

Ein parametrisierbares Verfahren zur Änderungsplanung für den Flexible Flow Shop mit integrierter Schichtmodellauswahl

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

vorgelegt von
Dipl.-Wirt.-Inf. Daniel Brüggemann
33098 Paderborn

Paderborn, August 2010

Dekan: Prof. Dr. Peter F. E. Sloane
Referent: Prof. Dr.-Ing. habil. Wilhelm Dangelmaier
Korreferentin: Prof. Dr. Leena Suhl

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe Wirtschaftsinformatik, insb. CIM am Heinz Nixdorf Institut der Universität Paderborn. Im Folgenden möchte ich mich für die während dieser Zeit erfahrene persönliche und fachliche Unterstützung bei den wichtigsten Menschen bedanken.

Als erstes möchte ich Herrn Prof. Dr.-Ing. habil. Wilhelm Dangelmaier für die wissenschaftliche Betreuung, die andauernde Bereitschaft zur Diskussion und den vielseitigen fachlichen Rat herzlich danken. Mein Dank gilt außerdem Frau Prof. Dr. Leena Suhl für die Übernahme des Korreferats und Herrn Prof. Dr.-Ing. Bernd Hellingrath und Herrn Dr. Andre Döring für die Teilnahme an meiner Promotionskommission sowie dem wertvollen Feedback.

Des Weiteren gilt mein Dank meinen Wegbegleitern am Institut. Die zahlreichen konstruktiven Diskussionen, die Unterstützung bei der Umsetzung der Ideen sowie die freundschaftliche Atmosphäre haben wesentlich zum Gelingen dieser Arbeit beigetragen.

Ganz herzlich bedanken möchte ich mich bei meinen Eltern und meiner Schwester für die in jeder nur erdenklichen Weise geleistete Unterstützung und Motivation. Abschließend gilt mein ganz besonderer Dank meiner Freundin. Trotz des großen Verzichts auf gemeinsame Zeit besaß sie immer viel Verständnis und hat mir in schwierigen Phasen meiner Promotion mit bestärkenden Worten zur Seite gestanden. Danke für Deine große Geduld!

Paderborn, August 2010
Daniel Brüggemann

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Symbolverzeichnis	ix
1 Einleitung	1
2 Problemdefinition	3
2.1 Produktionsplanung	3
2.1.1 Einführung in die Produktionsplanung	3
2.1.2 Ablaufplanung.....	7
2.1.3 Arten der Produktionsplanung	10
2.1.3.1 Prädiktive Produktionsplanung	10
2.1.3.2 Reaktive Produktionsplanung.....	10
2.2 Der Flexible Flow Shop.....	13
2.2.1 Aufbau des Produktionssystems	13
2.2.2 Zeitmodell	15
2.2.3 Planungspolitik	17
2.2.4 Zielsetzung	18
2.2.5 Komplexität der Planung.....	19
2.3 Planungsverfahren für den Flexible Flow Shop	21
2.3.1 Optimierungsverfahren	21
2.3.2 Heuristiken.....	22
2.4 Anforderungen an die Problemlösung	23
2.4.1 Anforderungen an ein Planungsverfahren	23
2.4.2 Anforderungen an ein Benchmarkingverfahren	25
3 Stand der Technik.....	27
3.1 Mehrstufige mathematische Modelle	27
3.1.1 Multi-Level Continuous Lot Sizing and Scheduling Problem.....	28
3.1.1.1 Grundmodell.....	29
3.1.1.1.1 Zielfunktion.....	29
3.1.1.1.2 Restriktionen	29
3.1.1.2 Modellerweiterungen	30
3.1.2 Multi-Level Discrete Lot Sizing and Scheduling Problem	31
3.1.2.1 Zielfunktion	32
3.1.2.2 Restriktionen	33
3.1.3 Optimale Verfahren für den Flexible Flow Shop	34
3.2 Heuristische Lösungsverfahren	36
3.2.1 Eröffnungsverfahren	36
3.2.1.1 Prioritätsregeln	36
3.2.1.2 Heuristische Verfahren für den Flexible Flow Shop.....	38

3.2.2	Verbesserungsverfahren	41
3.2.2.1	Definition der Nachbarschaft	43
3.2.2.2	Simulated Annealing.....	44
3.2.2.2.1	Allgemeines.....	44
3.2.2.2.2	Parameter.....	44
3.2.2.2.3	Ablauf	45
3.2.2.3	Tabu Search.....	47
3.2.2.3.1	Allgemeines.....	47
3.2.2.3.2	Parameter.....	48
3.2.2.3.3	Ablauf	48
3.2.2.4	Metaheuristiken für den Flexible Flow Shop.....	50
3.2.3	Methoden der reaktiven Planung.....	52
3.2.3.1	Strategien zum Umgang mit Störungen.....	52
3.2.3.2	Reaktive Verfahren für den Flexible Flow Shop	59
3.3	Bewertung von Planungsverfahren.....	60
4	Zu leistende Arbeit	65
4.1	Entwicklung eines mathematischen Modells	65
4.2	Entwicklung eines heuristischen Verfahrens	65
4.3	Entwicklung eines Benchmarkingverfahrens	66
5	Konzeption	69
5.1	Entwicklung eines mathematischen Modells	69
5.1.1	Grundmodell.....	69
5.1.1.1	Zielfunktion	71
5.1.1.2	Restriktionen	72
5.1.2	Erweiterungen zur reaktiven Planung.....	82
5.1.3	Gewichtete Zielfunktion	85
5.2	Entwicklung eines heuristischen Verfahrens	85
5.2.1	Aufbau des Planungsverfahrens.....	85
5.2.2	Parametrisierung	87
5.2.3	Entwicklung eines Verfahrens für den Horizont der Feinplanung	90
5.2.3.1	Eröffnungsverfahren.....	90
5.2.3.1.1	Bedarfssenkung	91
5.2.3.1.2	Bedarfserhöhung.....	94
5.2.3.1.3	Produktionsstufenabgleich	98
5.2.3.1.4	Hilfsfunktionen.....	101
5.2.3.2	Verbesserungsverfahren	111
5.2.3.2.1	Basislos bestimmen	113
5.2.3.2.2	Nachbarschaft erzeugen	114
5.2.3.2.3	Wahl eines Nachbarn	117
5.2.3.2.4	Aktualisierung der Variablen.....	117
5.2.3.2.5	Hilfsfunktionen.....	117
5.2.4	Entwicklung eines Verfahrens für den Horizont der Grobplanung	125
5.2.4.1	Modell der Grobplanung	126
5.2.4.1.1	Zielfunktion	128

