen für alle Perioden des Planungszeitraumes.

Anschließend werden die Primärbedarfe aller Aufträge sukzessiv ermittelt und mit einer eindeutigen Identifikationsnummer (ID_{PB}) gespeichert, die für den gesamten Planungsablauf gültig bleibt.

Auf Basis der Primärbedarfe werden im nächsten Schritt die Sekundärbedarfe sukzessiv abgeleitet. Die Bedarfsermittlung erstellt dazu eine Datenstruktur, welche die interne und externe Nachfrage aller Produkte innerhalb eines Planungszeitraums periodengenau und auftragsspezifisch abbildet.

Anhand einer definierten Prioritätsregel erfolgt die Auswahl eines Bedarfs rückwärtsorientiert, wobei erstens nach Konfliktreihenfolge, zweitens nach Dispositionsstufe, drittens nach Produkt und viertens nach Periode gewählt wird. Für die Konfliktreihenfolge wird dabei eine dynamische Konflikttabelle erzeugt, in der die mit einem Produkt konkurrierenden Produkte gespeichert werden.

Die Sekundärbedarfsermittlung erfolgt analytisch, indem ausgehend vom Fertigerzeugnis die jeweiligen Mengen der benötigten Vorprodukte, die in den vorgelagerten Dispositionsstufen in die Fertigung einfließen, ermittelt werden. Die ermittelten Sekundärbedarfe werden mit einer Identifikationsnummer (ID_{SB}) gespeichert.

Im Anschluss erfolgt die periodenbezogene Aufteilung des Bedarfs auf die zur Verfügung stehende Kapazität. Die Bedarfe werden dabei immer so spät wie möglich eingeplant, um die Lagerkosten zu verringern, wobei immer mit einer voll ausgenutzten Periode begonnen wird und halb genutzte Perioden in die Zukunft geschoben werden. Besteht laut Konflikttabelle ein Bedarf nach einem Konfliktprodukt und überschneiden sich die Fertigungszeiten, wird die Kapazität zwischen den beiden Aufträgen unter der Abwägung, dass die anfallenden Lagerkosten den zusätzlich anfallenden Rüstkosten gegenüber gestellt werden, aufgeteilt. Andernfalls wird die Kapazität auf der Maschine für den Bedarf direkt reserviert. Die Kapazität und der noch bestehende Bedarf werden entsprechend verringert.

Kann ein Bedarf in einer Periode nicht gedeckt werden, wird sukzessiv versucht, Kapazität aus der vorherigen Periode zu nutzen. Der Ablauf erfolgt bis der Bedarf gedeckt wurde. Jedem Auftrag werden als nächstes mehrere Deadlines (vgl. Definition 17) zugewiesen.

Definition 17: Deadline

Eine Deadline kennzeichnet den spätesten Startpunkt für einen Auftrag, um eine zeitlich einzuhaltende Frist unter spezifischen Restriktionen zu erfüllen.

Ziel ist es dabei, einen Auftrag mit seinen Vorgängern und Nachfolgern zu verknüpfen sowie das Optimierungspotential abzubilden. Dazu wird im Folgenden ein mehrstufiges Deadlinekonzept entwickelt. Die Berechnung der Deadlines erfolgt ausgehend von dem Fertigstellungstermin (FS) des externen Bedarfs²⁵⁷. Für den Primärbedarf wird die Deadline Optimal (DL_{Opt}) sowie die Deadline nach Plan (DL_P) ermittelt. Von dem Primärbedarf abgeleitete Sekundärbedarfe besitzen im Gegensatz zu dem Primärbedarf zusätzliche Nachfolgerbeziehungen innerhalb der Erzeugnisstruktur. Daher wird für Sekundärbedarfe zusätzlich die Deadline Nachfolger (DL_{NF}) berechnet.

Dabei gibt die DL_{Opt} eines Bedarfes die Periode an, zu der ein Bedarf so spät wie möglich eingelastet und produziert werden kann, so dass der Fertigstellungstermin exakt eingehalten wird. Dabei ist zu beachten, dass im besten Fall die benötigte Maschine bereits korrekt gerüstet ist und eine Umrüstung, die zusätzliche Rüstzeit verursacht, entfällt. Da das benötigte Betriebsmittel zur optimalen Zeit durch andere Produkte belegt sein kann oder durch den vorhandenen Rüstzustand zusätzliche Rüstzeiten anfallen können, verschiebt sich evtl. die Startperiode des betrachteten Bedarfs. Damit kann das noch bestehende Optimierungspotential eines bereits

²⁵⁷ Im Folgenden als Primärbedarf bezeichnet

eingelasteten Auftrags in Bezug auf eine mögliche spätere Fertigung abgebildet werden.

Den Startzeitpunkt, an dem der Bedarf in der Produktion unter Berücksichtigung der Kapazität und der Berücksichtigung von Kapazitätskonflikten mit anderen Aufträgen tatsächlich eingelastet werden kann, stellt die DL_P dar. Dabei ist die Vorgänger-Nachfolger-Abhängigkeit zu beachten, in dem Fertigungsaufträge der Vorprodukte spätestens zum Startzeitpunkt des Fertigungsauftrags des Nachfolgerproduktes abgeschlossen sein müssen.

Des Weiteren repräsentiert die DL_{NF} eines Sekundärbedarfs den spätesten Startzeitpunkt eines Auftrages, so dass der Fertigstellungstermin des Auftrages die DL_P des Nachfolgebedarfs einhält. Die Berechnung der DL_{NF} erfolgt dabei ohne Berücksichtigung von ggf. auf einer Maschine bestehenden Kapazitätskonflikten und damit auch ohne Berücksichtigung bereits belegter Kapazitäten.

In Abbildung 25 wird das mehrstufige Deadline Konzept illustriert.

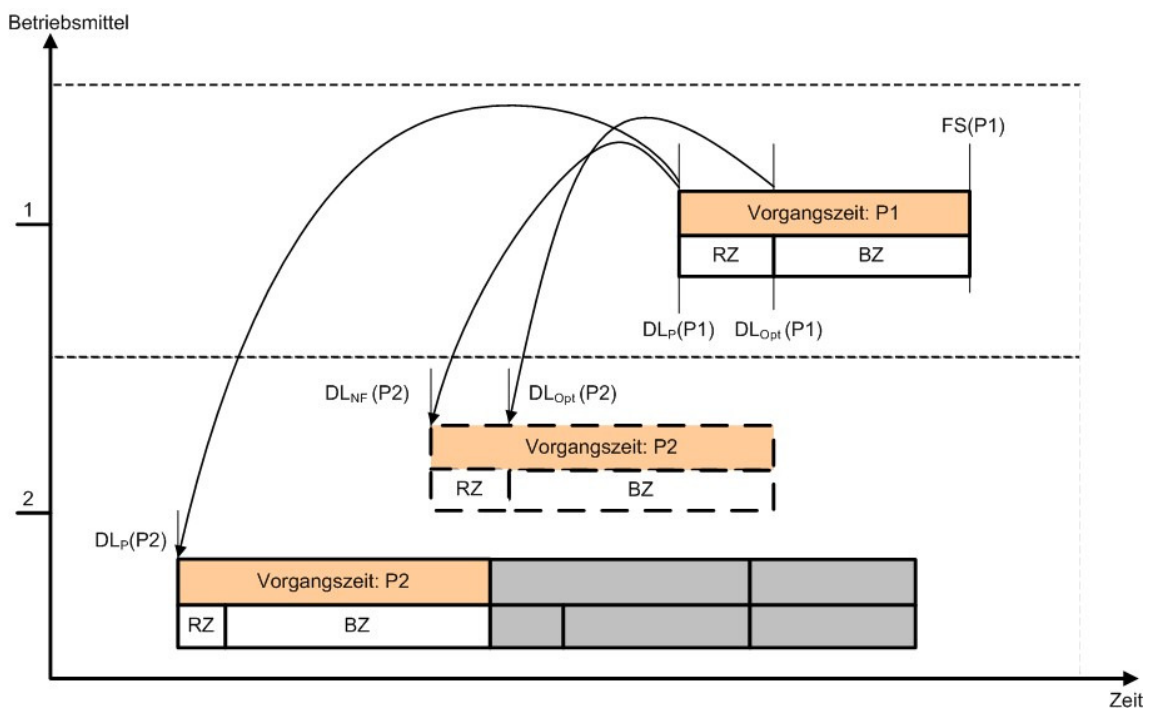


Abbildung 25: Deadlinekonzept

Für jeden Primär- und Sekundärbedarf werden die zwei bzw. drei Deadlines gespeichert und die DL_P , die DL_{Opt} und die DL_{NF} werden ggf. nachgelagert in der

Komponente *Einlastungsalgorithmus*²⁵⁸ aktualisiert. Anhand der Prioritätsreihenfolge wird anschließend der nächste Bedarf ausgewählt und rekursiv für alle weiteren Bedarfe vorgegangen.

In Abbildung 26 wird die Komponente *Kapazitätsorientierte Bedarfsermittlung* entwickelt, die von dem Komponentencluster *Planerstellung*²⁵⁹ aufgerufen wird.

Program KapazitätsorientierteBedarfsermittlung()

```

S1 begin
    /*Berechne den Bedarf für die Tiefe der Vorausschau. */
S2 for all Perioden do
S3   begin
S4     for all Produkte do
S5       begin
            /* Berechne aufgrund der Kundenbedarfe den Primärbedarf und speichere
            diesen in einer Matrix. */
S6         Primärbedarf [ ] := BerechnePrimärbedarf();
            /* Speichere die Identifikationsnummer des Primärbedarfs. */
S7         SpeichereIDPB();
            /* Berechne beginnend bei der letzten Periode rückwärtsorientiert den
            Sekundärbedarf. */
S8         Sekundärbedarf [ ] := BerechneSekundärbedarf();
            /* Speichere die Identifikationsnummern des Sekundärbedarfs. */
S9         SpeichereIDSB();
            Bedarfsmatrix [ ] := ErstelleBedarfsmatrix();
S10        end
S11      end
            /* Berechne die freie Kapazität. */
S12      BerechneKapazität();
S13      if (Bedarfsmatrix [ ] = Null) then
S14        begin
            /* Rückgabe der modifizierten Bedarfsmatrix. */
S15          return ModifizierteBedarfsmatrix [ ];
S16        end
S17      else then
S18        begin
            /* Wähle einen Bedarf nach Prioritätsreihenfolge: Konfliktreihenfolge,
            Dispositionsstufe, Periode, Produkt. */
S19          WähleBedarf();
            /* Prüfe ob ein Bedarf nach einem Konfliktprodukt besteht. */
S20          Konfliktbedarf := PrüfeKonfliktbedarf();
S21          if (Konfliktbedarf = true) then
S22            begin
                    /* Aufteilen der Kapazität. Erhöhe den Bedarf bis die Lagerkosten des
                    Konfliktprodukts größer sind als die Rüstkosten des Produkts plus die

```

²⁵⁸ vgl. Kapitel 5.2.2.1.2

²⁵⁹ Vgl. Kapitel 5.2.2.1

```

Rüstkosten des Konfliktprodukts. */
S23   SplitteKapazität();
S24   end
S25   else then
S26   begin
        /* Kapazität beachten. */
S27   Engpass := Kapazitätsaufteilung();
S28   if (Engpass = true) then
S29   begin
        /* Bedarf bei Konflikt vorziehen. */
S30   BedarfVorziehen();
S31   end
        /* Speichere die Deadline Optimal, die späteste Deadline für den
        Startzeitpunkt des Auftrags. */
S32   SpeichereDeadlineOptimalPrimärbedarf();
        /* Speichere die Deadline Plan, in der die Kapazitätskonkurrenz und der
        Liefertermin an den Kunden für den Startzeitpunkt des Auftrags
        berücksichtigt werden. */
S33   SpeichereDeadlinePlanPrimärbedarf();
        /* Speichere die Deadline Optimal, die späteste Deadline für den Startzeit
        punkt des Auftrags. */
S34   SpeichereDeadlineOptimalSekundärbedarf();
        /* Speichere die Deadline Plan für den Startzeitpunkt des Auftrags, in der die
        Kapazitätskonkurrenz und die Belieferung des Nachfolgers berücksichtigt
        werden. */
S35   SpeichereDeadlinePlanSekundärbedarf();
        /* Speichere die Deadline Nachfolger für den Startzeitpunkt des Auftrags, in
        der die Belieferung des Nachfolgers berücksichtigt wird. */
S36   SpeichereDeadlineNachfolgerSekundärbedarf();
S37   else then
S38   begin
        /* Speichere die modifizierte Bedarfsmatrix. */
S39   ModifizierteBedarfsmatrix [ ] := AktualisiereModifizierteBedarfsmatrix();
        /* Lösche den Bedarf aus der Liste noch zu betrachtender Bedarfe. */
S40   LöscheBedarf();
        /* Springe zu S13. */
S41   GoTo S13;
S42   end
S43   end
S44   end
S45 end.

```

Abbildung 26: Algorithmus der Komponente KapazitätsorientierteBedarfsermittlung

5.2.2.1.2 Einlastung der Aufträge

Auf Basis der Vorplanung durch die kapazitätsorientierte Bedarfsplanung erfolgt die Einlastung periodenweise vorwärtsorientiert und es werden parallele Maschinen beachtet. Die Auswahl eines Bedarfs erfolgt anhand einer Prioritätsreihenfolge, wobei erstens der Bedarf mit der frühesten DL_P gewählt wird und zweitens nach

Produktreihenfolge vorgegangen wird. Vorgänger-Nachfolger-Beziehungen werden damit berücksichtigt.

Der ermittelte Bruttobedarf eines nach Prioritätsreihenfolge gewählten Bedarfs wird mit den bereits vorhandenen Lagerbeständen verrechnet. Auf diese Weise wird ein vorhandener Lagerbestand so früh wie möglich verwendet und verursacht keine unnötigen Lagerkosten. Um das Optimierungspotential abzubilden, werden die Lagerbestände einem Auftrag zugeordnet und die Zuordnung entsprechend gespeichert. Mit Abschluss der Nettobedarfsrechnung liegen die konkreten noch zu produzierenden Nettobedarfmengen vor.

Für den gewählten Bedarf wird bei parallelen Maschinen die Priorität für die Wahl einer von mehreren parallelen Maschinen ermittelt. Eine prioritätsorientierte Zuordnung für die Einlastung eines Bedarfs auf einer Maschine erhöht die Wahrscheinlichkeit, ein Auftragspaket zu bilden, weil auf einer Maschine so potentiell viele gleichartige Produkte gefertigt werden.

Unter Berücksichtigung der verfügbaren Kapazität und der geplanten Rüst- und Produktionszeit erfolgt die Einlastung der noch zu produzierenden Menge auf einer Maschine. Die Einlastung eines Bedarfes erfolgt so spät wie möglich und darf die DL_{Opt} nicht überschreiten, da sonst der Fertigstellungstermin nicht eingehalten werden kann.

Tritt ein Planungskonflikt auf, wird die Komponente *Konfliktbehebungsalgorithmus*²⁶⁰ aufgerufen. Nach erfolgreicher Einlastung eines Bedarfs werden die Deadlines aktualisiert und sukzessiv der nächste Bedarf ausgewählt.

Die Komponente *Einlastungsalgorithmus* wird in Abbildung 27 entwickelt. Die Komponente *Einlastungsalgorithmus* wird von dem Komponentencluster *Planerstellung*²⁶¹ und der Komponente *Erhöhungsalgorithmus*²⁶² gestartet und ruft selbst im Bedarfsfall die Komponente *Konfliktbehebungsalgorithmus*²⁶³ auf.

²⁶⁰ Vgl. Kapitel 5.2.2.1.3

²⁶¹ Vgl. Kapitel 5.2.2.1

²⁶² Vgl. Kapitel 5.2.2.2.1

²⁶³ Vgl. Kapitel 5.2.2.1.3

```

Program Einlastungsalgorithmus()
S1 begin
S2 if (ModifizierteBedarfsmatrix [ ] = Null) then
S3 begin
    /* Rückgabe des Produktionsplans. */
S4 return Produktionsplan [ ];
S5 end
S6 else then
S7 begin
    /* Wähle einen Bedarf nach Prioritätsreihenfolge: Früheste Deadline Plan,
    Produkt. */
S8 WähleBedarf();
    /* Produktweise Priorität für die Zuordnung einer Maschine mit Look-ahead.
    */
S9 WähleMaschine();
    /* Nutze im Lager vorhandenen Bestand eines Produkts und verkleinere den
    Lagerbestand. */
S10 VerwendeBestand();
    /* Verringere den Bedarf um den verwendeten Lagerbestand. */
S11 VerringereBedarf();
    /* Ordne den Lagerbestand einem Produkt zu und speichere die Zuordnung.
    */
S12 ZuordnungBestandProdukt();
    /* Versuche, den Auftrag zu Beginn einer Periode zum spätesten möglichen
    Zeitpunkt einzulasten. Nicht später als Deadline Optimal. */
S13 Konflikt := Einlastung();
S14 if (Konflikt = true) then
S15 begin
    /* Starte die Konfliktbehebung. */
S16 Konfliktbehebung();
S17 end
S18 else then
S19 begin
    /* Speichere die Einlastung im Produktionsplan und aktualisiere den
    Produktionsplan. */
S20 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Verringere die Kapazität der belegten Maschine. */
S21 VerringereKapazität();
    /* Speichere den Termin der Fertigstellung. */
S22 SpeichereFertigstellungstermin();
    /* Aktualisiere die Deadline Plan. */
S23 UpdateDeadlinePlan();
    /* Aktualisiere die Deadline Nachfolger. */
S24 UpdateDeadlineNachfolger();
    /* Streiche den eingelasteten Bedarf aus der modifizierten Bedarfsmatrix. */
S25 StreicheBedarf();
    /* Springe zu S1. */
S26 GoTo S1;
S27 end
S28 end
S29 end.

```

Abbildung 27: Algorithmus der Komponente Einlastungsalgorithmus

5.2.2.1.3 Auflösung eines Planungskonflikts

Ein Planungskonflikt bedingt, dass der betroffene Bedarf²⁶⁴ noch nicht eingelastet ist. Die Auflösung des Planungskonflikts umfasst mehrere Handlungsalternativen, die zur Konfliktlösung sukzessiv durchlaufen werden. Dabei wird versucht, die Auswirkungen der Konfliktbehebung möglichst in die Zukunft zu schieben. Die bereits feststehende Identifikationsnummer (ID) und DL_P , DL_{NF} und DL_{Opt} eines Bedarfs werden dazu übernommen.

Um möglichst geringe Lagerkosten zu verursachen und eine Änderungsplanung möglichst weit in der Zukunft durchzuführen, erfolgt die Einlastung des Konfliktbedarfs so spät wie möglich. Für die Auflösung des Planungskonflikts werden Handlungsoptionen sukzessiv genutzt, die im Folgenden entwickelt werden.

Handlungsalternative eins versucht, eingelastete Bedarfe zu Auftragspaketen zusammenzufassen und freie Kapazitäten durch eine Reduzierung von Rüstzeiten zu schaffen.

Die Komponente *Konfliktbehebungsalgorithmus* (vgl. Abbildung 28) wird von der Komponente *Einlastungsalgorithmus*²⁶⁵ aufgerufen und startet selbst das Programm *Konfliktbehebungsalgorithmus_SchrittZwei()*, wo geprüft wird, ob immer noch ein Konfliktbedarf besteht. Ist dies nicht der Fall, wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁶⁶ geschickt.

Program Konfliktbehebungsalgorithmus()

```

S1 begin
S2   Potential := PrüfeRüstpotential();
S3   if (Potential = true) then
S4   begin
        /* Bilde Auftragspakete vor Deadline Optimal plus Produktionszeit, um
        Rüstzeit zu sparen. Kein mergen der Aufträge und beibehalten der Deadlines
        der Einzelaufträge. Die Aufträge nicht um die Rüstersparnis in die Zukunft
        verschieben. */
S5   BildeAuftragspakete();
        /* Laste den Konfliktbedarf ein. */
S6   Einlastung();
        /* Aktualisiere die verfügbare Kapazität. */
S7   AktualisiereKapazität();
        /* Verringere den Konfliktbedarf um den bereits eingelasteten Bedarf. */
S8   VerringereKonfliktbedarf();

```

²⁶⁴ Im Folgenden auch Konfliktbedarf genannt.