5.2.4.1.2	Restriktionen	128
5.2.4.2	Bedarfsglättungsverfahren	131
6	Realisierung	135
6.1	Benchmarkingplattform	135
6.1.1	Stammdatenkomponente	136
6.1.1.1	Klassenstruktur der Stammdaten	136
6.1.1.2	Frontend der Stammdatenverwaltung	137
6.1.2	Benchmarkingdesign	139
6.1.2.1	Parametrisierungskomponente	139
6.1.2.2	Workflowkomponente	144
6.1.3	Reportingkomponente	148
6.1.4	Konfigurationskomponente	151
6.2	Experimentelle Untersuchung	151
6.2.1	Anwendungsszenario 1	152
6.2.2	Anwendungsszenario 2	157
7	Zusammenfassung und Ausblick	163
7.1	Ergebnis der Arbeit	163
7.2	Ausblick	164
Literaturverzeichnis		167
Anhang A: Eröffnungs- und Verbesserungsverfahren		179
Anhang A: Symbolverzeichnis		179
Anhang A: Pseudocode Eröffnungsverfahren		182
Anhang A: Pseudocode Verbesserungsverfahren		193
Anhang A: Pseudocode gemeinsame Hilfsfunktionen		201
Anhang B: Bedarfsglättungsverfahren		211
Anhang B: Symbolverzeichnis		211
Anhang B: Pseudocode		211
Anhang C: Daten des Benchmarkings		213
Anhang C: Anwendungsszenario 1		213
Anhang C: Anwendungsszenario 2		214

Abbildungsverzeichnis

Abbildung 1: Einordnung grundlegender Planungsverfahren	7
Abbildung 2: Kostenorientierte Störungswirkungen	12
Abbildung 3: Allgemeiner Flexible Flow Shop.....	14
Abbildung 4: Aufbau eines Schichtmodells.....	16
Abbildung 5: Menge aller gültigen Schichtmodelle	17
Abbildung 6: Ablauf einer rollierenden Planung.....	18
Abbildung 7: Zusammenhang der Komplexitätsklassen	20
Abbildung 8: Modellierung und Lösungsverfahren.....	27
Abbildung 9: Auswirkungen der Rüstzustandsübertragung	31
Abbildung 10: Klassifikation verbreiteter Metaheuristiken	42
Abbildung 11: Allgemeines Prinzip der Nachbarschaftssuche.....	43
Abbildung 12: Grundsätzlicher Ablauf des Simulated Annealing	46
Abbildung 13: Grundsätzlicher Ablauf der Tabu Search.....	49
Abbildung 14: Beispiel eines Ursprungsplans	55
Abbildung 15: Einfaches Right Shifting.....	56
Abbildung 16: Right Shifting and Jumping	56
Abbildung 17: Änderung der Ausführungsreihenfolge	57
Abbildung 18: Durchlaufzeitverkürzung durch Splitting	58
Abbildung 19: Ablaufsteuerung der Benchmarkingplattform.....	67
Abbildung 20: Horizonte der Planung	86
Abbildung 21: Gesamtablauf des heuristischen Verfahrens	87
Abbildung 22: Gesamtaufbau des Eröffnungsverfahrens	91
Abbildung 23: Ablauf einer Bedarfssenkung.....	92
Abbildung 24: Beispiel einer Bedarfssenkung	94
Abbildung 25: Ablauf der Bedarfserhöhung	96
Abbildung 26: Beispiel einer Bedarfserhöhung	98
Abbildung 27: Beispiel: Interdependenzen zwischen Produktionsstufen	99
Abbildung 28: Ablauf der Funktion Produktionsstufenabgleich.....	100
Abbildung 29: Beispiel: aufgelöste Interdependenzen zwischen Produktionsstufen	100
Abbildung 30: Bestimmung des Startzeitpunktes eines Loses	102
Abbildung 31: Ablauf der Funktion: KalkLoszeiten	103
Abbildung 32: Berechnung des frühesten Anfangszeitpunktes.....	104
Abbildung 33: Ablauf der Berechnung von δ	105
Abbildung 34: Ablauf der Berechnung vom Losende.....	106
Abbildung 35: Übersicht über Inkonsistenzen der mehrstufigen Planung.....	109
Abbildung 36: Ablauf der Funktion: KalkPufferauslastung	110
Abbildung 37: Gesamtablauf des Verbesserungsverfahrens	112
Abbildung 38: Ablauf der Basislosbestimmung.....	113
Abbildung 39: Basislos-Flexibilitätsberechnung.....	114
Abbildung 40: Ablauf der Nachbarschaftsgenerierung	115
Abbildung 41: Berechnung losspezifischer Kosten.....	118
Abbildung 42: Alternative Startzeitpunkte eines Loses	120
Abbildung 43: Ablauf der Funktion: LoseVerschiebenOderTeilen	122
Abbildung 44: Ablauf der Funktion: InkremSchichtmodell	124
Abbildung 45: Ablauf des Bedarfsglättungsverfahrens	132
Abbildung 46: Klassendiagramm der Stammdatenkomponente	137

Abbildung 47: Abbildung der Erzeugnisstruktur.....	138
Abbildung 48: Stammdatenmasken der Benchmarkingplattform	139
Abbildung 49: Ablauf des Bedarfsgenerators	142
Abbildung 50: Bedarfsgenerator in der Benchmarkingkomponente.....	143
Abbildung 51: Verfahrensoptionen	143
Abbildung 52: Ablaufdiagramm Workflowkomponente	145
Abbildung 53: Übersichtsbrowser	149
Abbildung 54: Statistikbrowser mit Liniendiagramm	150
Abbildung 55: Gantt-Chart (Plan-Werte).....	151
Abbildung 56: Aufbau von Anwendungsszenario 2	157

Tabellenverzeichnis

Tabelle 1: Störungsklassen und -ursachen.....	11
Tabelle 2: Kostensituationen bei der Vorwärtsrechnung	119
Tabelle 3: Ergebnis AZ 1: Schichtmodellflexibilität (keine BV).....	153
Tabelle 4: Ergebnis AZ 1: Schichtmodellflexibilität (mit BV)	154
Tabelle 5: Ergebnis AZ 1: Rüstzeitintegration (keine BV)	155
Tabelle 6: Ergebnis AZ 1: Rüstzeitintegration (mit BV).....	156
Tabelle 7: Ergebnis AZ 2: Vergleich der Feinplanungsverfahren (keine BV)	159
Tabelle 8: Ergebnis AZ 2: Vergleich der Feinplanungsverfahren (mit BV).....	160
Tabelle 9: Ergebnis AZ 2: Hinzunahme der Verfahren der Grobplanung	160
Tabelle 10: AZ 1: Verfügbare Schichten mit Leistungsgrad und Kosten.....	213
Tabelle 11: AZ 1: Schichtmodelle mit Schichtzuordnung.....	213
Tabelle 12: AZ 1: Aufbau des Produktionssystems (Schichtflexibilität)	213
Tabelle 13: AZ 1: Aufbau des Produktionssystems (Rüstzeitanalyse).....	213
Tabelle 14: AZ 1: Konfiguration der Bedarfe.....	214
Tabelle 15: AZ 1: Einstellungen des Modells der Feinplanung	214
Tabelle 16: AZ 1: Einstellungen des Eröffnungsverfahrens	214
Tabelle 17: AZ 1: Einstellungen des Verbesserungsverfahrens	214
Tabelle 18: AZ 2: Verfügbare Schichten mit Leistungsgrad und Kosten.....	214
Tabelle 19: AZ 2: Schichtmodelle mit Schichtzuordnung.....	215
Tabelle 20: AZ 2: Aufbau des Produktionssystems	215
Tabelle 21: AZ 2: Konfiguration der Bedarfe.....	215
Tabelle 22: AZ 2: Einstellungen des Eröffnungsverfahrens	216
Tabelle 23: AZ 2: Einstellungen des Verbesserungsverfahrens	216
Tabelle 24: AZ 2: Einstellungen der Grobplanung	216

Symbolverzeichnis

$a_{e,j}$	Direktbedarfskoeffizient, der angibt, wie viele Einheiten des Erzeugnisses e zur Herstellung einer Einheit des Erzeugnisses j benötigt werden
A	Auftrag
b_e^g	Menge des Bedarfs an Erzeugnis e , der am Anfang des Planungshorizontes bereits gedeckt ist
$b_{e,x}$	Bedarf nach Erzeugnis e in Periode x
$b_{ui,ps}$	Bedarfsumfang einer Bedarfssenkung/-erhöhung i auf der Produktionsstufe ps
B	Hinreichend große Zahl
e	Erzeugnisindex
e^{ges}	Anzahl an Erzeugnissen
e_{pl}	Anzahl aller Erzeugnisse, die auf Produktionslinie pl gefertigt werden
e_{pu}	Anzahl aller Erzeugnisse, die über den Puffer pu laufen
$ep_{e,pu}$	Lagerfähigkeit von Erzeugnis e in Puffer pu
E	Menge aller Erzeugnisse
$E_{e,pl}$	Menge der Erzeugnisse, auf die von Erzeugnis e auf der Produktionslinie pl gerüstet werden kann
E_{pl}	Menge aller Erzeugnisse, die auf Produktionslinie pl gefertigt werden
E_{pu}	Menge der Erzeugnisse, die über den Puffer pu laufen
E_{ps}	Menge aller Erzeugnisse, die auf Produktionsstufe ps gefertigt werden
$f_{e,pl}$	Der Rüstzeitfortschritt von Erzeugnis e , der am Anfang des Planungshorizontes bereits gerüstet wurde
$flex_{ps}$	Flexibilität der Produktionsstufe ps
$flex_{ps}^{einp}$	Flexibilität in der Einplanung auf der Produktionsstufe ps
FSZ	Menge der Mikroperioden, in denen eine Fixierung der Schichtmodelle vorgenommen wird
FZ	Menge der Mikroperioden, die der eingefrorenen Zone zugeordnet sind und in denen eine Fixierung der Produktion vorgenommen wird
$h_{e,pl,\mu}$	Binärvariable, die beschreibt, ob ein Erzeugnis e in Mikroperiode μ auf Produktionslinie pl im alten Plan produziert wurde
$h_{e,pl,\mu}^c$	Hilfsvariable zur Linearisierung