²⁶⁵ Vgl. Kapitel 5.2.2.1.2

²⁶⁶ Vgl. Kapitel 5.2.2.1.2

```

S9   Produktionsplan [ ] := AktualisiereProduktionsplan();
      /* Starte den Konfliktbehebungsalgorithmus_SchrittZwei.*/
S10  Konfliktbehebungsalgorithmus_SchrittZwei();
S11  end
S12  else then
S13  begin
      /* Starte den Konfliktbehebungsalgorithmus_SchrittZwei.*/
S14  Konfliktbehebungsalgorithmus_SchrittZwei();
S15  end
S16 end.

```

Abbildung 28: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Eins

Die zweite Handlungsalternative verfolgt das Ziel, die Bedarfsmenge des Konfliktbedarfs zu senken, indem vor dem Einlastungstermin des Konfliktbedarfs freie Kapazitäten dazu genutzt werden, bestehende Aufträge nach dem gleichen Produkt zu vergrößern. Die auf diese Weise zusätzlich produzierten Mengen reduzieren die Bedarfsmenge des Konfliktbedarfs.

Das Programm *Konfliktbehebungsalgorithmus_SchrittZwei()* probiert somit, durch die Vergrößerung von bereits bestehenden Aufträgen den Konfliktbedarf im besten Fall aufzulösen. Nach dem Durchlauf des Algorithmus wird das Programm *Konfliktbehebungsalgorithmus_SchrittDrei()* gestartet und geprüft, ob noch immer ein Konfliktbedarf besteht. Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁶⁷ zurückgegeben. In Abbildung 29 wird der Algorithmus des Programms konzipiert.

```

Program Konfliktbehebungsalgorithmus_SchrittZwei()
S1 begin
      /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3 begin
      /* Rückgabe des Produktionsplans. */
S4   return Produktionsplan [ ];
S5 end
S6 else then
S7 begin
S8   Potential := PrüfeVergrößerungspotential();
S9   if (Potential = true) then
S10  begin
      /* Vergrößere bestehende Aufträge vor Deadline Optimal plus
      Produktionszeit. */

```

²⁶⁷ Vgl. Kapitel 5.2.2.1.2

```

S11   Auftragsvergrößerung();
        /* Aktualisiere die verfügbare Kapazität. */
S12   AktualisiereKapazität();
        /* Verringere den Konfliktbedarf. */
S13   VerringereKonfliktbedarf();
        /* Aktualisiere den Produktionsplan. */
S14   Produktionsplan [ ] := AktualisiereProduktionsplan();
        /* Starte den Konfliktbehebungsalgorithmus_SchrittDrei.*/
S15   Konfliktbehebungsalgorithmus_SchrittDrei();
S16   end
S17   else then
S18   begin
        /* Starte den Konfliktbehebungsalgorithmus_SchrittDrei.*/
S19   Konfliktbehebungsalgorithmus_SchrittDrei();
S20   end
S21   end
S22   end.

```

Abbildung 29: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Zwei

Im nächsten Schritt wird überprüft, ob auf den nachgelagerten Stufen des Bedarfs Potential zur Verschiebung besteht. Dieses ist der Fall, wenn die DL_{OPT} des Konfliktbedarfs größer ist als die DL_{NF} plus Rüstanteil. In dem Fall besteht die Möglichkeit, die Nachfolgeprodukte später einzulasten und damit ebenfalls den Konfliktbedarf später einzulasten.

Das Programm *Konfliktbehebungsalgorithmus_SchrittDrei()* versucht, nachgelagerte Produkte der gleichen Produktionsstruktur in die Zukunft zu schieben und so Kapazität zu schaffen. Anschließend wird das Programm *Konfliktbehebungsalgorithmus_SchrittVier()* gestartet und geprüft ob noch immer ein Konfliktbedarf besteht. Ist dies nicht der Fall, wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁶⁸ zurückgegeben. In Abbildung 30 wird der Algorithmus des Programms entwickelt.

²⁶⁸ Vgl. Kapitel 5.2.2.1.2

```

Program Konfliktbehebungsalgorithmus_SchrittDrei()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3   begin
    /* Rückgabe des Produktionsplans. */
S4     return Produktionsplan [ ];
S5   end
S6 else then
S7   begin
    /* Prüfe ob Deadline Nachfolger plus Rüstanteil kleiner ist als die Deadline
    Optimal. */
S8     Potential := PrüfePotentialAufNachgelagertenStufen();
S9     if (Potential = true) then
S10      begin
    /* Erstelle ein Backup des Produktionsplans. */
S11        ErstelleBackupDesProduktionsplan();
    /* Verschiebe nachgelagerte Produkte in die Zukunft. */
S12        Konflikt := VerschiebeRekursivNachgelagertesProdukt();
S13        if (Konflikt = true) then
S14          begin
    /* Lade das Backup des Produktionsplans. */
S15            LadeBackupDesProduktionsplans();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittVier.*/
S16            Konfliktbehebungsalgorithmus_SchrittVier();
S17          end
S18        else then
S19          begin
    /* Laste den Bedarf ein. */
S20            Einlastung();
    /* Aktualisiere die verfügbare Kapazität. */
S21            AktualisiereKapazität();
    /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S22            VerringereKonfliktbedarf();
    /* Aktualisiere den Produktionsplan. */
S23            Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittVier.*/
S24            Konfliktbehebungsalgorithmus_SchrittVier();
S25          end
S26        end
S27      else then
S28        begin
    /* Starte den Konfliktbehebungsalgorithmus_SchrittVier.*/
S29          Konfliktbehebungsalgorithmus_SchrittVier();
S30        end
S31      end
S32 end.

```

Abbildung 30: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Drei

Handlungsalternative vier versucht, einen in Konkurrenz auf der gleichen Maschine zu dem Konfliktbedarf stehenden Bedarf in die Zukunft zu verschieben, wodurch auf der vorher belegten Maschine Kapazität frei wird.

Das Programm *Konfliktbehebungsalgorithmus_SchrittVier()* probiert somit, Kapazität zu schaffen, indem konkurrierende Aufträge auf der gleichen Stufe rekursiv in die Zukunft geschoben werden. Anschließend wird das Programm *Konfliktbehebungsalgorithmus_SchrittFünf()* gestartet und geprüft ob noch immer ein Konfliktbedarf besteht. Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁶⁹ zurückgegeben. In Abbildung 31 wird der Algorithmus des Programms konzipiert.

```

Program Konfliktbehebungsalgorithmus_SchrittVier()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3   begin
    /* Rückgabe des Produktionsplans. */
S4     return Produktionsplan [ ];
S5   end
S6 else then
S7   begin
    /* Prüfe ob die Deadline Nachfolger des Produkt größer ist als die Deadline
    Plan des Produkts und ob die Deadline Plan des Konfliktprodukts kleiner ist als
    die Deadline Optimal des Konfliktprodukts. */
S8   Potential := PrüfePotentialKonkurrierenderAufträge();
S9   if (Potential = true) then
S10  begin
    /* Erstelle ein Backup des Produktionsplans. */
S11  ErstelleBackupDesProduktionsplan();
    /* Verschiebe das konkurrierende Produkt in die Zukunft */
S12  Konflikt := VerschiebeRekursivKonkurrierendesProdukt();
S13  if (Konflikt = true) then
S14  begin
    /* Lade das Backup des Produktionsplans. */
S15  LadeBackupDesProduktionsplans();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittFünf.*/
S16  Konfliktbehebungsalgorithmus_SchrittFünf();
S17  end
S18  else then
S19  begin
    /* Laste den Bedarf ein. */
S20  Einlastung();
    /* Aktualisiere die verfügbare Kapazität. */
S21  AktualisiereKapazität();
    /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S22  VerringereKonfliktbedarf();

```

²⁶⁹ Vgl. Kapitel 5.2.2.1.2

```

/* Aktualisiere den Produktionsplan. */
S23 Produktionsplan [ ] := AktualisiereProduktionsplan();
/* Starte den Konfliktbehebungsalgorithmus_SchrittFünf.*/
S24 Konfliktbehebungsalgorithmus_SchrittFünf();
S25 end
S26 end
S27 else then
S28 begin
/* Starte den Konfliktbehebungsalgorithmus_SchrittFünf.*/
S29 Konfliktbehebungsalgorithmus_SchrittFünf();
S30 end
S31 end
S32 end.

```

Abbildung 31: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Vier

In Schritt fünf wird versucht, Kapazität zu schaffen, indem die Rüstzeit dadurch verringert wird, dass die Reihenfolge, in der Aufträge auf einer Maschine bearbeitet werden, getauscht wird.

Das Programm *Konfliktbehebungsalgorithmus_SchrittFünf()* versucht, die Auftragsreihenfolge zu optimieren und auf diese Weise Rüstzeit zu sparen, die wiederum als freie Kapazität für die Einlastung des Konfliktbedarfs zur Verfügung steht. Anschließend wird das Programm *Konfliktbehebungsalgorithmus_SchrittSechs()* gestartet und geprüft, ob noch immer ein Konfliktbedarf besteht. Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁷⁰ zurückgegeben. In Abbildung 32 wird der Algorithmus des Programms formalisiert.

²⁷⁰ Vgl. Kapitel 5.2.2.1.2


```

Program Konfliktbehebungsalgorithmus_SchrittFünf()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3 begin
    /* Rückgabe des Produktionsplans. */
S4 return Produktionsplan [ ];
S5 end
S6 else then
S7 begin
    /* Vor Deadline Optimal plus Produktionszeit. Wenn die Deadline Nachfolger
    des Produkts Pm minus die Deadline Plan des Produkts Pm größer ist als die
    Produktionszeit des Produkts Pn und die neue Deadline Plan des Produkts Pn
    größer ist als die Fertigstellung des Vorgängers des Produkts Pn. */
S8 Potential := PrüfeAuftragsreihenfolge();
S9 if (Potential = true) then
S10 begin
S11 TauscheAufträge();
    /* Laste den Bedarf ein. */
S12 Einlastung();
    /* Aktualisiere die verfügbare Kapazität. */
S13 AktualisiereKapazität();
    /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S14 VerringereKonfliktbedarf();
    /* Aktualisiere den Produktionsplan. */
S15 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittSechs.*/
S16 Konfliktbehebungsalgorithmus_SchrittSechs();
S17 end
S18 else then
S19 begin
    /* Starte den Konfliktbehebungsalgorithmus_SchrittSechs.*/
S20 Konfliktbehebungsalgorithmus_SchrittSechs();
S21 end
S22 end
S23 end.

```

Abbildung 32: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Fünf

Der nächste Konfliktbehebungsschritt versucht, Kapazität zu schaffen, indem Rüstzeit dadurch gespart wird, dass bei parallelen Maschinen die Maschinenbelegung von Aufträgen getauscht wird.

Das Programm *Konfliktbehebungsalgorithmus_SchrittSechs()* probiert somit, durch Tauschen der Maschinenbelegung den Konfliktbedarf einzulasten. Nach dem Durchlauf des Algorithmus wird das Programm *Konfliktbehebungsalgorithmus_SchrittSieben()* gestartet und geprüft ob noch immer ein Konfliktbedarf besteht.

Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁷¹ zurückgegeben. In Abbildung 33 wird der Algorithmus des Programms entwickelt.

```

Program Konfliktbehebungsalgorithmus_SchrittSechs()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3 begin
    /* Rückgabe des Produktionsplans. */
S4 return Produktionsplan [ ];
S5 end
S6 else then
S7 begin
    /* Prüfe ob eine parallele Maschine bereit steht und ob die Deadline Plan
    kleiner ist als die Deadline Nachfolger. */
S8 ParalleleMaschine := PrüfeParalleleMaschine();
    /* Prüfe ob Rüstpotential besteht. */
S9 Potential := PrüfeRüstpotential();
S10 if (Potential = true) AND (ParalleleMaschine = true) then
S11 begin
S12 ÄndereZuordnung();
    /* Laste den Bedarf ein. */
S13 Einlastung();
    /* Aktualisiere die verfügbare Kapazität. */
S14 AktualisiereKapazität();
    /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S15 VerringereKonfliktbedarf();
    /* Aktualisiere den Produktionsplan. */
S16 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittSieben.*/
S17 Konfliktbehebungsalgorithmus_SchrittSieben();
S18 end
S19 else then
S20 begin
    /* Starte den Konfliktbehebungsalgorithmus_SchrittSieben.*/
S21 Konfliktbehebungsalgorithmus_SchrittSieben();
S22 end
S23 end
S24 end.

```

Abbildung 33: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Sechs

Handlungsalternative sieben versucht, Kapazität für die Einlastung des Konfliktbedarfs zu schaffen, indem einem bereits zugewiesenen Lagerbestand ein Bedarfstermin

²⁷¹ Vgl. Kapitel 5.2.2.1.2

weiter in der Gegenwart zugeordnet wird und damit Kapazität in der Gegenwart auf Kosten der Zukunft geschaffen wird.

Das Programm *Konfliktbehebungsalgorithmus_SchrittSieben()* versucht damit, durch die Neuzuweisung eines Lagerbestands Kapazität in der Gegenwart zu schaffen und in der Zukunft zu nutzen. Anschließend wird das Programm *Konfliktbehebungsalgorithmus_SchrittAcht()* gestartet und geprüft ob noch immer ein Konfliktbedarf besteht. Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁷² zurückgegeben. In Abbildung 34 wird der Algorithmus des Programms formalisiert.

```

Program Konfliktbehebungsalgorithmus_SchrittSieben()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3 begin
    /* Rückgabe des Produktionsplans. */
S4 return Produktionsplan [ ];
S5 end
S6 else then
S7 begin
    /* Prüfe ob der zugeordnete Bedarfstermin für den Lagerbestand größer ist als
    der aktuelle Bedarfstermin. */
S8 Potential := PrüfeBedarfsterminLagerbestand();
S9 if (Potential = true) then
S10 begin
    /* Ändere die Zuweisung von Lagerbestand zu Auftrag. */
S11 ÄndereZuweisung();
    /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S12 VerringereKonfliktbedarf();
    /* Erhöhe den Bedarf für den Auftrag in der Zukunft. */
S13 ErhöheBedarf();
    /* Starte die Komponente Änderungsplanung. */
S14 Produktionsplan [ ] := Änderungsplanung();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittAcht.*/
S15 Konfliktbehebungsalgorithmus_SchrittAcht();
S16 end
S17 else then
S18 begin
    /* Starte den Konfliktbehebungsalgorithmus_SchrittAcht.*/
S19 Konfliktbehebungsalgorithmus_SchrittAcht();
S20 end
S21 end
S22 end.

```

Abbildung 34: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Sieben

²⁷² Vgl. Kapitel 5.2.2.1.2

Die nächste Handlungsalternative zur Konfliktbehebung versucht, Kapazität zu schaffen, indem Konkurrenzaufträge vorgezogen werden. Notwendige Vorprodukte müssen zum Zeitpunkt des neuen Starttermins bereit stehen. Zur Einhaltung dieser technologischen Restriktionen wird versucht, die Vorgängerprodukte ebenfalls vorzuziehen, so dass alle Vorgänger-Nachfolger-Beziehungen eingehalten werden. Werden Aufträge vorgezogen, muss beachtet werden, den vorherigen Rüstzustand wiederherzustellen und dafür entsprechende Pufferzeiten einzurechnen, andernfalls verändert sich die Ausgangslage bereits eingeplanter Aufträge.

Das Programm *Konfliktbehebungsalgorithmus_SchrittAcht()* probiert somit, durch Vorziehen von Konkurrenzaufträgen den Konfliktbedarf einzulasten. Nach dem Durchlauf des Algorithmus wird das Programm *Konfliktbehebungsalgorithmus_SchrittNeun()* gestartet und geprüft, ob noch immer ein Konfliktbedarf besteht. Ist dies nicht der Fall, wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁷³ zurückgegeben. In Abbildung 35 wird der Algorithmus des Programms konzipiert.

Program Konfliktbehebungsalgorithmus_SchrittAcht()

```

S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3 begin
    /* Rückgabe des Produktionsplans. */
S4 return Produktionsplan [ ];
S5 end
S6 else then
S7 begin
    /* Wenn die Deadline Plan kleiner ist als die Deadline des Nachfolgers, prüfe
    ob der Konkurrenzauftrag vorgezogen werden kann. Vor Deadline Optimal
    plus Produktionszeit. Prüfe ob der Vorgänger des Konkurrenzauftrags vor der
    Deadline Plan fertig gestellt ist. */
S8 Potential := PrüfeVorziehen();
S9 if (Potential = true) then
S10 begin
    /* Erstelle ein Backup des Produktionsplans. */
S11 ErstelleBackupDesProduktionsplan();
    /* Verschiebe den konkurrierenden Auftrags in die Gegenwart. */
S12 Konflikt := VerschiebeRekursivKonkurrierendenAuftrag();
S13 if (Konflikt = true) then
S14 begin
    /* Lade das Backup des Produktionsplans. */
S15 LadeBackupDesProduktionsplans();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittNeun.*/
S16 Konfliktbehebungsalgorithmus_SchrittNeun();

```

²⁷³ Vgl. Kapitel 5.2.2.1.2

```
S17   end
S18   else then
S19   begin
        /* Laste den Bedarf ein. */
S20   Einlastung();
        /* Aktualisiere die verfügbare Kapazität. */
S21   AktualisiereKapazität();
        /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S22   VerringereKonfliktbedarf();
        /* Aktualisiere den Produktionsplan. */
S23   Produktionsplan [ ] := AktualisiereProduktionsplan();
        /* Starte den Konfliktbehebungsalgorithmus_SchrittNeun.*/
S24   Konfliktbehebungsalgorithmus_SchrittNeun();
S25   end
S26   end
S27   else then
S28   begin
        /* Starte den Konfliktbehebungsalgorithmus_SchrittNeun.*/
S29   Konfliktbehebungsalgorithmus_SchrittNeun();
S30   end
S31   end
S32 end.
```

Abbildung 35: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Acht

Konfliktbehebungsschritt neun versucht, Auftragspakete/Aufträge aufzuteilen. Dem liegt die Annahme zu Grunde, dass kleinere Aufträge besser verschoben werden können. Anschließend wird versucht, die Aufträge in die Zukunft oder in die Gegenwart zu verschieben und damit Kapazität zu schaffen.

Das Programm *Konfliktbehebungsalgorithmus_SchrittNeun()* versucht somit, Konfliktaufträge zu splitten und anschließend den Konfliktbedarf einzulasten. Anschließend wird das Programm *Konfliktbehebungsalgorithmus_SchrittZehn()* gestartet und geprüft, ob noch immer ein Konfliktbedarf besteht. Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁷⁴ zurückgegeben. In Abbildung 36 wird der Algorithmus des Programms formalisiert.

²⁷⁴ Vgl. Kapitel 5.2.2.1.2

```

Program Konfliktbehebungsalgorithmus_SchrittNeun()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3   begin
    /* Rückgabe des Produktionsplans. */
S4     return Produktionsplan [ ];
S5   end
S6 else then
S7   begin
    /* Prüfe ob der Konfliktauftrag zusammengefügt ist. */
S8     Potential := PrüfeKonfliktauftrag();
S9     if (Potential = true) then
S10      begin
    /* Prüfe ob durch eine Teilung kein neuer Konflikt entsteht. */
S11        Konfliktfrei := PrüfeKonflikt();
S12        if (Konfliktfrei = true) then
S13          begin
    /* Splitte das Auftragspaket. */
S14            Splitting();
    /* Addiere die Rüstzeit zu den Einzelaufträgen. */
S15            RüstzeitAddieren();
    /* Verschiebe den Konfliktauftrag. */
S16            Verschieben();
    /* Laste den Bedarf ein. */
S17            Einlastung();
    /* Aktualisiere die verfügbare Kapazität. */
S18            AktualisiereKapazität();
    /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S19            VerringereKonfliktbedarf();
    /* Aktualisiere den Produktionsplan. */
S20            Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte den Konfliktbehebungsalgorithmus_SchrittZehn.*/
S21            Konfliktbehebungsalgorithmus_SchrittZehn();
S22          end
S23        else then
S24          begin
    /* Starte den Konfliktbehebungsalgorithmus_SchrittZehn.*/
S25            Konfliktbehebungsalgorithmus_SchrittZehn();
S26          end
S27        end
S28      else then
S29        begin
    /* Starte den Konfliktbehebungsalgorithmus_SchrittZehn*/
S30          Konfliktbehebungsalgorithmus_SchrittZehn();
S31        end
S32      end
S33 end.