$h_{e,pl,\mu}^d$	Hilfsvariable für den Übergang des virtuellen Lagers in den realen Lagerbestand nach Beendigung eines Loses von Erzeugnis e in Mikroperiode μ der Produktionslinie pl
$h_{e,pl,\mu_1,\mu_2}^{dp}$	Hilfsvariable, die die produzierte Menge von Erzeugnis e , bei einer durchgehenden Produktion von Mikroperiode μ_1 bis μ_2 auf Produktionslinie pl , annimmt
$h_{e,pl,\mu}^{I,vir,ab}$	Hilfsvariable zur Rundung
h_{e,pl,μ_1,μ_2}^p	Hilfsvariable, die angibt, ob das Erzeugnis e von Mikroperiode μ_1 bis μ_2 durchgehend auf Produktionslinie pl produziert wird
$h_{e,pl,\mu}^s$	Hilfsvariable, die den Sekundärbedarf von Erzeugnis e in Mikroperiode μ auf Produktionslinie pl angibt
$h_{e,pl,\mu}^{sb}$	Hilfsvariable zur Rundung
$h_{e,pl,\mu}^v$	Hilfsvariable zur Linearisierung
h_{i,j,pl,μ_1,μ_2}^x	Hilfsvariable, die angibt, ob ein Rüstvorgang auf Produktionslinie pl von Erzeugnis i nach Erzeugnis j innerhalb den Mikroperioden μ_1 und μ_2 stattfindet
$I_{e,x}$	Lagerbestand für Erzeugnis e am Ende der Periode x
I_e^{init}	Initialer Lagerbestand von Erzeugnis e
$I_{e,pl,\mu}^{vir}$	Virtueller Lagerbestand eines Erzeugnisses e in Mikroperiode μ auf der Produktionslinie pl
$I_{e,pl,\mu}^{vir,r}$	Virtueller Lagerbestand eines Erzeugnisses e in Mikroperiode μ auf der Produktionslinie pl , wenn ein Los bei der rollierenden Planung makroperiodenübergreifend produziert wird
k_l^b	Kapazitätsbedarf von Los l
$k_{pl,t}$	Kapazitätsangebot einer Produktionslinie pl in Makroperiode t
$k_{pl,t}^{max}$	Maximales Kapazitätsangebot der Produktionslinie pl in Makroperiode t
k_{ps}^{diff}	Kapazitätserweiterungspotential auf der Produktionsstufe ps
k_t	Kapazitätsangebot in Periode t
k_{fe}	Kapazitätsfaktor für die Lagerung einer Einheit von Erzeugnis e
l	Losindex
$lg_{sm,pl,\mu}$	Leistungsgrad auf der Produktionslinie pl im Schichtmodell sm für Mikroperiode μ
lk_e	Lagerkostenfaktor für eine Einheit des Erzeugnisses e

lk_l	Lagerkosten von Los l
lm_l	Größe von Los l
L	Menge von Losen
LO_{ps}^{sum}	Kumulierte freie Kapazität auf der Produktionsstufe ps
$mengDiff$	Mengendifferenz aus Restbedarf und Losrest
mp	Maximaler Abschnitt durchgehender Produktion
mr_{pl}	Maximale Rüstzeiten zwischen zwei Erzeugnissen auf der Produktionslinie pl
M	Menge aller Mikroperioden
MM	Anzahl der Mikroperioden pro Makroperiode
μ	Mikroperiodenindex
n	Anzahl an Makroperioden
n_e	Anzahl der Nachfolger des Erzeugnisses e
n^{ps}	Anzahl der Produktionsstufen
N_e	Menge der Nachfolger des Erzeugnisses e
$p_{e,pl,t}$	Produktionskostensatz für ein Erzeugnis e in Makroperiode t auf Produktionslinie pl
$p_{e,x}$	Produktionskostensatz für Erzeugnis e in Periode x
p_{fpu}	Bestand des Puffers pu
p_{fpu}^{max}	Maximale Pufferkapazität des Puffers pu
pl	Produktionslinienindex
pl_{ps}	Anzahl an Produktionslinien einer Produktionsstufe ps
plf_{ps}	Planungsflexibilität der Produktionsstufe ps
$prf_{e,pl}$	Produktionsfähigkeit des Erzeugnisses e auf Produktionslinie pl
pu	Pufferindex
ps	Produktionsstufenindex
PL	Menge der Produktionslinien
PL_{ps}	Menge der Produktionslinien der Produktionsstufe ps
PS	Menge an Produktionsstufen
$PZ(f)$	Planungszeitraum f
PU	Menge aller Puffer
$q_{e,pl,t}$	Losgröße für Erzeugnis e in Makroperiode t auf Produktionslinie pl

$q_{e,x}$	Losgröße für Erzeugnis e in Periode x
$rest_l$	Restmenge von Los l , die noch nicht zur Bedarfsdeckung verwandt wurde
rk_e	Produktabhängige Rüstkosten für Erzeugnis e
rz^{diff}	Rüstzeitdifferenz
rz_e	Rüstzeit für Erzeugnis e
$rz_{e,pl}$	Rüstzeit für Erzeugnis e auf Produktionslinie pl
$rz_{i,j,pl}$	Rüstzeit von Erzeugnis i auf Erzeugnis j auf Produktionslinie pl
$rz_{l,k,pl}$	Rüstzeit von Los l auf Los k auf Produktionslinie pl
rp_{pl}	Der Rüstzustand einer Produktionslinie pl , der am Anfang des Planungshorizontes vorliegt
$RS_{e,x}$	Rückstandsmenge eines Erzeugnisses e in Periode x
s	Schichtindex
$sk_{sm,pl,t}$	Kosten für Schichtmodell sm für die Makroperiode t und die Produktionslinie l
sm	Schichtmodellindex
$sm_{pl,t}$	Schichtmodell
$sm_{sm,pl,t}$	Binärvariable, die angibt, ob das Schichtmodell sm in Makroperiode t für die Produktionslinie pl gewählt ist
$sms_{sm,pl,t}$	Indikator für die Schichtmodellauswahl, der anzeigt, ob das Schichtmodell sm im Ursprungsplan in Makroperiode t auf der Produktionslinie pl eingeplant ist
$SM_{pl,t}$	Menge aller in Makroperiode t für die Produktionslinie pl verfügbaren Schichtmodelle
t	Makroperiodenindex
tb_e	Stückbearbeitungszeit für Erzeugnis e
$tb_{e,pl}$	Stückbearbeitungszeit für Erzeugnis e auf Produktionslinie pl
$tb_{e,pl}^{temp}$	Tatsächliche Bearbeitungszeit von Erzeugnis e
T	Länge des Planungshorizontes
$T_{i,ps}$	Bedarfstermin einer Bedarfssenkung/-erhöhung i auf der Produktionsstufe ps
$T_{l,A,ps}^e$	Ende aller zum Auftrag A zugehörigen Lose des Loses l auf der Produktionsstufe ps

$T_{l,A,ps}^s$	Beginn aller zum Auftrag A zugehörigen Lose des Loses l auf der Produktionsstufe ps
$T_{l,pl,ps}^e$	Ende des Loses l auf Produktionslinie pl der Produktionsstufe ps
$T_{l,pl,ps}^s$	Startzeit des Loses l auf Produktionslinie pl der Produktionsstufe ps
$T_{l,ps}^{FAT}$	Frühester Anfangstermin des Loses l der Produktionsstufe ps
$T_{l,ps}^{SET}$	Spätester Endtermin des Loses l der Produktionsstufe ps
$T_{ps}^{b,FP}$	Beginn des Feinplanungshorizontes auf Produktionsstufe ps
$T_{ps}^{b,GP}$	Beginn des Grobplanungshorizontes auf Produktionsstufe ps
$T_{ps}^{b,eZ}$	Beginn der eingefrorenen Zone der Produktionsstufe ps
$T_{ps}^{e,eZ}$	Ende der eingefrorenen Zone der Produktionsstufe ps
$T_{ps}^{e,fS}$	Ende des fixierten Schichthorizontes der Produktionsstufe ps
$T_{ps}^{e,FP}$	Ende des Feinplanungshorizontes der Produktionsstufe ps
$T_{ps}^{e,GP}$	Ende des Grobplanungshorizontes der Produktionsstufe ps
$T_{s,pl,ps}^b$	Beginn einer Schicht s auf der Produktionslinie pl der Produktionsstufe ps
$T_{s,pl,ps}^e$	Ende der Schicht s auf der Produktionslinie pl der Produktionsstufe ps
uk_e	Umlanungskosten, die für die Umlanung eines Erzeugnisses e anfallen
$v_{pl,t}$	Anzahl der Erzeugnisse, die auf Produktionsline pl in Makroperiode t produziert wurden
vk_e	Verzugskostenfaktor für eine Einheit des Erzeugnisses e
vk_l	Verzugskosten von Los l
vlz_e	Vorlaufzeit von Erzeugnis e
vlz_e^v	Vorlaufzeit von dem Vorgängererzeugnis von Erzeugnis e
$w_{e,pl,t}$	Binärvariable, die angibt, ob zu Beginn der Makroperiode t auf Produktionslinie pl der Rüstzustand der Vorperiode für Erzeugnis e übernommen werden kann
$x_{e,pl,x}$	Binärvariable, die angibt, ob in der Periode x auf Produktionslinie pl eine Umrüstung für Erzeugnis e stattfindet
$x_{e,t}$	Binärvariable, die angibt, ob in der Periode t eine Umrüstung für Erzeugnis e stattfindet

$y_{e,x}$	Binärvariable, die angibt, ob in der Periode x eine Umrüstung stattfindet
$zeitDiff$	Zeitdifferenz zwischen Bedarfsdatum und Enddatum
ZD_{NL}	Differenz eines Loses zu dem Nachfolgerlos der gleichen Produktionsstufe
ZD_{NS}	Differenz eines Loses zu einem weiteren Los auf der nachgelagerten Produktionsstufe
ZD_{VL}	Differenz eines Loses zu dem Vorgängerlos der gleichen Produktionsstufe
ZD_{VS}	Differenz eines Loses zu einem weiteren Los auf der vorgelagerten Produktionsstufe
$Z_{e,pl,\mu}$	Kapazität für Erzeugnis e in Mikroperiode μ auf Produktionslinie pl

1 Einleitung

„Eines Tages wird alles gut sein, das ist unsere Hoffnung.
Heute ist alles in Ordnung, das ist unsere Illusion.“

(Voltaire)