```

Abbildung 36: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Neun

Die nächste Handlungsalternative zur Konfliktbehebung versucht, Handlungsspielraum zu schaffen, indem die Vorgänger eines Konfliktbedarfs vorgezogen werden. Notwendige Vorprodukte müssen dafür zum Zeitpunkt des neuen Starttermins bereit stehen. Weiterhin muss der vorherige Rüstzustand wiederhergestellt werden, um die Ausgangslage bereits eingeplanter Aufträge nicht zu verändern. Kann der Konfliktbedarf nach potentiell mehrfachem Durchlaufen der Handlungsalternativen nicht eingelastet werden, wird die Komponente Backloggingalgorithmus²⁷⁵ aufgerufen.

Das Programm *Konfliktbehebungsalgorithmus_SchrittZehn()* versucht damit, den Konfliktbedarf früher zu produzieren. Kann der Konflikt trotz aller Bemühungen nicht vollständig behoben werden, wird in diesem Notfall die Komponente *Backloggingalgorithmus*²⁷⁶ gestartet. Andernfalls wird der Produktionsplan an die Komponente *Einlastungsalgorithmus*²⁷⁷ zurückgegeben. In Abbildung 37 wird der Algorithmus des Programms formalisiert.

```

Program Konfliktbehebungsalgorithmus_SchrittZehn()
S1 begin
    /* Prüfe wie viel Konfliktbedarf noch besteht. */
S2 if (Konfliktbedarf = Null) then
S3   begin
    /* Rückgabe des Produktionsplans. */
S4     return Produktionsplan [ ];
S5   end
S6 else then
S7   begin
    /* Prüfe ob die Fertigstellung der Vorgänger vor der Deadline Plan liegt. */
S8     Potential := PrüfeVorgänger();
S9     if (Potential = true) then
S10    begin
    /* Erstelle ein Backup des Produktionsplans. */
S11      ErstelleBackupDesProduktionsplan();
    /* Verschiebe den konkurrierenden Auftrag in die Gegenwart. */
S12      Konflikt := VerschiebeRekursivVorgänger();
S13      if (Konflikt = true) then
S14      begin
    /* Lade das Backup des Produktionsplans. */
S15        LadeBackupDesProduktionsplans();
    /* Springe zu S31. */
S16        GoTo S31;
S17      end
S18    else then

```

²⁷⁵ Vgl. Kapitel 5.2.2.5

²⁷⁶ Vgl. Kapitel 5.2.2.5

²⁷⁷ Vgl. Kapitel 5.2.2.1.2

```
S19   begin
      /* Laste den Bedarf ein. */
S20   Einlastung();
      /* Verringere den Konfliktbedarf um den eingeplanten Bedarf. */
S21   VerringereKonfliktbedarf();
      /* Aktualisiere die verfügbare Kapazität. */
S22   AktualisiereKapazität();
      /* Aktualisiere den Produktionsplan. */
S23   Produktionsplan [ ] := AktualisiereProduktionsplan();
S24   end
S25   end
S26   else then
S27   begin
      /* Springe zu S31. */
S28   GoTo S31;
S29   end
S30   else then
S31   begin
      /* Prüfe ob weiterhin Konfliktbedarf besteht. */
S32   if (Konfliktbedarf = Null) then
S33   begin
      /* Rückgabe des Produktionsplans. */
S34   return Produktionsplan [ ];
S35   end
S36   else then
S37   begin
      /* Starte die Komponente Backloggingalgorithmus. */
S38   Backloggingalgorithmus();
S39   end
S40   end
S41 end.
```

Abbildung 37: Algorithmus der Komponente Konfliktbehebungsalgorithmus - Schritt Zehn

5.2.2.2 Änderung eines Produktionsplans

Die Änderung eines Produktionsplans basiert auf den Strategien „Führe Änderungen lokal durch“ (Strategie 2), „Reagiere online auf Ereignisse und repariere den Masterplan“ (Strategie 11) und „Nutze bestmöglich die Potentiale: Bestand, Kapazität und Zeit“ (Strategie 12).²⁷⁸

Die Lager sind bei der Ausführung der Änderungsplanung bereits teilweise gefüllt und es existiert ein Produktionsplan der umgesetzt wird. In Abhängigkeit davon, welchen Charakter eine Änderung hat, werden folgende Fälle unterschieden:

- Neuer Kundenauftrag.
- Bedarfserhöhung.
- Bedarfsverringering.
- Produktionsdefizit.
- Produktionsüberschuss.
- Kapazitätsverringering.

Ein neuer Kundenauftrag, eine zu kleine Produktionsmenge, eine Bedarfserhöhung und eine Kapazitätsverringering entsprechen einer erhöhten Produktionsmenge²⁷⁹. Eine Bedarfssenkung und eine zu große Produktionsmenge entsprechen dabei einer verringerten Produktionsmenge²⁸⁰. Die Änderungsplanung unterscheidet deshalb nur zwischen den beiden Fällen Erhöhung der Produktionsmenge²⁸¹ und Senkung der Produktionsmenge²⁸².

Das Komponentencluster *Änderungsplanung* (vgl. Abbildung 38) wird von den Komponenten *Kapazitätsanalysealgorithmus*²⁸³, *Plananalysealgorithmus*²⁸⁴, *Konflikt-*

²⁷⁸ Vgl. Kapitel 5.2.1

²⁷⁹ Im Folgenden wird unter einer erhöhten Produktionsmenge verstanden, dass ein oder mehrere der Szenarios „Neuer Kundenauftrag“, „zu kleine Produktionsmenge“, „Erhöhung eines Bedarfs“ oder „Verringerung der Kapazität“ eingetreten ist.

²⁸⁰ Im Folgenden wird unter einer verringerten Produktionsmenge verstanden, dass ein oder beide der Szenarios „Senkung eines Bedarfs“ oder „zu große Produktionsmenge“ eingetreten ist.

²⁸¹ Vgl. Kapitel 5.2.2.2.1

²⁸² Vgl. Kapitel 5.2.2.2.2

²⁸³ Vgl. Kapitel 5.2.3.1

²⁸⁴ Vgl. Kapitel 5.2.3.2

*behebungsalgorithmus*²⁸⁵, *Pufferungsalgorithmus*²⁸⁶, *Optimierungsalgorithmus*²⁸⁷ und *Nutzungsgradoptimierung*²⁸⁸ gestartet und ruft selbst die Komponenten *Erhöhungsalgorithmus*²⁸⁹ und *Senkungsalgorithmus*²⁹⁰ auf.

Program Änderungsplanung()

```

S1 begin
    /* Prüfe ob weniger Bedarf besteht, als bisher eingeplant wurde. */
S2 Bedarfssenkung := PrüfeBedarfssenkung();
    /* Prüfe ob mehr Bedarf besteht, als bisher eingeplant wurde. */
S3 Bedarfserhöhung := PrüfeBedarfserhöhung();
S4 if (Bedarfssenkung = true) then
S5 begin
    /* Starte die Bedarfssenkungskomponente und ändere den Produktionsplan. */
S6 Produktionsplan [ ] := Senkungsalgorithmus();
    /* Springe zu S1. */
S7 GoTo S1;
S8 end
S9 if (Bedarfserhöhung = true) then
S10 begin
    /* Starte die Komponente Erhöhungsalgorithmus.*/
S11 Produktionsplan [ ] := Erhöhungsalgorithmus();
    /* Springe zu S1. */
S12 GoTo S1;
S13 end
S13 else then
S14 begin
    /* Rückgabe des Produktionsplans. */
S15 return Produktionsplan [ ];
S16 end
S17 end.

```

Abbildung 38: Algorithmus des Komponentencuster Änderungsplanung

²⁸⁵ Vgl. Kapitel 5.2.2.1.3

²⁸⁶ Vgl. Kapitel 5.2.2.3

²⁸⁷ Vgl. Kapitel 5.2.2.4

²⁸⁸ Vgl. Kapitel 5.2.2.6

²⁸⁹ Vgl. Kapitel 5.2.2.2.1

²⁹⁰ Vgl. Kapitel 5.2.2.2.2

5.2.2.2.1 Erhöhung der Produktionsmenge

Die Erhöhung eines Bedarfs, ein neuer Kundenauftrag, ein Produktionsdefizit oder eine Kapazitätsverringering führen dazu, dass die Produktionsmenge ggf. erhöht wird. Ein neuer Kundenauftrag wird dabei nur akzeptiert, wenn der neue Produktionsplan gültig ist und der Mindestnutzen steigt. Andernfalls wird der Auftragstermin verschoben.

Für neue Einplanungen wird Kapazität reserviert, um Konflikte mit parallelen Einplanungen zu verhindern. Dabei wird versucht, die Auswirkungen möglichst weit in die Zukunft zu schieben und den geltenden Masterplan möglichst weitgehend beizubehalten sowie Auswirkungen einer Änderung so lokal wie möglich zu beheben.

Um die Anzahl der Änderungen mit Auswirkungen auf einen bestehenden Produktionsplan zu verringern, werden Toleranzgrenzen für die Erhöhung der Produktionsmenge verwendet. Abweichungen im Produktionsablauf liegen dabei innerhalb oder außerhalb des Toleranzbereichs.

Des Weiteren wird bei der Erhöhung der Produktionsmenge unterschieden, ob viele oder wenige Änderungen an einem bestehenden Produktionsplan nötig sind. Der Toleranzbereich für Änderungen und die Grenze für die Anzahl von Umplanungen werden dabei extern berechnet.

Mit diesen Überlegungen werden vier Fälle unterschieden:

- Die Erhöhung der Produktionsmenge liegt innerhalb des Toleranzbereichs und ein großer Änderungsumfang ist erforderlich.
- Die Erhöhung der Produktionsmenge liegt außerhalb des Toleranzbereichs und ein großer Änderungsumfang ist erforderlich.
- Die Erhöhung der Produktionsmenge liegt innerhalb des Toleranzbereichs und ein kleiner Änderungsumfang ist erforderlich.
- Die Erhöhung der Produktionsmenge liegt außerhalb des Toleranzbereichs und ein kleiner Änderungsumfang ist erforderlich.

Die Komponente *Erhöhungsalgorithmus* (vgl. Abbildung 39), wird von dem Komponentencuster *Änderungsplanung*²⁹¹ gestartet und ruft ggf. die Komponente *Einlastungsalgorithmus*²⁹² auf.

Im Fall eins wird das Szenario betrachtet, dass die Bedarfserhöhung größer als der Toleranzbereich ist und der Änderungsumfang kleiner oder gleich der Umplanungs-

²⁹¹ Vgl. Kapitel 5.2.2.2

²⁹² Vgl. Kapitel 5.2.2.1.2

grenze ist. Sind die Bedingungen für Szenario eins erfüllt wird die Bedarfserhöhung eingeplant, andernfalls wird der *Erhöhungsalgorithmus_FallZwei()* gestartet.

```

Program Erhöhungsalgorithmus()
S1 begin
    /* Berechne den Toleranzbereich.*/
S2 Toleranzbereich := Empfange(Toleranzbereich);
    /* Empfange die Umplanungsgrenze.*/
S3 Umplanungsgrenze := Empfange(Umplanungsgrenze);
    /* Analysiere den Umfang einer Änderung. */
S4 Änderung := AnalysiereÄnderungsumfang();
S5 if (Bedarf > Toleranzbereich) AND (Änderung <= Umplanungsgrenze) then
S6 begin
    /* Berechne aufgrund des Kundenbedarfs den Primärbedarf. */
S7 Primärbedarf [ ] := BerechnePrimärbedarf();
    /* Speichere die Identifikationsnummern des Primärbedarfs. */
S8 SpeichereIDPB();
    /* Berechne den Sekundärbedarf. */
S9 Sekundärbedarf [ ] := BerechneSekundärbedarf();
    /* Speichere die Identifikationsnummern des Sekundärbedarfs. */
S10 SpeichereIDSB();
    /* Berechne die freie Kapazität. */
S11 BerechneKapazität();
    /* Starte die Komponente Einlastungsalgorithmus. */
S12 Produktionsplan [ ] := Einlastungsalgorithmus();
    /* Rückgabe des Produktionsplans. */
S13 return Produktionsplan [ ];
S14 end
S15 else then
S16 begin
    /* Starte das Programm Erhöhungsalgorithmus_FallZwei. */
S17 Erhöhungsalgorithmus_FallZwei();
S18 end
S19 end.

```

Abbildung 39: Algorithmus der Komponente Erhöhungsalgorithmus

Im Fall zwei wird das Szenario betrachtet, dass die Bedarfserhöhung kleiner oder gleich dem Toleranzbereich ist und der Änderungsumfang kleiner oder gleich der Umplanungsgrenze ist. Sind die Bedingungen für Szenario zwei erfüllt, wird versucht die Bedarfserhöhung einzuplanen oder den Bedarf zurückzustellen. Andernfalls wird der *Erhöhungsalgorithmus_FallDrei()* gestartet.

Der Algorithmus wird in Abbildung 40 formalisiert.

```

Program Erhöhungsalgorithmus_FallZwei()
S1 begin
S2 if (Bedarf <= Toleranzbereich) AND (Änderung <= Umplanungsgrenze) then
S3 begin
    /* Prüfe ob es sich um einen neuen Auftrag handelt.*/
S4   NeuerAuftrag := AnalysiereBedarf();
S5   if (NeuerAuftrag = true) then
S6   begin
    /* Berechne aufgrund des Kundenbedarfs den Primärbedarf. */
S7   Primärbedarf [ ] := BerechnePrimärbedarf();
    /* Speichere die Identifikationsnummern des Primärbedarfs. */
S8   SpeichereIDPB();
    /* Berechne den Sekundärbedarf. */
S9   Sekundärbedarf [ ] := BerechneSekundärbedarf();
    /* Speichere die Identifikationsnummern des Sekundärbedarfs. */
S10  SpeichereIDSB();
    /* Berechne die freie Kapazität. */
S11  BerechneKapazität();
    /* Starte die Komponente Einlastungsalgorithmus.*/
S12  Produktionsplan [ ] := Einlastungsalgorithmus();
    /* Rückgabe des Produktionsplans. */
S13  return Produktionsplan [ ];
S14  end
S15  else then
S16  begin
    /* Keine Änderung des bestehenden Produktionsplans. */
S17  ZurückstellenDerÄnderung();
    /* Rückgabe des Produktionsplans. */
S18  return Produktionsplan [ ];
S19  end
S20  end
S21  else then
S22  begin
    /* Starte das Programm Erhöhungsalgorithmus_FallDrei. */
S23  Erhöhungsalgorithmus_FallDrei();
S24  end
S25 end.

```

Abbildung 40: Algorithmus der Komponente Erhöhungsalgorithmus - Fall Zwei

Im Fall drei wird das Szenario betrachtet, dass die Bedarfserhöhung größer ist als der Toleranzbereich und der Änderungsumfang größer als die Umplanungsgrenze ist. Trifft dies zu, wird die Bedarfserhöhung eingeplant oder ein späterer Liefertermin bestätigt. Andernfalls wird der *Erhöhungsalgorithmus_FallVier*() gestartet.

Der Algorithmus wird in Abbildung 41 entwickelt.

```

Program Erhöhungsalgorithmus_FallDrei()
S1 begin
S2 if (Bedarf > Toleranzbereich) AND (Änderung > Umplanungsgrenze) then
S3 begin
    /* Berechne den neuen Mindestnutzen. */
S4   MindestnutzenNeu := BerechneMindestnutzen();
S5 end
S6 if (MindestnutzenNeu < Mindestnutzen) then
S7 begin
    /* Prüfe ob es sich um einen neuen Auftrag handelt. */
S8   NeuerAuftrag := AnalysiereBedarf();
S9   if (NeuerAuftrag = true) then
S10  begin
    /* Bestätige einen späteren Lieferzeitpunkt, bis der Mindestnutzen steigt. */
S11  VerschiebeBedarfstermin();
    /* Rückgabe des Produktionsplans. */
S12  return Produktionsplan [ ];
S13  else then
S14  begin
    /* Berechne aufgrund des Kundenbedarfs den Primärbedarf. */
S15  Primärbedarf [ ] := BerechnePrimärbedarf();
    /* Speichere die Identifikationsnummern des Primärbedarfs. */
S16  SpeichereIDPB();
    /* Berechne den Sekundärbedarf. */
S17  Sekundärbedarf [ ] := BerechneSekundärbedarf();
    /* Speichere die Identifikationsnummern des Sekundärbedarfs. */
S18  SpeichereIDSB();
    /* Berechne die freie Kapazität. */
S19  BerechneKapazität();
S20  Produktionsplan [ ] := Einlastungsalgorithmus();
    /* Rückgabe des Produktionsplans. */
S21  return Produktionsplan [ ];
S22  end
S23  end
S24  else then
S25  begin
    /* Starte das Programm Erhöhungsalgorithmus_FallVier. */
S26  Erhöhungsalgorithmus_FallVier();
S27  end
S28 end.

```

Abbildung 41: Algorithmus der Komponente Erhöhungsalgorithmus - Fall Drei

Im Fall vier (vgl. Abbildung 42) wird das Szenario betrachtet, dass die Bedarfserhöhung kleiner oder gleich dem Toleranzbereich ist und der Änderungsumfang größer als die Umplanungsgrenze. Sind die Bedingungen für Szenario vier erfüllt, wird ein späterer Lieferzeitpunkt bestätigt oder keine Änderung vorgenommen.

```
Program Erhöhungsalgorithmus_FallVier()
S1 begin
S2   if (Bedarf <= Toleranzbereich) AND (Änderung > Umplanungsgrenze) then
S3   begin
S4     NeuerAuftrag := AnalysiereBedarf();
S5     if (NeuerAuftrag = true) then
S6       begin
S7         /* Verschiebe den Bedarfstermin. */
S7         VerschiebeBedarfstermin();
S8         /* Rückgabe des Produktionsplans. */
S8         return Produktionsplan [ ];
S9       end
S10    else then
S11    begin
S11      /* Keine Änderung des bestehenden Plans. */
S12      ZurückstellenDerÄnderung();
S13      /* Rückgabe des Produktionsplans. */
S13      return Produktionsplan [ ];
S14    end
S15  end
S16 end.
```

Abbildung 42: Algorithmus der Komponente Erhöhungsalgorithmus - Fall Vier

5.2.2.2.2 Senkung der Produktionsmenge

Wird ein Bedarf gesenkt, so wird versucht, den geltenden Masterplan möglichst weitgehend beizubehalten und die Auswirkungen rückwärtsorientiert in die Gegenwart zu ziehen. Um die Anzahl der Änderungen mit Auswirkungen auf einen bestehenden Produktionsplan zu verringern, werden ebenso, wie für die Erhöhung der Produktionsmenge, auch für die Senkung der Produktionsmenge Toleranzgrenzen verwendet. Abweichungen im Produktionsablauf liegen innerhalb oder außerhalb eines extern definierten Toleranzbereichs.

Mit diesen Vorüberlegungen wird zwischen zwei Szenarien unterschieden:

- Die Senkung der Produktionsmenge liegt außerhalb des Toleranzbereichs.
- Die Senkung der Produktionsmenge liegt innerhalb des Toleranzbereichs.

Ist die Senkung der Produktionsmenge größer als der Toleranzbereich wird ein bestehender Produktionsplan geändert, indem der Auftrag an Hand der Identifikationsnummer des Primär- und der Identifikationsnummer des Sekundärbedarfs gelöscht wird. Ist die Senkung der Produktionsmenge aber kleiner als der Toleranzbereich bleibt der Produktionsplan in der bestehenden Form erhalten.

Die Komponente *Senkungsalgorithmus* (vgl. Abbildung 43) wird dabei von dem Komponentencluster *Änderungsplanung*²⁹³ gestartet.

```

Program Senkungsalgorithmus()
S1 begin
S2 Toleranzbereich := Berechne(Toleranzbereich);
S3 if (Bedarfssenkung < Toleranzbereich) then
S4 begin
    /* Keine Änderung des bestehenden Plans. */
S5   ZurückstellenDerÄnderung();
S6 else then
S7 begin
    /* Lösche den Auftrag aus dem Produktionsplan. */
S8   LöscheAuftrag();
    /* Addiere bei zusammengesetzten Aufträgen die Rüstzeit. */
S9   AktualisiereRüstvorgänge();
    /* Prüfe ob noch eine Bedarfserhöhung vorliegt. */
S10  Bedarfserhöhung := PrüfeBedarfserhöhung();
S11  if (Bedarfserhöhung = true) then
S12  begin
    /* Rückgabe des Produktionsplans. */
S13    return Produktionsplan [ ];
S14  end
S15  else then
S16  begin
    /* Verschiebe konkurrierende Aufträge in die Zukunft. */
S17    VerschiebeKonkurrenzaufträge();
    /* Verbessere die Auslastung der freien Kapazität. */
S18    Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Rückgabe des Produktionsplans. */
S19    return Produktionsplan [ ];
S20  end
S21 end
S22 end.

```

Abbildung 43: Algorithmus der Komponente Senkungsalgorithmus

5.2.2.3 Pufferung von Engpässen

Das Prinzip Pufferung beruht auf den Strategien „Plane nicht zu fest und halte Optionen offen“ (Strategie 5), „Puffere Engpässe durch flexible Zwischenprodukte proaktiv“ (Strategie 10) und „Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans“ (Strategie 14).²⁹⁴

²⁹³ Vgl. Kapitel 5.2.2.2

²⁹⁴ Vgl. Kapitel 5.2.1

Engpässe begrenzen die Kapazität eines Produktionssystems. Aus diesem Grund müssen, um die Kapazität zu erhöhen oder diese auf einem hohen Stand zu halten, Engpässe ständig mit Vorprodukten versorgt werden. Die Einrichtung von Puffern vor einem Engpass ermöglicht dabei einen reibungslosen Fluss. Puffer können in Form von freien Kapazitäten auf vorgelagerten Produktionsstufen, in Form von Lagerbeständen oder aus einer Kombination der beiden Faktoren bestehen.

Die proaktive Pufferung von Engpässen durch den Aufbau von Lagerbeständen, hat dabei den Nachteil, dass systemimmanent höhere Lagerkosten verursacht werden. Die proaktive Pufferung von Engpässen durch das Vorhalten von Kapazitäten hat dabei den Nachteil, dass systemimmanent höhere Opportunitätskosten verursacht werden. Das Vorhalten von Kapazitäten wird an dieser Stelle nicht weiter fokussiert sondern hierzu auf Kapitel 5.2.3.1 verwiesen, wo ein Beitrag zur Planung von Kapazitäten unter Unsicherheit geleistet wird. Damit liegt der Schwerpunkt hier auf der Pufferung von Engpässen durch den gezielten Aufbau von Beständen.

Für alle kritischen Produkte werden extern vom Produktionsplaner drei feste Bestellpunkte definiert. Die Auswahl der Bestellpunkte erfolgt z.B. aufgrund folgender spezifischer Kriterien: Anzahl der Folgeprodukte, Bestellhäufigkeit, Produktionsdauer, Produkte von Zulieferern, freie Kapazität für zukünftige Perioden, Lagerkosten und Erfahrungswerte.

Die Engpasspufferung erfolgt regelbasiert, wobei laufend parallel alle Produkte geprüft werden. Für jedes Produkt werden drei Bestellpunkte extern vorgegeben. Standardmäßig werden die Bestellpunkte auf minus eins gesetzt und somit nie unterschritten und deshalb auch keine Lagerbestände aufgebaut.

Wurde ein Bestellpunkt gesetzt und der erste Bestellpunkt unterschritten, wird berechnet wann wie viel Kapazität frei ist. Dabei wird ein Bestellvorgang nur ausgelöst, wenn freie Kapazität für eine Mindestauftragsgröße vorhanden ist. Ziel ist es dabei, das Produktionssystem nicht durch Pufferaufträge zu belasten, denen noch gar kein Kundenbedarf gegenüber steht, nur um einen Puffer zu füllen. Gleichzeitig verhindert die Mindestauftragsgröße, dass das Produktionssystem durch mit kleinen Auftragsmengen verbundenen hohen Rüstzeiten belastet wird. Der Bedarfstermin für das Produkt ist der spätest mögliche Produktionsbeginn, zu dem noch freie Kapazität vorhanden ist.

Bei Unterschreitung des zweiten Bestellpunkts entfällt die Restriktion der Mindestauftragsgröße und auch kleine Aufträge werden eingeplant, wenn Kapazität verfügbar ist. Der Ablauf orientiert sich dabei an dem Ablauf bei der Unterschreitung des ersten Bestellpunkts.

Wird der dritte Bestellpunkt unterschritten, wird in Erweiterung zu der Unterschreitung des zweiten Bestellpunkts der früheste mögliche Produktionsbeginn, zu dem freie

Kapazität vorhanden ist, als Bedarfstermin vorgegeben. Der Ablauf orientiert sich an dem Ablauf bei Unterschreitung des ersten Bestellpunkts.

Die Komponente *Pufferungsalgorithmus* (vgl. Abbildung 44) prüft laufend parallel zu den anderen Komponenten ob ein Bestellpunkt für ein kritisches Produkt unterschritten wurde. Die Bestellpunkte der Produkte werden dabei ebenfalls intern parallel überwacht. Die Komponente *Pufferungsalgorithmus* startet bei Bedarf das Komponentencuster *Änderungsplanung*²⁹⁵.

Program Pufferungsalgorithmus()

```

S1 begin
S2   for all Produkte do
S3   begin
      /* Empfange den vom Produktionsplaner vorgegeben Bestellpunkte eins */
S4   BestellpunktEins [ ] := Empfange(BestellpunktEins, Produkt);
      /* Empfange den vom Produktionsplaner vorgegeben Bestellpunkte zwei */
S5   BestellpunktZwei [ ] := Empfange(BestellpunktZwei, Produkt);
      /* Empfange den vom Produktionsplaner vorgegeben Bestellpunkte drei */
S6   BestellpunktDrei [ ] := Empfange(BestellpunktDrei, Produkt);
S7   end
S8   for all Produkte do parallel
S9   begin
S10  Lagermenge [ ] := BerechneLagermenge();
S11  if (Lagermenge [ ] < BestellpunktDrei [ ]) then
S12  begin
      /* Berechne wann wie viel Kapazität auf den vorgelagerten Stufen
      vorhanden ist. */
S13  KapazitätVorhanden := AnalysiereKapazität();
S14  if (KapazitätVorhanden = true) then
S15  begin
      /* Wird der dritte Bestellpunkt unterschritten wird der früheste mögliche
      Produktionsbeginn, zu dem freie Kapazität vorhanden ist als Bedarfstermin
      vorgegeben. */
S16  ErstellePufferauftrag();
      /* Starte das Komponentencuster Änderungsplanung. */
S17  Produktionsplan [ ] := Änderungsplanung();
      /* Rückgabe des Produktionsplans. */
S18  return Produktionsplan [ ];
S19  end
S20  end
S21  else if (Lagermenge [ ] < BestellpunktZwei [ ]) then
S22  begin
      /* Berechne wann wie viel Kapazität auf den vorgelagerten Stufen
      vorhanden ist. */
S23  KapazitätVorhanden := AnalysiereKapazität();
S24  if (KapazitätVorhanden = true) then

```

²⁹⁵ Vgl. Kapitel 5.2.2.2

```
S25   begin  
      /* Wird der zweite Bestellpunkt unterschritten wird der späteste mögliche  
      Produktionsbeginn, zu dem freie Kapazität vorhanden ist als  
      Bedarfstermin vorgegeben. Ohne Mindestauftragsmenge. */  
S26   ErstellePufferauftrag();  
      /* Starte das Komponentencluster Änderungsplanung. */  
S27   Produktionsplan [ ] := Änderungsplanung();  
      /* Rückgabe des Produktionsplans. */  
S28   return Produktionsplan [ ];  
S29   end  
S30   end  
S31   else if (Lagermenge [ ] < BestellpunktEins [ ]) then  
S32   begin  
      /* Berechne wann wie viel Kapazität auf den vorgelagerten Stufen  
      vorhanden ist. */  
S33   KapazitätVorhanden := AnalysiereKapazität();  
S34   if (KapazitätVorhanden = true) then  
S35   begin  
      /* Wird der erste Bestellpunkt unterschritten wird der späteste mögliche  
      Produktionsbeginn, zu dem Kapazität für die Mindestauftragsmenge  
      vorhanden ist, als Bedarfstermin vorgegeben. */  
S36   ErstellePufferauftrag();  
      /* Starte das Komponentencluster Änderungsplanung. */  
S37   Produktionsplan [ ] := Änderungsplanung();  
      /* Rückgabe des Produktionsplans. */  
S38   return Produktionsplan [ ];  
S39   end  
S40   end  
S41   else then  
S42   begin  
      /* Rückgabe des Produktionsplans. */  
S43   return Produktionsplan [ ];  
S44   end  
S45   end  
S46 end.
```

Abbildung 44: Algorithmus der Komponente Pufferungsalgorithmus

5.2.2.4 Optimierung eines gültigen Produktionsplans

Die Optimierung eines Produktionsplans gründet sich auf die Strategien „Führe Änderungen lokal durch“ (Strategie 2), „Plane und nutze die Kapazitäten“ (Strategie 3), „Verwende Jobprioritäten und Prioritätsreihenfolgen“ (Strategie 6), „Reagiere online auf Ereignisse und repariere den Masterplan“ (Strategie 11), „Nutze bestmöglich die Potentiale: Bestand, Kapazität und Zeit“ (Strategie 12), „Optimiere fortlaufend und verknüpfe Vorgänger und Nachfolger“ (Strategie 13) und „Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans“ (Strategie 14).²⁹⁶

Ziel ist es, einen bestehenden Masterplan in Hinblick auf ungenutzte Potentiale zu verbessern. Die Optimierung läuft vorwärts orientiert und strebt eine kontinuierliche Verbesserung eines bestehenden Produktionsplans an. Hierzu wird ein Katalog von Optimierungspotentialen sukzessiv überprüft, z.B. im Hinblick auf die Dauer der Lagerzeit, die Höhe der Lagerbestände, die Verwendung der Lagerbestände und die Länge der Rüstzeit.

Des Weiteren kann durch die Bildung von Auftragspaketen Kapazität geschaffen, die entweder verringert werden kann oder von Aufträgen, die aus der Zukunft vorgezogen werden, genutzt wird. In der Zukunft kann die frei werdende Kapazität dann verringert werden. Dabei werden die gesparten Opportunitätskosten mit den zusätzlichen Lagerkosten verglichen.

Die Komponente *Optimierungsalgorithmus* (vgl. Abbildung 45) wird von dem Komponentencenter *Planerstellung*²⁹⁷ aufgerufen und sucht laufend parallel zur Ausführung eines Masterplans nach Optimierungspotential. Die Komponente *Optimierungsalgorithmus* ruft das Komponentencenter *Änderungsplanung*²⁹⁸ auf. Dabei gliedert sich die Komponente in neun Schritte, die sukzessiv gestartet werden. Rückgabewert ist der potentiell optimierte Produktionsplan.

Die Komponente *Optimierungsalgorithmus()* sucht nach Optimierungspotential, indem es prüft ob die Aufträge später gefertigt werden können. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittZwei()* gestartet.

²⁹⁶ Vgl. Kapitel 5.2.1

²⁹⁷ Vgl. Kapitel 5.2.2.1

²⁹⁸ Vgl. Kapitel 5.2.2.2

```

Program Optimierungsalgorithmus()
S1 begin
    /* Prüfe ob Deadline Plan plus Rüstanteil kleiner ist als die Deadline
    Optimal.*/
S2 Potential := PrüfePotential();
S3 if (Potential = true) then
S4 begin
    /* Prüfe ob für den Auftrag oder seine Nachfolger in der Zukunft Kapazität
    verfügbar ist, um den Auftrag zu verschieben. Rückwärts orientiert. */
S5 Kapazität := PrüfeKapazität();
S6 if (Kapazität = true) then
S7 begin
    /* Senke den Bedarf.*/
S8 Bedarfssenkung();
    /* Aktualisiere die verfügbare Kapazität. */
S9 AktualisiereKapazität();
    /* Erhöhe den Bedarf zu einem späteren Zeitpunkt. */
S10 Bedarfserhöhung();
    /* Starte die Komponente Änderungsplanung.*/
S11 Produktionsplan [ ] := Änderungsplanung();
    /* Starte das Programm Optimierungsalgorithmus_SchrittZwei. */
S12 Optimierungsalgorithmus_SchrittZwei();
S13 end
S14 else then
S15 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittZwei. */
S16 Optimierungsalgorithmus_SchrittZwei();
S17 end
S18 end
S19 else then
S20 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittZwei. */
S21 Optimierungsalgorithmus_SchrittZwei();
S22 end
S23 end.

```

Abbildung 45: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Eins

Das Programm *Optimierungsalgorithmus_SchrittZwei()* versucht, Rüstzeiten zu sparen und damit Kapazität zu schaffen, indem es aufeinander folgende gleichartige Aufträge bündelt. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittDrei()* gestartet.

In Abbildung 46 wird der Algorithmus konzipiert.

```

Program Optimierungsalgorithmus_SchrittZwei()
S1 begin
    /* Prüfe ob Deadline Plan plus Rüstanteil kleiner ist als die Deadline
    Optimal.*/
S2 Potential := PrüfePotential();
    /* Prüfe ob Rüstpotential besteht. */
S3 Rüstpotential := PrüfeRüstpotential();
S4 if (Potential = true) AND (Rüstpotential = true) then
S5 begin
    /* Bilde Auftragspakete um Rüstzeit zu sparen, aber kein mergen der Aufträge
    und beibehalten der Deadlines der Einzelaufträge. Die Aufträge nicht um die
    Rüstersparnis in die Zukunft verschieben. */
S6 BildeAuftragspakete();
    /* Aktualisiere den bestehenden Produktionsplan. */
S7 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte das Programm Optimierungsalgorithmus_SchrittDrei. */
S8 Optimierungsalgorithmus_SchrittDrei();
S9 end
S10 else then
S11 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittDrei. */
S12 Optimierungsalgorithmus_SchrittDrei();
S13 end
S14 end.

```

Abbildung 46: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Zwei

Das Programm *Optimierungsalgorithmus_SchrittDrei()* optimiert die Auftragsreihenfolge und versucht, die Rüstzeiten zu verringern. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittVier()* gestartet.

In Abbildung 47 wird der Algorithmus entwickelt.

```

Program Optimierungsalgorithmus_SchrittDrei()
S1 begin
    /* Prüfe ob Deadline Plan plus Rüstanteil kleiner ist als die Deadline
    Optimal.*/
S2 Potential := PrüfePotential();
S3 if (Potential = true) then
S4 begin
    /* Prüfe ob durch ein Tauschen der Reihenfolge Lagerkosten gespart werden
    können. */
S5 Lagerpotential := PrüfeLagerpotential();
    /* Prüfe ob die Deadline Nachfolger des zweiten Produkten minus die
    Deadline Plan des zweiten Produkt größer ist als die Produktionszeit des
    ersten Produkts. */
S6 Nachfolgerpotential := PrüfeNachfolgerpotential();
    /* Prüfe ob die Deadline Plan des zweiten Produkten größer ist als der

```

```

    Fertigstellungstermin des Vorgängers. */
S7  Vorgängerpotential := PrüfeNachfolgerpotential();
S8  if (Lagerpotential = true) AND (Vorgängerpotential = true) AND (Nachfolgerpo-
    tential = true) then
S9  begin
    /* Produziere Produkt eins später und Produkt zwei früher. */
S10 TauscheReihenfolge();
    /* Aktualisiere den bestehenden Produktionsplan. */
S11 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte das Programm Optimierungsalgorithmus_SchrittVier. */
S12 Optimierungsalgorithmus_SchrittVier();
S13 end
S14 else then
S15 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittVier. */
S16 Optimierungsalgorithmus_SchrittVier();
S17 end
S18 end
S19 else then
S20 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittVier. */
S21 Optimierungsalgorithmus_SchrittVier();
S22 end
S23 end.

```

Abbildung 47: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Drei

Das Programm *Optimierungsalgorithmus_SchrittVier()* versucht, ein konkurrierendes Produkt in die Zukunft zu verschieben. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittFünf()* gestartet.

In Abbildung 48 wird der Algorithmus konzipiert.

```

Program Optimierungsalgorithmus_SchrittVier()
S1 begin
    /* Prüfe ob Deadline Plan kleiner ist als die Deadline Nachfolger. */
S2 PotentialAuftrag := PrüfePotentialAufNachgelagertenStufen();
    /* Prüfe ob Deadline die Plan des Konkurrenzauftrages kleiner ist als die
    Deadline Optimal des Konkurrenzauftrages. */
S3 PotentialKonkurrenzauftrag := PrüfePotentialKonkurrenzauftrag();
S4 if (PotentialAuftrag = true) AND (PotentialKonkurrenzauftrag) then
S5 begin
    /* Speichere ein Backup des Produktionsplans. */
S6 ErstelleBackupDesProduktionsplans();
    /* Verschiebe nachgelagerte Produkte in die Zukunft. */
S7 Konflikt := VerschiebeRekursivNachgelagertesProdukt();
S8 if (Konflikt = true) then
S9 begin
    /* Lade das gespeicherte Backup des Produktionsplans. */
S10 Produktionsplan [ ] := LadeBackupDesProduktionsplans();

```

```

/* Starte das Programm Optimierungsalgorithmus_Schritt5. */
S11   Optimierungsalgorithmus_Schritt5();
S12   end
S13   else then
S14   begin
        /* Aktualisiere die verfügbare Kapazität. */
S15   AktualisiereKapazität();
        /* Verschiebe einen bestehenden Bedarf. */
S16   VerschiebeAuftrag();
        /* Aktualisiere den Produktionsplan. */
S17   Produktionsplan [ ] := AktualisiereProduktionsplan();
        /* Starte das Programm Optimierungsalgorithmus_SchrittFünf. */
S18   Optimierungsalgorithmus_SchrittFünf();
S19   end
S20   end
S21   else then
S22   begin
        /* Starte das Programm Optimierungsalgorithmus_SchrittFünf. */
S23   Optimierungsalgorithmus_SchrittFünf();
S24   end
S25   end.

```

Abbildung 48: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Vier

Das Programm *Optimierungsalgorithmus_SchrittFünf()* versucht, nachgelagerte Bestandteile des Endprodukts in die Zukunft zu verschieben. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittSechs()* gestartet.

Der Algorithmus wird in Abbildung 49 formalisiert.

```

Program Optimierungsalgorithmus_SchrittFünf()
S1 begin
        /* Prüfe ob Deadline Nachfolger plus Rüstanteil kleiner ist als die Deadline
        Optimal. */
S2 PotentialAuftrag := PrüfePotentialAufNachgelagertenStufen();
S3 if (PotentialAuftrag = true) then
S4   begin
            /* Speichere ein Backup des Produktionsplans. */
S5   ErstelleBackupDesProduktionsplans();
            /* Verschiebe nachgelagerte Aufträge in die Zukunft. */
S6   Konflikt := VerschiebeRekursivNachgelagerteProdukte();
S7   if (Konflikt = true) then
S8     begin
                /* Lade das gespeicherte Backup des Produktionsplans. */
S9     Produktionsplan [ ] := LadeBackupDesProduktionsplans();
                /* Starte das Programm Optimierungsalgorithmus_SchrittSechs. */
S10    Optimierungsalgorithmus_SchrittSechs();
S11    end
S12    else then
S13    begin

```



```

S14      /* Aktualisiere den Produktionsplan. */
        Produktionsplan [ ] := AktualisiereProduktionsplan();
        /* Starte das Programm Optimierungsalgorithmus_SchrittSechs. */
S15      Optimierungsalgorithmus_SchrittSechs();
S16      end
S17      end
S18      else then
S19      begin
        /* Starte das Programm Optimierungsalgorithmus_SchrittSechs. */
S20      Optimierungsalgorithmus_SchrittSechs();
S21      end
S22      end.