In den letzten Jahren forcieren eine wirtschaftlich angespannte Situation mit einer sinkenden Kaufkraft, eine abnehmende kundenspezifische Markenaffinität bei der Auswahl von Produkten und ein permanenter Wandel der Anforderungen des Marktes einen erhöhten Wettbewerbsdruck. Unternehmen müssen daher mehr denn je die Fähigkeit besitzen, flexibel auf Auftragsschwankungen reagieren zu können und das bei einer gleichzeitig möglichst wirtschaftlichen Nutzung der vorhandenen Ressourcen. Als „*Kern eines jeden Unternehmens*“ [Schu06] ist von diesen Entwicklungen besonders die Produktionsplanung und -steuerung betroffen. An diese werden umso höhere Anforderungen gestellt, desto komplexer das vorliegende Produktionssystem ist. In dieser Arbeit wird ein besonders komplexes und gleichzeitig in der betrieblichen Praxis häufig vorkommendes Produktionssystem, ein Flexible Flow Shop, betrachtet. Beispiele hierfür finden sich in der Automobil-, Elektro-, Maschinenbau-, Textil- und Nahrungsmittelindustrie. Die in der Literatur vorherrschende Beschreibung wird jedoch in der Art erweitert, als dass neben einer reinen Belegungsplanung vorhandener Produktionslinien auch eine parallele Bestimmung wirtschaftlicher Schichtmodelle erfolgen soll. So stellen die Arbeitskosten nicht nur einen hohen Anteil an den gesamten Produktionskosten dar, sondern deren flexible Zuordnung ermöglicht darüber hinaus, die Kapazitätsangebote in einer effektiven Art zu variieren. Zudem wird ein dynamisches Planungsumfeld betrachtet. In diesem können für die zu planenden Bedarfe und Kapazitäten jederzeit Änderungen (im weiteren Verlauf dieser Arbeit als Störung bezeichnet) auftreten, die eine unveränderte Ausführung eines vormals erstellten Produktionsplans verhindern. Beispiele für solche Störungen sind prozessbedingte Unsicherheiten (z.B. Maschinenausfälle) sowie nachfrageseitige Ereignisse wie Eilaufträge oder Kundenstornierungen. In der Praxis wird häufig versucht, die Auswirkungen solcher Unsicherheiten über Sicherheitsbestände und Pufferzeiten zu minimieren. Dies führt jedoch zu höheren Kosten sowie einer verminderten Produktivität des gesamten Produktionssystems. Eine Alternative stellen Änderungsplanungsverfahren dar, deren Ziel u.a. die bestmögliche Integration von Störungen in einen bestehenden Plan ist. Dadurch können eine hohe Nervosität des Plans (welche gravierende Auswirkungen auf die vom Produktionsplan abhängigen Unternehmensbereiche – z.B. der Werkslogistik oder den Einkauf – haben kann) sowie jeweils lange Rechenzeiten vermieden werden. Um einen bestehenden Plan möglichst wenig zu verändern, stehen verschiedene Mechanismen zur Verfügung. Hierzu zählen bspw. eine Veränderung der Ausführungsreihenfolge nur einzelner Lose oder – wie heute in vielen Unternehmen möglich – eine kurzfristige Adaption vormals geplanter Schichtmodelle. Dadurch kann z.B. binnen kürzester Zeit eine weitere Nachschicht eingeplant und folglich das Kapazitätsangebot erhöht werden. Zur Planung eines allgemeinen Flexible Flow Shop werden in der Literatur Lösungen auf Basis unterschiedlicher Techniken diskutiert. Es ist jedoch zu konstatieren, dass aufgrund der Komplexität des Problems, beson-

ders bei der Forderung nach einer integrierten Planung aller Produktionsstufen, heuristische Vorgehensweisen am zielführendsten sind. Auch wenn mit einer Heuristik keine Optimalität garantiert werden kann, so erstellen sie im Allgemeinen eine relativ gute Lösung in kurzer Zeit. Daher soll auch in dieser Arbeit ein heuristisches Verfahren entwickelt werden. Neben kurzen Lösungszeiten wird das Verfahren zudem verschiedene Möglichkeiten zur Parametrisierung bieten. Damit soll ein Unternehmen das Verfahren individuellen Bedürfnissen anpassen können. Für den Einsatz unter praxisrelevanten Bedingungen ist es jedoch entscheidend, wie stark sich die Resultate der Heuristik von einem Optimum unterscheiden. Aus diesem Grund wird zudem ein mathematisches Modell formuliert und dessen Ergebnis mit dem der Heuristik verglichen. Zur Bewertung dieser Verfahren wird eine Benchmarkingplattform erstellt. Diese soll anhand unterschiedlicher Kenngrößen auf Basis gleicher Bedingungen (z.B. dem gleichen dynamischen Planungsumfeld) die Verfahren vergleichen sowie deren Ergebnisse transparent darstellen.

Diese Arbeit ist in sieben Kapitel gegliedert. Im Anschluss an diese Einleitung wird in Kapitel 2 die Problemstellung dieser Arbeit vorgestellt. Dabei wird zunächst ein genereller Einblick in die Aufgaben der Produktionsplanung (Kapitel 2.1) und die Merkmale des Flexible Flow Shop (Kapitel 2.2) gegeben. An diese schließt sich im Kapitel 2.3 eine Klassifikation der grundsätzlichen Lösungsmöglichkeiten an. Das Kapitel 3 arbeitet den Stand der Technik auf. Dazu werden zuerst die als relevant betrachteten mathematischen Modelle abgegrenzt (Kapitel 3.1) und anschließend detailliert erläutert. Darauf folgen die relevanten Entwicklungen aus dem Bereich der heuristischen Lösungsverfahren (Kapitel 3.2). Hierbei wird eine generelle Unterscheidung hinsichtlich unterschiedlicher Verfahrenstypen vorgenommen. Abschließend werden in Kapitel 3.3 Möglichkeiten zur Bewertung von Planungsverfahren diskutiert. Im Kapitel 4 werden dann die aus dem Stand der Technik sowie der Problemstellung resultierenden zu leistenden Arbeiten vorgestellt. In Kapitel 5 werden die entwickelten Verfahren dargestellt. Zu Beginn wird in Kapitel 5.1 das mathematische Modell mit unterschiedlichen Erweiterungen betrachtet. Dem schließt sich in Kapitel 5.2 das heuristische Verfahren an. Im Kapitel 6 werden die Verfahren evaluiert. Hierzu werden in Kapitel 6.1 die Komponenten und Funktionalitäten der Benchmarkingplattform vorgestellt. Im Kapitel 6.2 erfolgt anschließend der eigentliche Vergleich der Lösungsgüte der Verfahren. Den Abschluss der Arbeit bildet das Kapitel 7 mit einem Fazit und einem Ausblick auf zukünftig mögliche Entwicklungen.