```

Abbildung 49: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Fünf

Das Programm *Optimierungsalgorithmus_SchrittSechs()* versucht, Aufträge, die vor dem Liefertermin fertig gestellt werden, später zu fertigen. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittSieben()* gestartet.

In Abbildung 50 wird der Algorithmus entwickelt.

```

Program Optimierungsalgorithmus_SchrittSechs()
S1 begin
    /* Prüfe ob die Fertigstellung des Auftrages kleiner ist als der Bedarfstermin des
    Kunden. */
S2 PotentialAuftrag := PrüfePotential();
S3 if (PotentialAuftrag = true) then
S4 begin
    /* Speichere ein Backup des Produktionsplans. */
S5 ErstelleBackupDesProduktionsplans();
    /* Verschiebe vorgelagerte Aufträge sukzessiv in die Zukunft. */
S6 Konflikt := VerschiebeRekursivVorgelagerteProdukte();
S7 if (Konflikt = true) then
S8 begin
    /* Lade das gespeicherte Backup des Produktionsplans. */
S9 Produktionsplan [ ] := LadeBackupDesProduktionsplans();
    /* Starte das Programm Optimierungsalgorithmus_SchrittSieben. */
S10 Optimierungsalgorithmus_SchrittSieben();
S11 end
S12 else then
S13 begin
    /* Aktualisiere den Produktionsplan. */
S14 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Starte das Programm Optimierungsalgorithmus_SchrittSieben. */
S15 Optimierungsalgorithmus_SchrittSieben();
S16 end
S17 end
S18 else then
S19 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittSieben. */

```

```

S20   Optimierungsalgorithmus_SchrittSieben();
S21   end
S22 end.

```

Abbildung 50: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Sechs

Das Programm *Optimierungsalgorithmus_SchrittSieben()* versucht, den Lagerbestand bestmöglich zu verwenden und möglichst geringe Lagerkosten zu verursachen. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittAcht()* gestartet.

In Abbildung 51 wird der Algorithmus formalisiert.

```

Program Optimierungsalgorithmus_SchrittSieben()
S1 begin
    /* Prüfe ob der Bedarfstermin für einen Lagerbestand später liegt als ein
    Konkurrenzbedarfstermin für den Lagerbestand. */
S2   Potential := PrüfeBedarfstermin();
S3   if (Potential = true) then
S4   begin
        /* Ändere die Zuweisung eines Lagerbestandes. */
S5   ÄndereZuweisung();
        /* Senke den Bedarf für ein Produkt. */
S6   Bedarfssenkung();
        /* Erhöhe den Bedarf nach dem Produkt zu einem späteren Zeitpunkt. */
S7   Bedarfserhöhung();
        /* Starte die Komponente Änderungsplanung. */
S8   Produktionsplan [ ] := Änderungsplanung();
        /* Starte das Programm Optimierungsalgorithmus_SchrittAcht. */
S9   Optimierungsalgorithmus_SchrittAcht();
S10  end
S11  else then
S12  begin
        /* Starte das Programm Optimierungsalgorithmus_SchrittAcht. */
S13  Optimierungsalgorithmus_SchrittAcht();
S14  end
S15 end.

```

Abbildung 51: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Sieben

Das Programm *Optimierungsalgorithmus_SchrittAcht()* versucht, Fertigungsaufträge bestmöglich zu nutzen. Anschließend wird das Programm *Optimierungsalgorithmus_SchrittNeun()* gestartet.

In Abbildung 52 wird der Algorithmus konzipiert.

```

Program Optimierungsalgorithmus_SchrittAcht()
S1 begin
    /* Prüfe ob der zugewiesene Bedarfstermin für einen internen Auftrag später
    liegt als ein Konkurrenzbedarfstermin. */
S2 Potential := PrüfeBedarfstermin();
S3 if (Potential = true) then
S4 begin
    /* Ändere die Zuordnung eines internen Auftrages. */
S5 ÄndereZuweisung();
    /* Aktualisiere die Identifikationsnummer des Sekundärbedarfs. */
S6 AktualisiereSekundärbedarfID();
    /* Starte die Komponente Änderungsplanung. */
S7 Produktionsplan [ ] := Änderungsplanung();
    /* Starte das Programm Optimierungsalgorithmus_SchrittNeun. */
S8 Optimierungsalgorithmus_SchrittNeun();
S9 end
S10 else then
S11 begin
    /* Starte das Programm Optimierungsalgorithmus_SchrittNeun. */
S12 Optimierungsalgorithmus_SchrittNeun();
S13 end
S14 end.

```

Abbildung 52: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Acht

Das Programm *Optimierungsalgorithmus_SchrittNeun()* sucht nach Optimierungspotential auf parallelen Maschinen, um die Summe aus Lagerkosten und Rüstkosten zu senken. Anschließend wird der optimierte Produktionsplan zurückgegeben.

In Abbildung 53 wird der Algorithmus entwickelt.

```

Program Optimierungsalgorithmus_SchrittNeun()
S1 begin
    /* Prüfe ob eine parallele Maschine bereit steht und ob die zusätzlichen
    Lagerkosten für die beiden Produkte minus das Rüstpotential minus die
    eingesparten Lagerkosten größer als Null sind. */
S2 Potential := PrüfeBedarfstermin();
    /* Prüfe ob durch einen Tausch ein Konflikt entsteht. */
S3 Konfliktfrei := PrüfeKonflikt();
S4 if (Potential = true) AND (Konfliktfrei = true)then
S5 begin
    /* Ändere die Zuweisung zu einer Maschine. */
S6 ÄndereZuweisung();
    /* Aktualisiere die verfügbare Kapazität. */
S7 AktualisiereKapazität();
    /* Aktualisiere den Produktionsplan. */
S8 Produktionsplan [ ] := AktualisiereProduktionsplan();
    /* Rückgabe des optimierten Produktionsplans. */
S9 return Produktionsplan [ ];

```

```

S10 end
S11 else then
S12 begin
    /* Rückgabe des Produktionsplans. */
S13     return Produktionsplan [ ];
S14 end
S15 end.

```

Abbildung 53: Algorithmus der Komponente Optimierungsalgorithmus - Schritt Neun

5.2.2.5 Behandlung von Ausnahmen

Das Prinzip Ausnahmebehandlung fußt auf den Strategien „Nutze bestmöglich die Potentiale: Bestand, Kapazität und Zeit“ (Strategie 12) und „Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans“ (Strategie 14).²⁹⁹

Vor jeder Auftragsannahme muss geprüft werden, ob es möglich ist den Auftrag fristgerecht auszuführen. Kommt es aufgrund einer schlechten Planung oder Störung dennoch dazu, dass ein Auftrag nicht fristgerecht erfüllt werden kann, muss es möglich sein, Aufträge im Ausnahmefall später zu liefern. Dazu werden die Aufträge ermittelt, die zurückgestellt werden müssen und damit auch später gefertigt werden können, um möglichst geringe Gesamtkosten über alle Perioden zu verursachen und wieder lieferfähig zu sein.

Die Komponente *Backloggingalgorithmus* (vgl. Abbildung 54) wird von der Komponente *Konfliktbehebungsalgorithmus*³⁰⁰ gestartet, wenn ein Konflikt nicht gelöst werden kann und ruft selbst das Komponentencluster *Änderungsplanung*³⁰¹ auf.

```

Program Backloggingalgorithmus()
S1 begin
    /* Ermittle den verbleibenden Bedarf. */
S2     BerechneBedarf();
    /* Ermittle verschiedene Pakete von Aufträgen die zurückgestellt werden müssen
    um wieder lieferfähig zu sein. */
S3     BerechneAuftragspakete();
    /* Berechne die gesamten Verspätungskosten der Pakete. */
S4     BerechneBackloggingkosten();
    /* Wähle die besten Pakete. */
S5     WählePakete();
    /* Bilde hieraus wieder mehrere Pakete und ermittle die gesamten

```

²⁹⁹ Vgl. Kapitel 5.2.1

³⁰⁰ Vgl. Kapitel 5.2.2.1.3

³⁰¹ Vgl. Kapitel 5.2.2.2

```

    Verspätungskosten. */
S6 BerechneAuftragspakete();
    /* Berechne die gesamten Verspätungskosten. */
S7 BerechneBackloggingkosten();
    /* Wähle das beste Paket. */
S8 WählePaket();
    /* Streiche die Aufträge. */
S9 SenkeBedarf();
    /* Aktualisiere die zur Verfügung stehende Kapazität. */
S10 AktualisiereKapazität();
    /* Erhöhe den Bedarf und laste die Aufträge später wieder ein. */
S11 ErhöheBedarf();
    /* Starte das Komponentencluster Änderungsplanung und aktualisiere den
    Produktionsplan. */
S12 Produktionsplan [ ] := Änderungsplanung();
    /* Rückgabe des Produktionsplans. */
S13 return Produktionsplan [ ];
S14 end.

```

Abbildung 54: Algorithmus der Komponente Backloggingalgorithmus

5.2.2.6 Verbesserung der Auslastung eines Produktionssystems

Das Prinzip Nutzungsgrad basiert auf den Strategien „Führe Änderungen lokal durch“ (Strategie 2), „Plane und nutze die Kapazitäten“ (Strategie 3), „Verwende Jobprioritäten und Prioritätsreihenfolgen“ (Strategie 6), „Nutze bestmöglich die Potentiale: Bestand, Kapazität und Zeit“ (Strategie 12), „Optimiere fortlaufend und verknüpfe Vorgänger und Nachfolger“ (Strategie 13) und „Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans“ (Strategie 14).³⁰²

Dabei existieren mehrere Gründe, die Auslastung des Produktionssystems zu verbessern. Zum einen kann das bestehende Kapazitätsangebot am Anfang des Planungszeitraumes ausgelastet werden und verursacht keine bzw. geringere Opportunitätskosten. Zum anderen wird Potential für später eingehende Aufträge geschaffen, weil auf den, vor dem Vorziehen von den Aufträgen belegten Maschinen, Kapazität frei wird. Des Weiteren kann versucht werden, die in der Zukunft so geschaffene Kapazität zu verringern.

Zur Verbesserung der Auslastung wird versucht, die Kapazitätsauslastung der einzelnen Maschinen in der ersten und zweiten Planungswoche zu erhöhen. Dazu werden eigentlich bereits später eingeplante Aufträge ggf. in den Zeitraum verschoben, deren Nutzungsgrad erhöht werden soll (vgl. Abbildung 55).

³⁰² Vgl. Kapitel 5.2.1

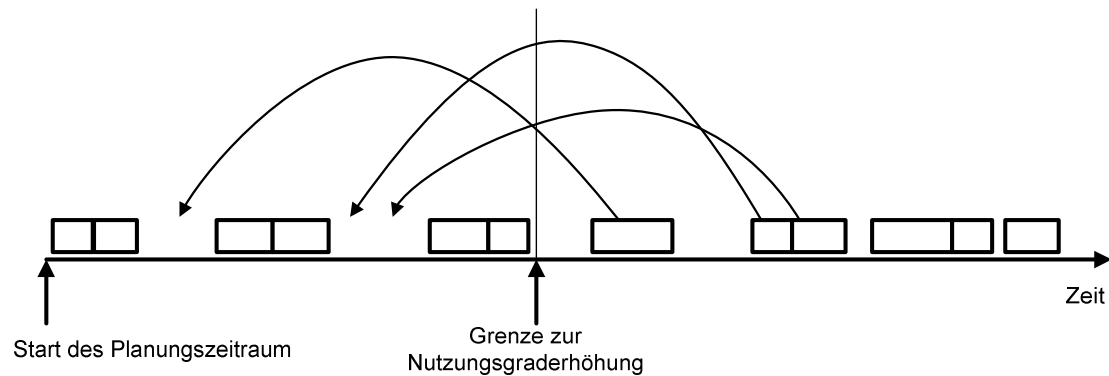


Abbildung 55: Verbesserung der Auslastung eines Produktionssystems

Die Potentiale Kapazität, Rüstzustand und Rüstvorgänge werden vorwärts orientiert optimiert. Ein Bedarf, der in der Zukunft liegt wird dabei vorgezogen, wenn die verkleinerten Opportunitätskosten plus die evtl. Rüstersparnis höher sind als die Lagerkosten plus die ggf. zusätzlich anfallenden Rüstkosten.

Die Komponente *Nutzungsgradoptimierung* wird von der Komponente *Plananalysealgorithmus*³⁰³ aufgerufen und startet selbst das Komponentencluster *Änderungsplanung*³⁰⁴. Dabei wird die Ausnutzung der Kapazitäten in der Woche eins und zwei verbessert.

Der Algorithmus wird in Abbildung 56 entwickelt.

Program Nutzungsgradoptimierung()

```

S1 begin
    /* Ermittle die freie Kapazität in Woche eins und zwei. */
S2 ErmittleKapazität();
    /* Ermittle ob die Lagerkosten bis zum Bedarfszeitpunkt eines Produkts <
    Opportunitätskosten für die freie Kapazität und ob die Fertigstellung des
    Vorgängers vor der neuen Deadline Plan ist. Freie Kapazität vorwärts orientiert
    auslasten. */
S3 Potential := Vergrößerungspotential();
S4 if (Potential = true) then
S5 begin
    /* Vergrößere bestehende Aufträge und nutze die Kapazität vor und nach dem
    bestehendem Auftrag. */
S6 Auftragsvergrößerung();
    /* Senke den Bedarf, der durch die Vergrößerung eines Auftrages früher
  
```

³⁰³ Vgl. Kapitel 5.2.3.2

³⁰⁴ Vgl. Kapitel 5.2.2.2

```

S7   Bedarfssenkung();
      /* Aktualisiere die verfügbare Kapazität. */
S8   AktualisiereKapazität()
      /* Starte das Komponentencluster Änderungsplanung. */
S9   Produktionsplan [ ] := Änderungsplanung();
      /* Sperre die verschobenen Bedarfe für eine Woche für die Komponente
      Optimierungsalgorithmus. */
S10  OptimierungsalgorithmusSperren();
S11  end
      /* Prüfe ob die Lagerkosten plus die Rüstkosten für ein Splitting des Auftrages
      kleiner sind als die Opportunitätskosten und ob die Fertigstellung des
      Vorgängers vor der neuen Deadline Plan ist. Freie Kapazität vorwärts orientiert
      auslasten. */
S12  Nutzungspotential := ErmittleNutzungspotential();
S13  if (Nutzungspotential = true) then
S14  begin
      /* Für alle Aufträge die in dem Zeitraum gefertigt werden in dem die Kapazität
      noch angepasst werden kann. */
S15  for all Aufträge do
S16  begin
      /* Prüfe ob ein Auftrag vorgezogen werden kann. */
      Vorziehen := PrüfeVorziehen();
S17  if (Vorziehen = true) then
S18  begin
      /* Ziehe Aufträge vor und laste die Kapazität aus. */
S19  AufträgeVorziehen();
      /* Senke den Bedarf, der durch die Vergrößerung eines Auftrages früher
      gefertigt wird. */
S20  Bedarfssenkung();
      /* Aktualisiere die verfügbare Kapazität. */
S21  AktualisiereKapazität()
      /* Sperre die verschobenen Bedarfe für eine Woche für die Komponente
      Optimierungsalgorithmus. */
S22  OptimierungsalgorithmusSperren();
      /* Starte das Komponentencluster Änderungsplanung.*/
S23  Produktionsplan [ ]:= Änderungsplanung();
S24  end
S25  end
S26 end.

```

Abbildung 56: Algorithmus der Komponente Nutzungsgradoptimierung

5.2.3 Ein Beitrag zur Produktionsplanung und -steuerung unter Unsicherheit

Kapitel 5.2.3.1 leistet einen Beitrag dazu, Kapazitäten vorausschauend unter Unsicherheit zu planen. Die Erstellung eines Masterplans unter Unsicherheit wird in Kapitel 5.2.3.2 fokussiert. Eine Erweiterung bestehender Ansätze um die Möglichkeit Perioden zu gewichten, wird in Kapitel 5.2.3.3 entwickelt. Algorithmen werden in der in Kapitel 5.1.2.2.1 definierten Pseudocode Notation konzipiert.

5.2.3.1 Planung der Kapazitäten

Die Planung der Kapazitäten gründet sich auf den Strategien „Erstelle Lösungspopulationen“ (Strategie 1), „Plane und nutze die Kapazitäten“ (Strategie 3), „Vergrößere den Planungshorizont“ (Strategie 4) und „Plane nicht zu fest und halte Optionen offen“ (Strategie 5) aus Kapitel 5.2.1.

In Kapitel 5.2.3.1.1 wird der Lösungsraum begrenzt und in Kapitel 5.2.3.1.2 der Aufbau des Spielbaums konzipiert. Das Funktionsprinzip wird in Kapitel 5.2.3.1.3 entwickelt.

5.2.3.1.1 Lösungsraum

Betrachtungsgrundlage für die Planung der Kapazität ist der gesamte Lösungsraum des Problems. Der Lösungsraum ist dabei wie folgt aufgebaut: Je weiter von heute aus in die Zukunft geblickt wird, desto größer wird der Lösungsraum in der Tiefe. In der Breite wird der Lösungsraum durch Ereignisse, wie die Störung des Produktionsablaufs, vergrößert. In Abbildung 57 wird der Zusammenhang illustriert.

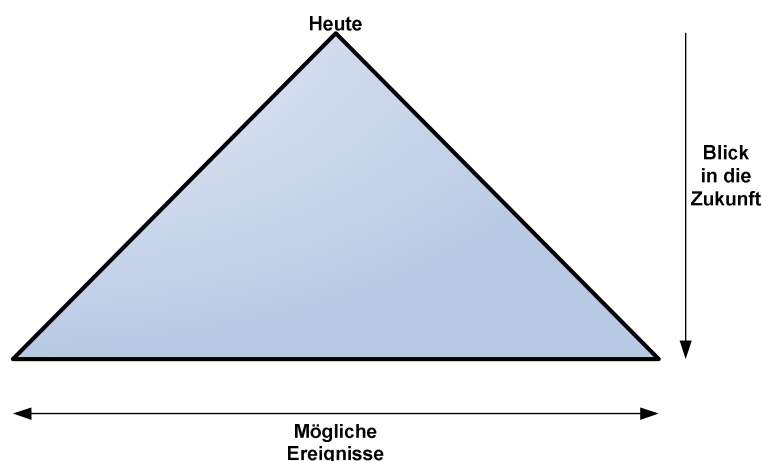


Abbildung 57: Lösungsraum des Planungsproblems

Die Analyse des Stands der Technik zeigt, dass es für eine gute Lösung nicht erforderlich ist, den gesamten Lösungsraum zu betrachten. Auf Grundlage dieser Einschätzung erfolgt eine Beschränkung des Lösungsraums in der Breite und in der Tiefe. Für die Planung der Kapazität wird dabei nur der beschränkte Lösungsraum betrachtet, wobei die Beschränkung in Breite und Tiefe abhängig von den zur Verfügung stehenden Planungsdaten und der Rechenleistung ist. Der Zusammenhang wird in Abbildung 58 veranschaulicht.

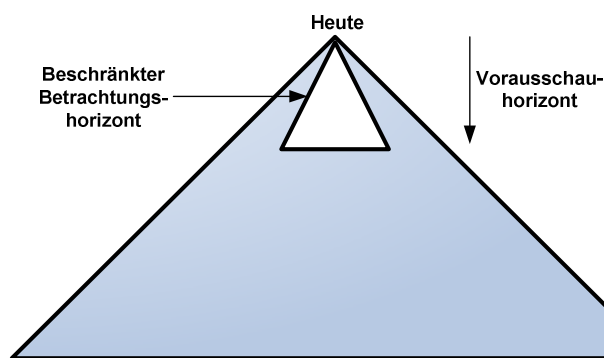


Abbildung 58: Beschränkter Lösungsraum des Planungsproblems

5.2.3.1.2 Spielbaum

Basierend auf einem rollierenden Planungshorizont, bei dem fortlaufend Produktionspläne erstellt werden, erfolgt die Erstellung eines Spielbaums. Ziel des Unternehmens ist es dabei, den Gewinn von T_0 bis T_e zu maximieren.³⁰⁵ Hierzu wird auf Grundlage der aktuellen Daten ein Spielbaum erstellt, der es ermöglicht, sich aus heutiger Sicht bestmöglich für einen langfristig optimalen Produktionsablauf aufzustellen. Aufgrund unvollständiger Informationen zur Entscheidungszeit muss dabei eine optimale Entscheidung zum Zeitpunkt T_i nicht notwendigerweise identisch sein mit der optimalen Entscheidung unter der Voraussetzung, dass die zukünftigen Realisierungen der Daten bekannt sind.

Die Realisierungszüge der Natur werden simuliert und das Unternehmen reagiert auf jeden dieser möglichen Züge. Anschließend erfolgt eine Bewertung der Kapazitätsniveaus, wobei eine optimale Entscheidung zum Zeitpunkt T_i nur im Erwartungswert optimal sein kann.

In Abbildung 59 wird der Aufbau des Spielbaums für die Planung der Kapazitäten unter Unsicherheit illustriert.

³⁰⁵ Vgl. hierzu die Klassifikation des Betrachtungshorizonts in Kapitel 5.1.1.1.1.

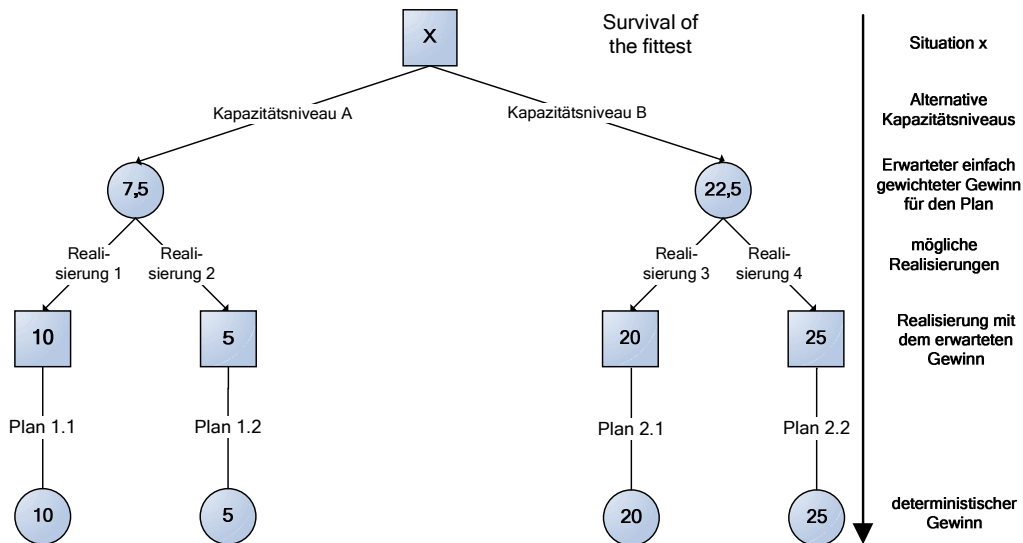


Abbildung 59: Spielbaum zur Planung der Kapazitäten unter Unsicherheit

5.2.3.1.3 Funktionsprinzip

Mit dem Ziel, die Kapazitäten so zu wählen, dass der Gewinn über alle Perioden optimiert wird und die Lieferfähigkeit gewährleistet ist, wird die Planung der Kapazitäten wöchentlich gestartet. Dazu werden für unterschiedliche Kapazitätsniveaus Produktionspläne erzeugt und anschließend sukzessive die Ausführung, der sich aus den Kapazitätsniveaus ergebenden Produktionspläne, simuliert. Ergebnis der Analyse ist die Festlegung der Kapazität für die übernächste Woche.

Die Komponente *Kapazitätsanalysealgorithmus* (vgl. Abbildung 60) ruft innerhalb des Programmablaufs das Programm *KapazitätsanalyseOptimierer()* auf. Der Algorithmus terminiert mit der Rückgabe der am besten bewerteten Kapazität für die übernächste Woche.

Program Kapazitätsanalysealgorithmus()

S₁ **begin**

/* Empfange die Vorausschautiefe. */

S₂ VorausschauTiefe := **Empfange**(VorausschauTiefe);

/* Setze die Zählervariable t die anzeigt in welcher Planungstiefe sich das Verfahren befindet. */

S₃ t := 0;

/* Empfange die Stamm- und Bewegungsdaten. */

S₄ **EmpfangePlanungsdaten**();

/* Starte die Analyse der Kapazität im Teil des Optimierers. */

S₅ Kapazität [] := **KapazitätsanalyseOptimierer**();

/* Rückgabe der für die übernächste Woche gewählten Kapazität. */

```
S6   return Kapazität [ ];
S7 end.
```

Abbildung 60: Algorithmus der Komponente Kapazitätsanalysealgorithmus

Das Programm *KapazitätsanalyseOptimierer()* wird von der Komponente *Kapazitätsanalysealgorithmus* gestartet und ruft das externe Komponentencluster *Planerstellung*³⁰⁶ auf, das für jedes betrachtete Szenario einen Produktionsplan erstellt.

Für ein gewähltes Kapazitätsniveau und einen dazugehörigen Produktionsplan wird mithilfe des Programms *RealisierungenNatur()* eine Bewertung ermittelt. Dazu wird dieses rekursiv aufgerufen und der Produktionsablauf simuliert. Dabei kann nach jedem Simulationsschritt der Produktionsplan durch eine Änderungsplanung angepasst werden. Das Programm *KapazitätsanalyseOptimierer()* reagiert auf die Realisierungen und startet dazu das Komponentencluster *Änderungsplanung*³⁰⁷.

In Abbildung 61 wird der Ablauf im Quellcode formalisiert.

```
Program KapazitätsanalyseOptimierer()
S1 begin
    /* Planungstiefe Null. */
S2   if (t = 0) then
S3     begin
S4       t := t + 1;
          /* Empfange wie viele alternative Kapazitäten analysiert werden sollen. */
S5       AnzahlKapazitätsVarianten := Empfange(AnzahlKapazitätsVarianten);
          /* Erstelle Kapazitätsvarianten. */
S6       for i := 1 to AnzahlKapazitätsVarianten do
S7         begin
          /* Erstelle Varianten von Kapazitätsniveaus. */
S8           Kapazität [ ] := FestlegungDerKapazität();
          /* Erzeuge eine eindeutige Identifikationsnummer für den Produktionsplan. */
S9           ErzeugeID();
          /* Erstelle einen Produktionsplan für das Kapazitätsniveau. */
S10          Produktionsplan [ ] := Planerstellung();
          /* Betrachte die möglichen Züge der Natur und ermittle den gewichteten
          Gewinn. */
S11          GewichteterGewinn(Produktionsplan [ ]) := RealisierungenNatur();
S12          i := i + 1;
S13        end
S14      end
S15    else if (t < VorausschauTiefe) then
```

³⁰⁶ Vgl. Kapitel 5.2.2.1

³⁰⁷ Vgl. Kapitel 5.2.2.2

```

S16 begin
S17   t := t + 1;
      /* Empfange wie viele alternative Pläne erstellt werden sollen. */
S18   AnzahlPlanVarianten := Empfange(AnzahlPlanVarianten);
      /* Erstelle u Produktionspläne. */
S19   for i := 1 to AnzahlPlanVarianten do
S20     begin
      /* Erzeuge eine eindeutige Identifikationsnummer für den Produktionsplan. */
S21       ErzeugeID();
      /* Erstelle einen Produktionsplan für das Kapazitätsniveau. */
S22       Produktionsplan [ ] := Änderungsplanung();
      /* Betrachte die möglichen Züge der Natur und ermittle den gewichteten
S23         Gewinn. */
S23       GewichteterGewinn(Produktionsplan [ ]) := RealisierungenNatur();
S24       i := i + 1;
S25     end
S26   end
      /* Abbruch der Vorausschau und Berechnung der erwarteten Kosten. */
S27   else then
S28     begin
      /* Wähle die beste Kapazität. */
S29       Kapazität [ ] := WähleKapazität();
      /* Rückgabe der gewählten Kapazität. */
S30       return Kapazität [ ];
S31     end
S32 end.

```

Abbildung 61: Algorithmus der Komponente Kapazitätsanalysealgorithmus - KapazitätsanalyseOptimierer

Das Programm *RealisierungenNatur()* wird rekursiv von dem Programm *KapazitätsanalyseOptimierer()* gestartet und ruft selbst wiederum das Programm *BerechneEinmalGewichteterGewinn()* auf. Dabei wird, um die Laufzeit zu verbessern, nur eine begrenzte Anzahl von Realisierungen betrachtet.

Eine Realisierung ist hierbei eine konkrete Ausprägung aus einer Menge von möglichen Ausprägungen, z.B. ein konkreter Kundenauftrag. Rückgabewert des Programms *RealisierungenNatur()* ist der einmal gewichtete Gewinn eines gewählten Kapazitätsniveaus. Alle Szenarien werden auf diese Weise analysiert.

In Abbildung 62 wird der Algorithmus konzipiert.

```

Function RealisierungenNatur()
S1 begin
S2   if (t <= Vorausschautiefe) then
S3     begin
      /*Empfange die Anzahl der möglichen Realisierungszüge. */
S4       v := Empfange(AnzahlRealisierungen);

```

```

S5      /* Erzeuge eine kleine Liste plausibler Realisierungszüge. */
      SamplingRealisierungen();
      /* Für alle Realisierungen. */
S6      for k := 1 to v do
S7      begin
      /* Erzeuge eine eindeutige Identifikationsnummer für die Realisierung. */
S8      ErzeugeID();
      /* Starte rekursiv das Programm KapazitätsanalyseOptimierer(). */
S9      Kapazität [ ] := KapazitätsanalyseOptimierer();
S10     k := k + 1;
S11     end
S12 end
      /* Abbruchkriterium. */
S13 else if (t > VorausschauTiefe) then
S14 begin
      /* Berechne den einmal gewichteten Gewinn für alle Kapazitätsniveaus einer
      Ebene. */
S15     EinmalGewichteterGewinn := BerechneEinmalGewichteterGewinn();
      /* Rückgabe des einmal gewichteten Gewinns. */
S16     return EinmalGewichteterGewinn;
S17 end
S18 end.

```

Abbildung 62: Algorithmus der Komponente Kapazitätsanalysealgorithmus - RealisierungenNatur

Die gewählte Kapazität wird durch das Programm *BerechneEinmalGewichteterGewinn()*, das von dem Programm *RealisierungenNatur()* gestartet wird, bewertet. Dabei werden der Gewinn und die Kosten berechnet und mit der Eintrittswahrscheinlichkeit gewichtet. Rückgabewert des Programms ist der einmal gewichtete Gewinn.

In Abbildung 63 wird die Funktionsweise veranschaulicht.

```

Program BerechneEinmalGewichteterGewinn()
S1 begin
      /* Unter der Bedingung, dass für ein Kapazitätsniveau Produktionspläne erstellt
      wurden und mehrere Szenarien auftreten können, berechne den gewichteten
      Gewinn eines Kapazitätsniveaus. */
S2 for all Realisierungen einer Ebene do
S3 begin
      /* Berechne die Kosten für eine Realisierung mit verringerten Opportunitäts-
      kosten für die angepasste Kapazität. */
S4     KostenRealisierung := BerechneKosten();
      /* Berechnen den Gewinn für eine Realisierung. */
S5     GewinnRealisierung := BerechneGewinn();
      /* Berechnen die Eintrittswahrscheinlichkeit für eine Realisierung. */
S6     WahrscheinlichkeitFürRealisierung := BerechneEintrittswahrscheinlichkeit();
      /* Berechne den Gesamtgewinn. */
S7     Gewinn := Gewinn + Eintrittswahrscheinlichkeit * GewinnRealisierung

```

```

/* Berechne die summierten Eintrittswahrscheinlichkeiten. */
S8   Eintrittswahrscheinlichkeiten := Eintrittswahrscheinlichkeiten +
      WahrscheinlichkeitFürRealisierung;
S9   end
/* Berechne den einmal gewichteten Gewinn. */
S10  EinmalGewichteterGewinn := Gewinn/Eintrittswahrscheinlichkeiten;
/* Rückgabe des einmal gewichteten Gewinns eines Kapazitätsniveaus. */
S11  return EinmalGewichteterGewinn;
S12  end.

```

**Abbildung 63: Algorithmus der Komponente Kapazitätsanalysealgorithmus -
BerechneEinmalGewichteterGewinn**

Das Programm *WähleKapazität()* wird von dem Programm *KapazitätsanalyseOptimierer()* gestartet und ermittelt aus allen potentiellen Kapazitätsniveaus, das mit der besten Bewertung und gibt dieses zurück.

In Abbildung 64 wird der Ablauf formalisiert.

```

Program WähleKapazität()
S1  begin
      /* Indikatorvariable für den Produktionsplan. */
S2  j := 1;
S3  for all Produktionspläne do
S4  begin
      /* Suche die Kapazität mit dem höchsten gewichteten Gewinn. */
S5  if(EinmalGewichteterGewinn[Produktionsplan[j]] > EinmalGewichteterGewinn[
      Produktionsplan[j+1]]) then
S6  begin
      /* Speichere die gewählte Kapazität. */
S7  Kapazität [ ]:= ErmittleKapazität(Produktionsplan[j+1]);
S8  j := j + 1;
S9  end
S10 else then
S11 begin
      /* Erhöhe die Laufvariable. */
S12 j := j + 1;
S13 end
S14 end
      /* Rückgabe der gewählten Kapazität. */
S15 return Kapazität [ ];
S16 end.

```

**Abbildung 64: Algorithmus der Komponente Kapazitätsanalysealgorithmus -
WähleKapazität**

5.2.3.2 Erstellung eines Masterplans

Die Erstellung eines Masterplans unter Unsicherheit basiert auf den Strategien „Erstelle Lösungspopulationen“ (Strategie 1), „Vergrößere den Planungshorizont“ (Strategie 4), „Plane nicht zu fest und halte Optionen offen“ (Strategie 5), „Erstelle einen Masterplan und beachte alle Restriktionen“ (Strategie 7), „Gewichte die Perioden und verwende variable Auftragspakete“ (Strategie 8) und „Parallelisiere Prozesse und berechne den Mindestnutzen eines Plans“ (Strategie 14).³⁰⁸

Der Lösungsraum wird in Kapitel 5.2.3.2.1 abgegrenzt und der Spielbaum in Kapitel 5.2.3.2.2 dargestellt. In Kapitel 5.2.3.2.3 wird das Funktionsprinzip konzipiert.

5.2.3.2.1 Lösungsraum

Der betrachtete Lösungsraum ist identisch mit der Eingrenzung des Lösungsraums in Kapitel 5.2.3.1.1 und wird an dieser Stelle nicht erneut definiert.

5.2.3.2.2 Spielbaum

Im Gegensatz zu dem in Kapitel 5.2.3.1.2 entwickelten Spielbaum werden hier keine alternativen Kapazitätsniveaus fokussiert, sondern alternative Produktionspläne analysiert. Des Weiteren wird der Gewinn nicht nur aufgrund der erwarteten Eintrittswahrscheinlichkeit gewichtet, sondern darüber hinausgehend die Gewichtung der betrachteten Periode³⁰⁹ zu Grunde gelegt.

Abbildung 65 illustriert den Aufbau des Spielbaums für die Erstellung eines Masterplans unter Unsicherheit.

³⁰⁸ Vgl. Kapitel 5.2.1

³⁰⁹ Vgl. Kapitel 5.2.3.3

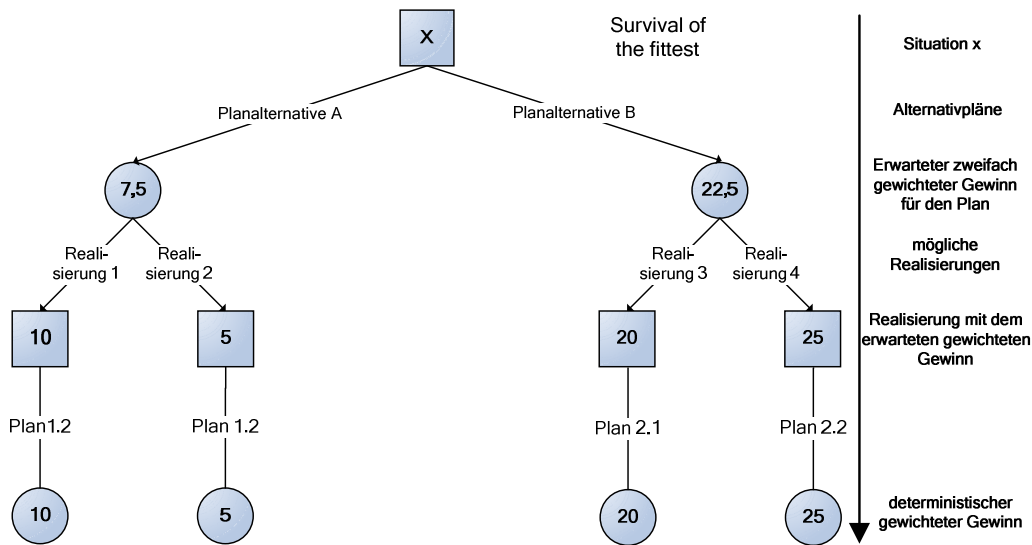


Abbildung 65: Spielbaum zur Erstellung eines Masterplans unter Unsicherheit

5.2.3.2.3 Funktionsprinzip

Die Erstellung eines Masterplans unter Unsicherheit wird immer wöchentlich und im Bedarfsfall, z.B. bei einer besonders großen Bedarfserhöhung, gestartet. Ziel ist es, einen Produktionsplan zu finden, der zu einem hohen Gesamtgewinn über alle Perioden führt und die Lieferfähigkeit verbessert. Dazu werden für das festgelegte Kapazitätsniveau mehrere Produktionspläne erzeugt und die Ausführung des Produktionsplans sukzessive simuliert, wobei nach jedem Simulationsschritt der Produktionsplan angepasst werden kann.

Für einen Produktionsplan werden erst die voraussichtlich entstehenden Kosten anhand der Zielfunktion berechnet und anschließend der zweifach gewichtete Gewinn. Dieser wird erstens anhand der Eintrittswahrscheinlichkeiten und zweitens anhand der Periodengewichtung berechnet. Gewinne für Vorleistungen werden anteilig angerechnet, wobei Pufferaufträge nur mit Herstellkosten eingehen.

Es wird einmalig eine Auswahl an Szenarien berechnet, für die ein Produktionsplan erstellt wird. Für den Kundenbedarf wird der Erwartungswert gebildet und aufgerundet. Für die Produktionsmenge wird der Erwartungswert gebildet und abgerundet. Anschließend werden die Produktionspläne verglichen und ein Masterplan ausgewählt, der auch Optionen für zukünftige Änderungen offen hält. Für den gewählten Masterplan wird der Mindestnutzen durch eine Worst-Case-Analyse berechnet.

Die Komponente *Plananalysealgorithmus* (vgl. Abbildung 66) startet die Komponenten *Periodengewichtung*³¹⁰ und *Nutzungsgradoptimierung*³¹¹, sowie das Programm

³¹⁰ Vgl. Kapitel 5.2.3.3

PlananalyseOptimierer(). Der Algorithmus terminiert mit der Rückgabe des Masterplans und des Mindestnutzens.

```

Program Plananalysealgorithmus()
S1 begin
    /* Setze die Zählervariable t, die anzeigt in welcher Planungstiefe sich das
    Verfahren befindet. */
S2 t := 0;
    /* Empfange die Anzahl der Perioden, die bei der Planerstellung beachtet werden
    sollen. */
S3 VorausschauTiefe := Empfange(VorausschauTiefe);
    /* Empfange die Planwoche in der die Kapazität angepasst wird. */
S4 PlanWoche := Empfange(PlanWoche);
    /* Empfange die Stamm- und Bewegungsdaten. */
S5 EmpfangePlanungsdaten();
    /* Berechne die Periodengewichtung. */
S6 Periodengewichtung [ ] := Periodengewichtung();
    /* Starte die Erstellung eines Produktionsplans im Teil des Optimierers. */
S7 Produktionsplan [ ] := PlananalyseOptimierer();
    /* Starte die Komponente Nutzungsgradoptimierung. */
S8 Masterplan [ ] := Nutzungsgradoptimierung();
    /* Starte die Worst-Case Analyse für den Masterplan. */
S9 Mindestnutzen := WorstCaseAnalyse();
    /* Rückgabe des gewählten Masterplans und des Mindestnutzens. */
S10 return Masterplan [ ], Mindestnutzen;
S11 end.