2 Problemdefinition

„Wer den Hafen nicht kennt, in den er segeln will,
für den ist kein Wind der richtige.“

(Lucius Annaeus Seneca)

Um ein Verständnis für die Probleme und die im Rahmen dieser Arbeit entwickelten Lösungen zu bekommen, werden im Folgenden die Grundlagen gelegt. Hierzu erfolgt im Kapitel 2.1 eine allgemeine Einführung in die Planung der betrieblichen Produktion. Neben der Schaffung eines einheitlichen Begriffsverständnisses wird zudem ein besonderer Schwerpunkt auf dem Dilemma der Ablaufplanung liegen. Der zweite Teil der Problemdefinition präzisiert in Kapitel 2.2 das in dieser Arbeit unterstellte Produktionssystem. Hierzu werden zunächst die Problemklasse des Flexible Flow Shop allgemein beschrieben und anschließend um die Annahmen dieser Arbeit erweitert. Das dann folgende Kapitel führt die Heuristik und die Optimierung als methodische Ansätze zur Lösung von Ablaufplanungsproblemen ein. Abschließend werden die Anforderungen an die Problemlösung in Kapitel 2.4 zusammengefasst.

2.1 Produktionsplanung

2.1.1 Einführung in die Produktionsplanung

Produktion stellt die Bildung von Faktorkombinationen in betrieblichen Systemen durch Anwendung technischer und konzeptioneller Verfahren dar. Dabei werden die zur Verfügung stehenden Produktionsfaktoren in neue absetzbare Güter, die als Erzeugnis¹ bezeichnet werden, transformiert (vgl. [Cors07]). Nach Dangelmaier sind Transformationsprozesse „[...]Geschehen, bei denen eine Menge an materiellen und immateriellen Elementen als Input eingesetzt wird, um einen andersartigen Output zu erhalten. Handelt es sich bei einem solchen Transformationsprozess um die Produktion, so bezeichnet man den zu transformierenden Input und den angestrebten Output als Güter. Output der Produktion sind die Produkte, wobei materielle und immaterielle Produkte sowie Dienstleistungen unterschieden werden. Input der Produktion sind die Produktionsfaktoren“ [Dang09]. Produktionsfaktoren können in Repetier- und Potentialfaktoren differenziert werden. Einmalig in die Produktion einfließende Güter werden als Repetierfaktoren bezeichnet (z.B. Roh-, Hilfs- und Betriebsstoffe sowie Vor- bzw. Zwischenerzeugnisse). Potentialfaktoren sind dagegen dauerhaft nutzbar und können mehrmals in der Produktion eingesetzt werden (z.B. Arbeitskräfte und Maschinen²). Die Verrichtungen/Tätigkeiten der Potentialfaktoren haben quantitative und qualitative Potentiale und werden als Kapazität bezeichnet. Die quantitative Kapazität bezeichnet die Menge der möglichen Einsätze innerhalb einer bestimmten Zeit mit entsprechender Intensität des

¹ „Erzeugnisse sind in sich geschlossene, aus einer Anzahl von Gruppen und/oder Teilen bestehende funktionsfähige Gegenstände (z.B. Maschinen, Geräte) als Fertigungs-Enderzeugnis“ (DIN 6789). Im Rahmen dieser Arbeit werden die Begriffe „Erzeugnis“ und „Produkt“ synonym verwendet.

² Im Rahmen dieser Arbeit werden die Begriffe „Maschine“ und „Produktionslinie“ synonym verwendet.