```

Abbildung 66: Algorithmus des Komponentenclusters Plananalysealgorithmus

Das Programm *PlananalyseOptimierer()* wird von der Komponente *Plananalysealgorithmus* gestartet und ruft das Komponentencluster *Planerstellung*³¹² auf, das für die unterschiedlichen Szenarios Produktionspläne berechnet. Der Produktionsablauf wird dabei durch das Programm *RealisierungenNatur()* simuliert. Das Programm *PlananalyseOptimierer()* reagiert auf die Realisierungen und startet das Komponentencluster *Änderungsplanung*³¹³.

In Abbildung 67 wird der Algorithmus in Pseudocode formalisiert.

³¹¹ Vgl. Kapitel 5.2.2.6

³¹² Vgl. Kapitel 5.2.2.1

³¹³ Vgl. Kapitel 5.2.2.2

```

Program PlananalyseOptimierer()
S1 begin
    /* Überprüfung der Planungstiefe. */
S2 if (t = 0) then
S3   begin
S4     t := t + 1;
    /* Empfange wie viel alternative Pläne erstellt werden sollen. */
S5   AnzahlPlanVarianten := Empfange(AnzahlPlanVarianten);
    /* Erstelle u Produktionspläne. */
S6   for i := 1 to AnzahlPlanVarianten do
S7     begin
    /* Erzeuge eine eindeutige Identifikationsnummer für den Produktionsplan. */
S8       ErzeugeID();
S9       Produktionsplan [ ] := Planerstellung();
    /* Betrachte die möglichen Züge der Natur und ermittle den gewichteten
S10        Gewinn. */
S10      GewichteterGewinn(Produktionsplan [ ]) := RealisierungenNatur();
S11      i := i + 1;
S12     end
S13   end
    /* Überprüfe die Planungstiefe. */
S14   else if (t < VorausschauTiefe) then
S15     begin
    /* Erstelle u Produktionspläne. */
S16     for i := 1 to AnzahlPlanVarianten do
S17       begin
    /* Erzeuge eine eindeutige Identifikationsnummer für den Produktionsplan. */
S18         ErzeugeID();
    /* Starte die Komponente Änderungsplanung. */
S19         Produktionsplan [ ] := Änderungsplanung();
    /* Betrachte die möglichen Züge der Natur und ermittle den gewichteten
S20        Gewinn. */
S20        GewichteterGewinn(Produktionsplan [ ]) := RealisierungenNatur();
S21        i := i + 1;
S22       end
S23     end
    /* Abbruch der Vorausschau und Berechnung der erwarteten Kosten. */
S24     else if (t => VorausschauTiefe) then
S25       begin
    /* Ermittle den besten Produktionsplan. */
S26       Produktionsplan [ ] := WählePlan();
    /* Rückgabe des gewählten Produktionsplans. */
S27       return Produktionsplan [ ];
S28     end
S29   end.

```

Abbildung 67: Algorithmus der Komponente Plananalysealgorithmus - PlananalyseOptimierer

Das Programm *RealisierungenNatur()* wird rekursiv von dem Programm *PlananalyseOptimierer()* gestartet und ruft selbst das Programm *BerechneZweifach-*

GewichteterGewinn() auf. Dabei wird eine begrenzte Anzahl von Realisierungen betrachtet, um die Laufzeit zu verbessern.

Eine einzelne Realisierung ist eine konkrete Ausprägung aus einer Menge von potentiellen Ausprägungen, die jeweils eine bestimmte Wahrscheinlichkeit besitzen. Die Realisierungen werden durch die Funktion *SamplingRealisierungen()* bestimmt. Rückgabewert des Programms *RealisierungenNatur()* ist der zweifach gewichtete Gewinn eines Produktionsplans. Alle berechneten Produktionspläne werden so analysiert.

In Abbildung 68 wird der Algorithmus formalisiert.

```

Program RealisierungenNatur()
S1 begin
S2   if (t <= Vorausschautiefe) then
S3   begin
S4     /*Empfange die Anzahl der möglichen Realisierungszüge. */
S4     v := Empfange(AnzahlRealisierungen);
S4     /* Erzeuge eine kleine Liste von plausiblen Realisierungszügen. */
S5     SamplingRealisierungen();
S5     /* Für alle Realisierungen erstelle Produktionspläne. */
S6     for k := 1 to v do
S7       begin
S8         /* Erzeuge eine eindeutige Identifikationsnummer für die Realisierung. */
S8         RealisierungID := ErzeugeID();
S8         /* Starte das Programm PlananalyseOptimierer() und reagiere auf die
S9         Realisierungen. */
S9         Produktionsplan [ ] := PlananalyseOptimierer();
S10        k := k + 1;
S11       end
S12     end
S13   else if (t > VorausschauTiefe) then
S14     begin
S15       /* Berechne den zweifach gewichteten Gewinn für alle Produktionspläne einer
S15       Ebene. */
S15       ZweifachGewichteterGewinn := BerechneZweifachGewichteterGewinn();
S15       /* Rückgabe des zweifach gewichteten Gewinns. */
S16     return ZweifachGewichteterGewinn;
S17   end
S18 end.

```

Abbildung 68: Algorithmus der Komponente Plananalysealgorithmus - RealisierungenNatur

Ein ausgeführter Produktionsplan wird durch das Programm *BerechneEinmalGewichteterGewinn()* bewertet, das von dem Programm *RealisierungenNatur()* gestartet wird. Dabei werden Gewinn und Kosten berechnet und mit der Eintrittswahrscheinlichkeit

und der Periodengewichtung gewichtet. Der Rückgabewert des Programms ist damit der zweifach gewichtete Gewinn.

In Abbildung 69 wird der Algorithmus entwickelt.

```

Program BerechneZweifachGewichteterGewinn()
S1 begin
    /* Unter der Bedingung, dass ein Produktionsplan ausgeführt wurde, berechne
    den zweifach gewichteten Gewinn eines Plans. */
S2 for all Realisierungen einer Ebene do
S3 begin
    /* Berechnen die Kosten für eine Realisierung mit den verringerten Opportuni-
    tätskosten für die angepasste Kapazität. */
S4 KostenRealisierung := BerechneKosten();
    /* Berechnen den Gewinn für eine Realisierung. */
S5 GewinnRealisierung := BerechneGewinn();
    /* Beachte die Periodengewichtung. */
S6 GewichteterGewinnRealisierung := BerechneGewichteterGewinn();
    /* Berechne die Eintrittswahrscheinlichkeit für eine Realisierung. */
S7 WahrscheinlichkeitFürRealisierung := BerechneEintrittswahrscheinlichkeit();
    /* Berechne den Gesamtgewinn. */
S8 Gewinn := Gewinn + Eintrittswahrscheinlichkeit * GewichteterGewinnRealisie-
    rung
    /* Berechne die summierten Eintrittswahrscheinlichkeiten. */
S9 Eintrittswahrscheinlichkeiten := Eintrittswahrscheinlichkeiten +
    WahrscheinlichkeitFürRealisierung;
S10 end
    /* Berechne den zweifach gewichteten Gewinn. */
S11 ZweifachGewichteterGewinn := Gewinn/Eintrittswahrscheinlichkeiten;
    /* Rückgabe des zweifach gewichteten Gewinns. */
S12 return ZweifachGewichteterGewinn;
S13 end.

```

Abbildung 69: Algorithmus der Komponente Plananalysealgorithmus - BerechneZweifachGewichteterGewinn

Das Programm *WählePlan()* wird von dem Programm *PlananalyseOptimierer()* gestartet und ermittelt aus allen gültigen Produktionsplänen den Produktionsplan mit der besten Bewertung.

In Abbildung 70 wird der Algorithmus formalisiert.

```

Program WählePlan()
S1 begin
    /* Wähle den Produktionsplan mit dem höchsten gewichteten Gewinn. */
S2 for all Produktionspläne do
S3   begin
S4     if(ZweifachGewichteterGewinn[Produktionsplan[j]] > ZweifachGewichteter-
        Gewinn[Produktionsplan[j+1]] then
S5       begin
            /* Speichere den besten Produktionsplan. */
S6         Produktionsplan [ ] := Produktionsplan[j+1] ;
            /* Erhöhe die Laufvariable. */
S7         j := j + 1;
S8       end
S9     else then
S10      begin
S11        j := j + 1;
S12      end
S13    end
        /* Rückgabe des gewählten Produktionsplans. */
S14    return Produktionsplan [ ];
S15 end.

```

Abbildung 70: Algorithmus der Komponente Plananalysealgorithmus - WählePlan

5.2.3.3 Gewichtung der Perioden

Die Gewichtung der Perioden beruht auf den Strategien „Vergrößere den Planungshorizont“ (Strategie 4), „Plane nicht zu fest und halte Optionen offen“ (Strategie 5) und „Gewichte die Perioden und verwende variable Auftragspakete“ (Strategie 8).³¹⁴

Ein Produktionsplaner bewegt sich sukzessive durch den Planungshorizont und die Gesamtkosten des Produktionsablaufs werden immer durch die verursachten Ist-Kosten erhöht. Um den Sachverhalt zu verdeutlichen, ist es sinnvoll zwei extreme Situationen zu fokussieren, die im Folgenden analysiert werden.

Der erste Extrempunkt besteht darin, immer nur die Kosten die jetzt anfallen, zu minimieren. Werden keine Verzögerungs- und Ausfallkosten berücksichtigt, wird vermutlich überhaupt nicht produziert; selbst mit Berücksichtigung von Verzögerungs- und Ausfallkosten wird die Zukunft vernachlässigt.

Der zweite Extrempunkt besteht darin, die Produktion über den gesamten Lebenszeitraum des Unternehmens zu optimieren und jede Periode gleichbedeutend einzubeziehen. Hierbei besteht das Problem darin, dass sich viele Einflussfaktoren gegenüber der ursprünglichen Planung laufend ändern. Somit wird ein vorher optimaler

³¹⁴ Vgl. Kapitel 5.2.1

Produktionsplan möglicherweise suboptimal oder sogar ungültig und der erzielte Planungsablauf potentiell schlechter, als wenn von Anfang an immer tageweise gesteuert worden wäre. Deshalb sollen die Kosten für eine Periode in der Zukunft weniger stark berücksichtigt werden, weil sich bis dahin ggf. noch viel ändert und der Produktionsplan damit vermutlich ohnehin hinfällig wird.

Im Folgenden werden Kombinationen aus beiden Strategien entwickelt, die eine spezifische nicht konstante Gewichtung der einzelnen Perioden ermöglichen, wobei durch entsprechende Wahl der Parameter dabei auch die beiden Extremsituationen erzeugt werden können.

Die linear fallende Funktion $f(x) = T - b \cdot x$ beschreibt eine stetige Bewertung in jeder Periode. Somit ergibt sich eine geringere Bewertung für die späteren Perioden (vgl. Abbildung 71).

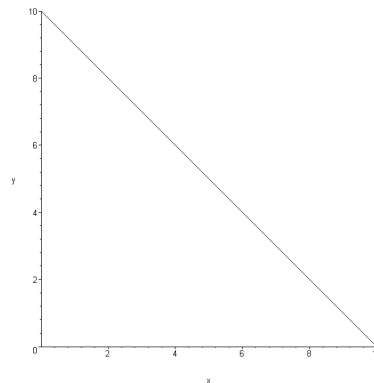


Abbildung 71: Periodengewichtung - Linear fallende Funktion

Die invertierte Funktion $f(x) = \frac{1}{b \cdot x}$ ermöglicht eine schnelle Reduzierung der Plankosten (vgl. Abbildung 72).

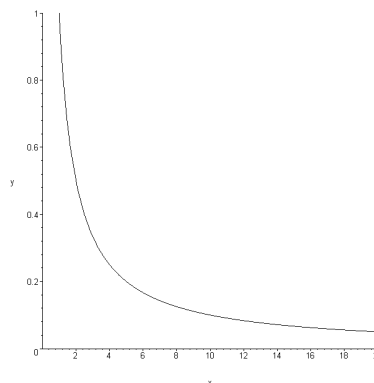


Abbildung 72: Periodengewichtung - Invertierte Funktion

Bei der exponentiellen Abnahme $f(x) = a^{-x}$ ändert sich der Funktionswert in gleich großen Intervallen jeweils um den gleichen Faktor a (vgl. Abbildung 73).

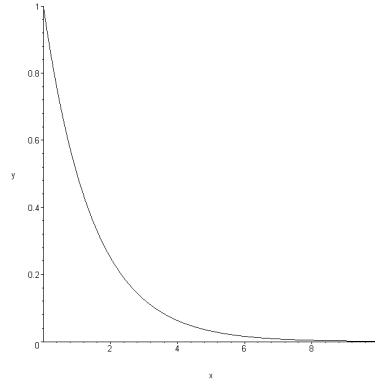


Abbildung 73: Periodengewichtung - Exponentielle Abnahme

Die Exponentialfunktion mit der Basis e $f(x) = e^{-\lambda x}$ ist ein Sonderfall der Exponentialfunktionen durch die oftmals auch ein Zerfallsprozess beschrieben wird, wobei λ die Zerfallgeschwindigkeit definiert (vgl. Abbildung 74).

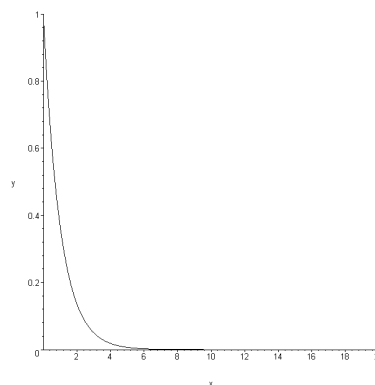


Abbildung 74: Periodengewichtung - Exponentialfunktion

Die Komponente *Periodengewichtung* (vgl. Abbildung 75) wird von der Komponente *Plananalysealgorithmus*³¹⁵ gestartet und ermittelt die spezifische Gewichtung für die betrachteten Perioden.

³¹⁵ Vgl. Kapitel 5.2.3.2

Program Periodengewichtung()

```
S1 begin  
S2   for all Perioden do  
S3   begin  
      /* Wähle aus der Funktionsauswahl die Gewichtungsfunktion. */  
S4   WähleFunktion();  
      /* Gewichte die Perioden mit dem sich aus der Gewichtungsfunktion  
      ergebenden Gewichtungsfaktor. */  
S5   Periodengewichtung [ ] := GewichtePerioden();  
      /* Rückgabe der Periodengewichtung. */  
S6   return Periodengewichtung [ ];  
S7   end  
S8 end.
```

Abbildung 75: Algorithmus der Komponente Periodengewichtung

6 Fazit und Ausblick

„(Ver)suche nicht in die Fußstapfen der weisen Männer zu treten, suche was diese suchten.“

Matson Bashô (*1644)

Erstes Ziel dieser Arbeit war es, eine Möglichkeit zu finden, wie unterschiedliche Planungsverfahren für verschiedene Planungsprobleme objektiv verglichen werden können. Dazu wurden Planungsproblem, Simulation des Produktionsablaufs und Planungsverfahren voneinander entkoppelt. Es wurde ein adaptierbares stochastisches Optimierungsmodell entwickelt, das speziell das betrachtete Planungsproblem abbildet und zugleich erweiterbar für weitere Problemstellungen ist. Weiterhin wurde eine Benchmarkplattform entwickelt, die es ermöglicht, Planungsverfahren in einer simulierten Produktion zu vergleichen und eine Schnittstelle für Planungsverfahren schafft.

Zweites Ziel dieser Arbeit war es, das betrachtete Produktionssystem bestmöglich aufzustellen. Dazu wurden im Stand der Technik analysierte Planungsstrategien weiterentwickelt und so die Grundlage für die Entwicklung neuer Planungsansätze geschaffen, um einen Beitrag zur PPS und zur PPS unter Unsicherheit zu leisten.

Aller Voraussicht nach werden die Anforderungen an die Produktion weiter wachsen und in der Summe die Produktionsplanung für enorm viele Produktvarianten immer komplexer. Simultan wird die Planungsunsicherheit aufgrund der Globalisierung und komplexer weltweit verteilter Produktionsnetzwerke weiter steigen. Dabei sollten in Anbetracht des mittlerweile wenig umstrittenen Zusammenhangs zwischen Klimawandel und industrieller Produktion bei der Produktionsplanung zukünftig ökologische Aspekte betrachtet werden. Dabei kann die Planerstellung auf kurze Transportwege, eine hohe Auslastung, eine nachhaltige Ressourcenverwendung und eine effiziente Entsorgungslogistik abzielen.

Schnell wachsende Märkte in Indien und China bieten ein gewaltiges Absatzpotential für die europäische Automobilzulieferindustrie. Dieses Potential gilt es, im Hinblick auf eine von dem Wachstum ausgehende enorme Bedrohung für die Umwelt mit dem nötigen Weitblick zu nutzen.

Darüber hinaus könnte ein Produktionssystem betrachtet werden, in dem Mitarbeiter und Maschinen vollständig geleast sind. Mit zunehmender Automatisierung in der Automobilzulieferindustrie sind dabei flexible Roboter vorstellbar, die in verschiedenen Werken einsetzbar sind und von einer Leiharbeitsfirma bereit gestellt werden.

7 Literaturverzeichnis

- [ASN02] Abbass, H. A.; Sarker, R. A.; Newton, C. S.: Data mining: a heuristic approach, Idea Group Publishing, London, 2002.
- [Alt06] Altemeier, S.: Analyse und Klassifizierung von deterministischen Losgrößenplanungsmodellen, Diplomarbeit Universität Paderborn, Wirtschaftsinformatik insbes. CIM, Paderborn, 2006.
- [And29] Andler, K.: Rationalisierung der Fabrikation und optimale Losgröße, München, R. Oldenbourg, 1929.
- [BB01] Bartsch, H.; Bickenbach, P.: Supply Chain Management mit SAP APO, Supply-Chain-Modelle mit dem Advanced Planner & Optimizer 3.1, SAP Press, 2001.
- [Bet06] Betge, D.: Produktion und Logistik, Koordination in Advanced Planning and Scheduling-Systemen, Gabler Edition Wissenschaft, Wiesbaden, 2006.
- [BGR94] Bichler, K.; Gerster, W.; Reuter, R.: Logistik-Controlling mit Benchmarking, Praxisbeispiele aus Industrie und Handel, Gabler, Wiesbaden, 1994.
- [BKP05] Bichler, K.; Krohn, R.; Philippi, P.: Gabler Kompakt-Lexikon, Logistik, Gabler, Wiesbaden, 2005.
- [BL06] Balla, J.; Layer, F.: Produktionsplanung mit SAP APO-PP/DS, Galileo Press, Mai 2006.
- [Bon98] Bongaerts, L.: Integration of scheduling and control in holonic manufacturing systems, KU Leuven, 1998.
- [BP79] Baker, K. R.; Peterson, D. W.: An Analytic Framework for Evaluating Rolling Schedules, Management Science, 25, 1979.
- [Bus04] Busch, A.: Kollaborative Änderungsplanung in Unternehmensnetzwerken der Serienfertigung, HNI-Verlagsschriftenreihe, Band 148, Paderborn, 2004.
- [CDM+02] Calhoun, K. M.; Deckro, R. F.; Moore, J. T.; Chrissis, J. W.; Van Hove, J. C.: Planning and re-planning in project and production scheduling, Omega, Band 30, 2002.
- [CJK79] Carlson, R.; Jucker, J.; Kropp, D.: Less nervous MRP systems: A dynamic economic lot-sizing approach, Management Science 25, 1979.
- [Cor98] Corsten, H.: Produktionswirtschaft, 7. Auflage, Oldenbourg Verlag, 1998.

-
- [Dan03] Dangelmaier, W.: Produktion und Information, System und Modell, Springer, Berlin, Heidelberg, New York, 2003.
- [Dan99] Dangelmaier, W.: Fertigungsplanung, Planung von Aufbau und Ablauf der Fertigung, Springer, Berlin, Heidelberg, New York, 1999.
- [Dem68] DeMatteis, J. J.: An economic lot-sizing technique I – the part-period algorithm, IBM Systems Journal 7(1), 1968.
- [DH95] Drexl, A.; Haase, K.: Proportional Lotsizing and Scheduling, International Journal of Production Economics, Band 40, 1995.
- [DHH+97] Dangelmaier, W.; Henkel, S.; Holtkamp, R.; Langemann, T.; Schallner, H.: Engpaßorientiertes Verfahren für eine verteilte PPS in einem Multi-Agenten-Ansatz, GI-Softwaretechnik-Trends Band 17, Heft 3, Workshop „Arbeitsplatzrechner-Integration zur Prozessverbesserung“ im Rahmen der 27. Jahrestagung der Gesellschaft für Informatik. Gesellschaft für Informatik, Aachen, September 1997.
- [DHH+97a] Dangelmaier, W.; Henkel, S.; Holtkamp, R.; Langemann, T.; Schallner, H.: Ein Multi-Agenten-Ansatz für die verteilte PPS, Anwendung des engpassorientierten Verfahrens zur Auftragseinplanung, wt Produktion und Management 87, 1997.
- [DHL+97] Dangelmaier, W.; Holtkamp, R.; Langemann, T.; Sobiech, H.-J.; Ziersch, F.: Modeling of Production Planning and Control for the Integration of Information Systems, In: Sen, A. et al (Eds.): 4th International Conference on Computer Integrated Manufacturing (ICCIM) '97, Volume 1, Singapore, Springer, Berlin, Heidelberg, New York, 1997.
- [DKL04] Donninger, C.; Kure, A.; Lorenz., U.: Parallel Brutus: The first distributed, accelerated chess program, In: Proc. of 18th International Parallel & Distributed Processing Symposium (IPDPS), 2004.
- [DLM+98] Dangelmaier, W.; Langemann, T.; Maßberg, W.; Schallner, H.: An agent based approach for distributed production planning and control, In: Proceedings of CIRP-ICME'98, Teti, R. (Ed.), CIRP, Italien, 1998.
- [DSV97] Domschke, W.; Scholl, A.; Voss, S.: Produktionsplanung, Berlin, Springer, Berlin, Heidelberg, New York, 1997.
- [DV83] Debot, M.; Van Wassenhove, L.: Cost increases due to demand uncertainty in MRP lot sizing aus Decision Sciences 14, 1983.
- [DW97] Dangelmaier, W.; Warnecke, H. J.: Fertigungslenkung, Springer, Berlin, Heidelberg, New York, 1997.

-
- [EGH+03] Ehrhoff, J.; Grothklags, S.; Halbsgut, J.; Lorenz, U.; Sauerwald, T.: The repair game: Robust plans and disturbance management in aircraft scheduling by the help of game tree search, In: Proc. of the 43rd Annual Symposium of the Airline Group of the International Federation of Operational Research Societies (AGIFORS), 2003.
- [EGL04] Ehrhoff, J.; Grothklags S.; Lorenz, U.: Störungsmanagement und Planung unter Unsicherheiten, angewendet auf die Flugplanung, GOR: Entscheidungsunterstützende Systeme in Supply Chain Management und Logistik, Physica Verlag, 2004.
- [EGL05] Ehrhoff, J.; Grothklags, S.; Lorenz, U.: Das Reparaturspiel als Formalisierung von Planung unter Zufallseinflüssen, angewendet in der Flugplanung. In Entscheidungsunterstützende Systeme in Supply Chain Management und Logistik, Physika-Verlag, 2005.
- [FTD07] Financial Times Deutschland: Kollege Roboter, 22. Januar 2007, Auflage 128244, 2007.
- [FL03] Framinan, J. M.; Leisten, R.: An effecient constructive heuristic for flowtime minimisation in permutation flow shop, Omega, 31, 2003.
- [Fle90] Fleischmann, B.: The Discrete Lot-Sizing and Scheduling Problem, European Journal of Operational Research, Vol. 44, 1990.
- [GC87] Goldratt, E.M.; Cox, J.: Das Ziel, Höchstleistung in der Fertigung, Hamburg, 1987.
- [GKK04] Gerdes, I.; Klawonn, F.; Kruse, R.: Genetische Algorithmen – Strategien und Optimierungsverfahren- Beispielanwendungen, Vieweg-Verlag, 1. Auflage, Wiesbaden, 2004.
- [GLW+03] Gautam, D.; Lassen, S.; Wienecke, K.; Lücke, T.: Markspiegel Business Software - ERP/PPS, Hrsg.: Luczak, H.; Stich, V., 2. Auflage, Aachen, 2003.
- [GMW07] Guojun, G.; Chaoqun, M.; Jianhong, W.: Data Clustering, Theory, Algorithms and Applications, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.
- [Gol88] Goldratt, M. E.: Computerized shop floor scheduling, In: International Journal of Production Research, 26, 1988.
- [Gre88] Greiner, T.: Ein Algorithmus zur kapazitätsorientierten Bildung von Losen, Springer, Berlin, Heidelberg, New York, 1988.

-
- [Gro79] Groff, G.: A lot sizing rule for time-phased component demand, *Production and Inventory Management* 20(1), South-Western College Pub., 1979.
- [GT04] Günther, H.-O.; Tempelmeier, H.: *Produktion und Logistik*, Springer, Berlin, Heidelberg, New York, 2004.
- [Hac84] Hackstein, R.: *Produktionsplanung und -steuerung (PPS): Ein Handbuch für die Betriebspraxis*, VDI-Verlag, Düsseldorf, 1984.
- [HBH88] Horvitz, E.; Breese, J.; Henrion, M.: Decision theory in expert systems and artificial intelligence, *Journal of Approximate Reasoning*, Special Issue on Uncertainty in Artificial Intelligence, 1988.
- [HC96] Ho, C.-J.; Carter, P.: An investigation of alternative dampening procedures to cope with MRP system nervousness, *International Journal of Production Research* 34, 1996.
- [HCM+86] Ho, C.-J.; Carter, P.; Melnyk, S.; Narasimhan, R.: Quantity versus timing change in open order: A critical evaluation, *Production and Inventory Management* 27, 1986.
- [Hei06] Heidenreich, J.: *Adaptierbare Änderungsplanung für Mengen und Kapazitäten in Produktionsnetzwerken der Serienfertigung*, HNI-Verlagsschriftenreihe, Band 182, Heinz Nixdorf Institut, Paderborn, 2006.
- [HGH02] Heuser, R.; Günther, F.; Hatzfeld, O.: *Integrierte Planung mit SAP, Konzeption, Methodik, Vorgehen*, Galileo Press, 2002.
- [Hin75] Hinrichsen, J.: Branch and Bound Verfahren zur Lösung des Rundreiseproblems, *Verkehrswissenschaftliche Studien* 28, Vandenhoeck & Ruprecht in Göttingen, Hamburg, 1975.
- [HLR95] Ho, C.-J.; Law, W.-K.; Rampal, R.: Uncertainty-dampening methods for reducing MRP system nervousness, *International Journal of Production Research* 33, 1995.
- [Hro04] Hromkovič, J.: *Algorithmics for Hard Problems*, Springer, Berlin, Heidelberg, New York, 1998.
- [Ho89] Ho, C.-J.: Evaluating the impact of operating environments on MRP system nervousness, *International Journal of Production Research* 27, 1989.
- [Hop05] Hoppe, M.: *Bestandsoptimierung mit SAP*, SAP PRESS, Bonn, 2005.

-
- [HS96] Hidle, J. L. und Sen, S.: Stochastic Decomposition, Kluwer Academic Publishers, 1996.
- [KD97] Kimms, A.; Drexl, A.: Proportional Lot Sizing and Scheduling: Some Extensions, John Wiley & Sons, Inc. Networks 32, 1998.
- [Ker93] Kernler, H.: PPS der 3. Generation - Grundlagen, Methoden, Anregungen, Hüthig Buch Verlag GmbH, Heidelberg, 1993.
- [Kit82] Kittel, T.: Produktionsplanung und -steuerung in Klein- und Mittelbetrieb - Chancen und Risiken des EDV-Einsatzes, Expert Verlag, Grafenau, 1982.
- [KKK87] Karmarkar, U.S.; Kekre, S.; Kekre, S.: The Deterministic Lotsizing Problem with Startup and Reservation Costs, Operations Research, Vol. 35, 1987.
- [KM75] Knuth, D.E.; Moore, R.W.: An analysis of alpha beta pruning, Artificial Intelligence, 1975.
- [KMZ00] Knolmayer, G.; Mertens, P.; Zeier, A.: Supply Chain Management auf Basis von SAP-Systemen - Perspektiven der Auftragsabwicklung für Industriebetriebe, Berlin, Heidelberg, New York, 2000.
- [KSH01] Kleywegt, A. J.; Shapiro, A.; Homem-De-Mello, T.: The sample average approximation method for stochastic discrete optimization, SIAM Journal of Optimization, 2001.
- [Kur98] Kurbel, K.: Produktionsplanung und -steuerung: methodische Grundlagen von PPS-Systemen und Erweiterungen, 3. Aufl., Oldenbourg Verlag, München, 1998.
- [LA86] Lee, T. S.; Adam, E.: Forecasting error evaluation in material requirements planning MRP production-inventory systems aus Management Science 32, Providence R.I., 1986.
- [Leg99] Legner, C.: Benchmarking informationssystemgestützter Geschäftsprozesse, Gabler, Wiesbaden, 1999.
- [Lor00] Lorenz, U.: Controlled Conspiracy-2-Search, In Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS), 2000.
- [Lö05] Lödding, H.: Verfahren der Fertigungssteuerung, Springer, Berlin, Heidelberg, New York, 2005.

-
- [LSW06] Linderoth, J.; Shapiro, A.; Wright, S.: The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142, 2006.
- [MD90] Minifie, J.; Davis, R.: Interaction effects on MRP nervousness, *International Journal of Production Research* 28, 1990.
- [Mer02] Merz, F.: *Mut zur Zukunft, Wie Deutschland wieder an die Spitze kommt*, Goldmann, Berlin, 2002.
- [MF04] Michalewicz, Z.; Fogel, D. B.: *How to Solve It: Modern Heuristics*, 2. Auflage, Springer, Berlin, Heidelberg, New York, 2004.
- [MSU99] Möhring, R. H.; Schulz, A.S.; Uetz, M.: Approximation in Stochastic Scheduling: The Power of LP-Based Priority Schedules, *Journal of ACM*, Vol. 46, No. 6, 1999.
- [Müs07] Müssig, F.: *Grafikpower zum Abtanzen, C'T*, Ausgabe 23, 2007.
- [MWG95] Mas-Colell, A.; Whiston, M.D.; Green, J.R.: *Microeconomic Theory*, Oxford University Press, 1995.
- [NEH83] Nawaz, M.; Enscore, Jr.; Ham, I.: A Heuristic Algorithm for the m-Machine, n-job Flow-shop Sequencing Problem, *Omega*, 11, 1983.
- [New97] Newborn, M.: *Kasparov versus Deep Blue, Computer Chess comes of age*, Springer, Berlin, Heidelberg, New York, 1997.
- [Nil82] Nilsson, N. J.: *Principles of Artificial Intelligence, Symbolic Computation*, Springer, Berlin, Heidelberg, New York, 1982.
- [Pal99] Palmer, D.: *Maintenance Planning and Scheduling Handbook*, McGraw-Hill, New York, 1999.
- [Pap83] Papadimitriou, C.: Games against nature, In: *Proceedings of the 24th IEEE Symp. on the Foundations of Computer Science (FOCS)*, 1983.
- [Pec99] Peckhaus, V.: Abduktion und Heuristik, In: *Rationalität, Realismus, Revision, Vorträge des 3. Internationalen Kongresses der Gesellschaft für Analytische Philosophie*, Julian Nida-Rümelin, Walter de Gruyter (Hrsg.): Berlin/New York 1999 (*Perspektiven der Analytischen Philosophie*; 23), Erlangen, 1999.
- [PW05] Pochet, Y.; Wolsey, L. A.: *Production Planning by Mixed Integer Programming*, Springer, Berlin, Heidelberg, New York, 2005.
- [Rei83] Reinefeld, A.: An improvement of the cout tree search algorithm, *ICCA Journal*, 1983.

-
- [RN03] Russel, S.; Norvig, P.: Artificial Intelligence: modern approach, Prentice Hall, Upper Saddle River, 2003.
- [Rob91] Robrade, A. D.: Dynamische Einprodukt-Lagerhaltungsmodelle bei periodischer Bestandsüberwachung, Physica Verlag, Heidelberg, 1991.
- [RS01] Römisch, W.; Schultz, R.: Multistage stochastic integer programming: introduction, In: M. Grötschel, S.O. Krumke, and J. Rambau, editors, On-line Optimization of Large Scale Systems, Springer, Berlin, Heidelberg, New York, 2001.
- [Rüt05] Rüter, M.: Ein Beitrag zur klassifizierenden Modularisierung von Verfahren für die Produktionsplanung, HNI-Verlagschriftenreihe, Paderborn, 2005.
- [Sch80] Schomburg, E.: Entwicklung eines betriebstypologischen Instrumentariums zur systematischen Ermittlung von Anforderungen an EDV-gestützte Produktionsplanungs- und -steuerungssysteme im Maschinenbau, Dissertation TH Aachen, Aachen, 1980.
- [Sch92] Schneeweiß, C.: Einführung in die Produktionswirtschaft, 4. Aufl. Springer, Berlin, Heidelberg, New York, 1992.
- [Sch01] Scholl, A.: Robuste Planung und Optimierung: Grundlagen, Konzepte und Methoden, Experimentelle Untersuchungen, Springer, Berlin, Heidelberg, New York, 2001.
- [Sch02] Schönsleben, P.: Integrales Logistikmanagement, Planung und Steuerung von umfassenden Geschäftsprozessen, 3. Aufl., Springer, Berlin, Heidelberg, New York, 2002.
- [Sch04] Schulte, C.: Logistik Wege zur Optimierung der Supply Chain, 4. Aufl., Verlag Vahlen, Kamp-Lintfort, 2004.
- [Sen00] Sen, S.: Stochastic Programming: Computational Issues and Challenges, In: encyclopedia of or/ms, SIE Department, University of Arizona, 2000.
- [She04] Sheffi, Y.: Demand Variability and Supply Chain Flexibility - Driving from pure Cost to Leand Flexibility Strategies; In: Prockl, G.; Bauer, A.; Pflaum, A.; Müller-Seinhart, U.: Entwicklungspfade und Meilensteine moderner Logistik: Skizzen einer Roadmap, Wiesbaden, 2004.
- [SHF94] Schöneburg, E.; Heinzmann, F.; Feddersen, S.: Genetische Algorithmen und Evolutionsstrategien, Verlag Addison-Wesley, Bonn, Reading, Massachusetts, 1994.

-
- [Spe92] Speith, G.: Vorgehensweise zur Beurteilung und Auswahl von Produktionsplanungs- und -steuerungssystemen, Dissertation, TH Aachen, Aachen, 1992.
- [SS04] Shmoys, D. B.; Swamy, C.: Stochastic optimization is (almost) as easy as deterministic optimization, In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS04), 2004.
- [SSV93] Schultz, R.; Stougie, L.; Van Der Vlerk, M.: Two-Stage Stochastic Integer Programming: A Survey, Paper 520, Groningen State, Institute of Economic Research, 1993.
- [Sta03] Stadtler, H.: Multilevel Lot Sizing with Setup Times and Multiple Constrained Resources: Internally Rolling Schedules with Lot-Sizing Windows. *Operations Research*, 51, 2003.
- [Tch07] Tchato, J. B.: Klassifizierung und Implementierung von Planungsverfahren, Diplomarbeit, Universität Paderborn, Wirtschaftsinformatik insbes. CIM, Paderborn, 2007.
- [Tem06] Tempelmeier, H.: Material-Logistik, Modelle und Algorithmen für die Produktionsplanung und -steuerung in Advanced Planning-Systemen, 6. Aufl., Springer, Berlin, Heidelberg, New York, 2006.
- [Tem04] Tempelmeier, H.: Produktion und Logistik, 6. Aufl., Springer, Berlin, Heidelberg, New York, 2004.
- [Tem03] Tempelmeier, H.: Material-Logistik, Modelle und Algorithmen für die Produktionsplanung und -steuerung und das Supply Chain Management, 5. Aufl., Springer, Berlin, Heidelberg, New York, 2003.
- [TT99] Thieß, G.; Tschiene, P.: Handbuch zur Wettkampflehre, Aachen, 1999.
- [VS04] Vahrenkamp, R.; Siepermann, C.: Produktionsmanagement, 5. Auflage, R. Oldenbourg Verlag, München Wien, 2004.
- [VW06] Voß, S.; Woodruff, D. L.: Introduction to Computational Optimization Models for Production Planning in a Supply Chain, Springer, Berlin, Heidelberg, New York, 2006.
- [Wem81] Wemmerlöv, U.: The ubiquitous EQQ – its relation to discrete lot sizing heuristics international aus *Journal of Operations & Production Management* 1, Emerald MCB Press, 1981.
- [Wem82] Wemmerlöv, U.: A comparison of discrete single stage lot-sizing heuristics with special emphasis on rules based on the marginal cost principle aus *Engineering Costs and Production Economics* 7, Elsevier Science, 1982.

-
- [Wen95] Wendt, O.: Tourenplanung durch Einsatz naturanaloger Verfahren, Deutscher Universitäts Verlag, Wiesbaden, 1995.
- [Wes05] Wesemann, J.: Aktionensteuerung: Herstellung der Konsistenz in Mikro- und Makrostrukturen, Diplomarbeit Universität Paderborn, Wirtschaftsinformatik insbes. CIM, Paderborn, 2005.
- [Wik05-ol] Wikipedia: Störung, Unter: <http://de.wikipedia.org/wiki/Störung>, 2005.
- [Wik07-ol] Wikipedia: Benchmarking, Unter: <http://de.wikipedia.org/wiki/Benchmark>, 2007.
- [Wik08-ol] Wikipedia: PSPACE, Unter: <http://de.wikipedia.org/wiki/PSPACE>, 2008.
- [WIZ07-ol] Welt in Zahlen: Vergleich Deutschland, USA, Indien, China, PKW je 1000 Einwohner, Unter: www.welt-in-zahlen.de, 2007.
- [Wöh02] Wöhe, G.: Einführung in die allgemeine Betriebswirtschaftslehre, Verlag Vahlen, 21. Aufl, München 2002.
- [Wol98] Wolsey, Laurence A.: Integer Programming. John Wiley & Sons, Inc., 1998.
- [WW58] Wagner, H. M.; Whitin, T. M.: Dynamic version of the economic lot size model, Management Science Vol. 5, Providence R.I., 1958.
- [WW84] Wemmerlöv, U.; Whybark, D. C.: Lot-sizing under uncertainty in a rolling schedule environment aus International Journal of Production Research 22, Emerald MCB Press, 1984.
- [Zäp98] Zäpfel, G.: Grundlagen und Möglichkeiten der Gestaltung dezentraler PPS-Systeme, In: Corsten, H.; Gössinger, R. (Hrsg.): Dezentrale Produktionsplanung und -steuerungs-Systeme, Kohlhammer, Stuttgart, 1998.
- [ZB05] Zäpfel, G.; Braune, R.: Moderne Heuristiken der Produktionsplanung am Beispiel der Maschinenbelegung, Vahlen Verlag, München, 2005.
- [ZK02] Zdrowomyslaw, N.; Kasch, R.: Betriebsvergleiche und Benchmarking für die Managementpraxis, Oldenbourg, 2002.
- [ZR87] Zoller, K.; Andreas, D.: Dynamische Bestellmengen- und Losgrößenplanung–Verfahrensübersicht und Vergleich aus OR Spectrum 9, Springer, Berlin, Heidelberg, New York, 1987.