Potentialfaktors (z.B. Bestimmung des Einsatzes, der Dauer und der Geschwindigkeit von Maschinen und Arbeitskräften). Die qualitative Kapazität beschreibt die Eigenart und Güte der Leistungen, die ein Potentialfaktor erbringen kann (so kann es z.B. auf einer Produktionsstufe eines Unternehmens spezielle Maschinen zur Herstellung nur einer bestimmten Erzeugnisart geben). Ein Produktionssystem ist die Summe aller Arbeitssysteme³ in einem festgelegten Bereich, dessen Aufgabe es ist, am Eingang in das System eingehende Materialien in einen definierten Ausgangszustand zu transformieren und am Ausgang abzugeben (vgl. [VDI 93]).⁴ Ein solches Produktionssystem kann nach der Fertigungsart und der Fertigungsform unterschieden werden. „*Die Fertigungsart wird charakterisiert durch die Häufigkeit der Leistungswiederholungen im Produktionsprozess. Eine Differenzierung erfolgt anhand der Auflagenhöhe von Fertigungsaufträgen und der Wiederholhäufigkeit gleicher oder ähnlicher Fertigungsobjekte. Die Fertigungsart hat maßgebend Einfluss auf die Gestaltung des Produktionsprozesses. Bestimmt werden die Gestaltung des Fertigungsablaufs, der Vorbereitungsgrad der Fertigung sowie die Flexibilität und der Automatisierungsgrad der Fertigungsmittel*“ [GKA07]. Hierbei lassen sich folgende Merkmalsausprägungen differenzieren: Einzelfertigung, Einzel- und Kleinserienfertigung, Serienfertigung und Massenfertigung. Bei der Einmal-, Einzel- und Kleinserienfertigung werden nur sehr geringe Stückzahlen produziert. Im Rahmen der Serienfertigung werden Erzeugnisse von einer geringen bis zu einer großen Stückzahl hergestellt. Der hohe Vorbereitungsaufwand hat zum Ziel, unter Berücksichtigung des hohen Anteils an Wiederholungen im Prozess, eine möglichst kostengünstige Produktion zu erreichen. Im Extremfall der Serienfertigung, der Massenfertigung, erfolgt die Produktion einer großen Auflagenhöhe von identischen bzw. fast identischen Erzeugnissen. Daraus resultiert ein Fertigungsablauf, der über längere Zeiträume konstant ist. Dies setzt hoch spezialisierte Arbeitssysteme voraus, die dem Fertigungsablauf angepasst sind. Vor dem Hintergrund dieser Tatsache begründet sich eine hohe Störempfindlichkeit der Massenfertigung (vgl. [GKA07]). Im Gegenteil zur Fertigungsart beschreibt die Fertigungsform nicht die Menge, sondern die räumliche Anordnung sowie die Abstimmung der Kapazitäten der Arbeitssysteme und deren Transportbeziehungen. Hierbei werden fünf Merkmalsausprägungen unterschieden (vgl. [GKA07]). Dies sind die Baustellenfertigung, Werkstattfertigung, Gruppen-/Linienfertigung, Insel-/Zellenfertigung sowie die Fließ- bzw. Reihenfertigung. Im Rahmen dieser Arbeit ist die Fließ- und Reihenfertigung von zentralem Interesse. Hierbei werden die Arbeitssysteme so angeordnet, dass die Erzeugnisse möglichst ohne Unterbrechung und mit möglichst wenigen Zwischentransporten produziert werden können (vgl. [WöDö05]). Nach der Fertigstellung der Erzeugnisse werden diese direkt zum nächsten Arbeitssystem transportiert, wodurch kurze Durchlaufzeiten erzielt werden. Dabei sind die Taktzeiten der einzelnen Arbeitssysteme in der Regel aufeinander abgestimmt. Da eine solche Fertigungsform über keine bzw. eine nur sehr geringe Flexibilität verfügt, ist eine gute Planung besonders wichtig.

In der Produktion bezeichnet Planung grundsätzlich einen auf Basis unvollkommener Information durchgeführten, systematischen und rationalen Prozess zur Lösung von (Entscheidungs-)Problemen unter Beachtung subjektiver Ziele (vgl. [DoSc05]). Im Rahmen der Pro-

³ Arbeitssysteme sind Zusammenfassungen von Arbeitskräften und Betriebsmitteln zu kleinsten elementaren Einheiten im Produktionsprozess (vgl. [GüTe05]).

⁴ Im Rahmen dieser Arbeit wird ausschließlich die diskrete bzw. diskontinuierliche Produktion betrachtet.

duktionsplanung erfolgt hierbei die Festlegung, wie der Produktionsprozess in einem Unternehmen ablaufen soll.⁵ Da für die Produktionsplanung der Output in der Regel in Form von Nachfragemengen exogen gegeben ist, liegt der Fokus auf einer Minimierung des Einsatzes der Inputfaktoren. Demnach besteht das Ziel in dem möglichst kostenminimalen Einsatz der vorhandenen Kapazitäten (bspw. Arbeitskräfte, Maschinen) zur Fertigung einer vorgeschriebenen Menge an Erzeugnissen. Zu diesem Zweck lässt sich die Produktionsplanung in drei Planungsebenen einteilen. Dazu zählen die strategische, die taktische und die operative Planung (vgl. [JaBi99]).⁶ Die auf diesen drei Planungsebenen getroffenen Entscheidungen differenzieren sich hinsichtlich des Planungshorizontes⁷, der Bedeutung für das Unternehmen, des Aggregationsgrades der verwendeten Daten sowie der Managementebenen (vgl. [GüTe00]). Ziel dieser Dekomposition ist es, die hohe Komplexität der Produktionsplanung zu verringern, indem einfache, handhabbare Teilprobleme erzeugt werden. Auf der obersten Ebene, der strategischen Planung, wird ein Planungszeitraum von fünf bis zehn Jahren betrachtet⁸. Die Hauptaufgabe ist das Schaffen und Erhalten einer wettbewerbsfähigen Produktion (vgl. [Dang09]). Dies umfasst die Festlegung der Unternehmensziele, der Ressourcen zur Erreichung dieser Ziele sowie der Strategien zur Beschaffung, den Einsatz und Verwendung dieser Ressourcen (vgl. [KiSt01]). Die taktische Planung konkretisiert die Vorgaben der strategischen Planung durch Produktionstechnologie und die Struktur des Produktionssystems und umfasst einen Planungszeitraum von ca. ein bis fünf Jahren. Zudem werden die Potentialfaktoren angepasst sowie genauere Entscheidungen über die Erzeugnisse, die Produktionsplanung und die Personal- und Betriebsmittelkapazitäten getroffen (vgl. [Dang09]). Auf Basis der Ergebnisse der strategischen und taktischen Planung sowie der zu erwartenden oder vorliegenden Kundenaufträge hat die operative Planung⁹ (Planungszeitraum bis zu einem Jahr) zur Aufgabe, den mengen- und zeitmäßigen Produktionsablauf unter Beachtung der verfügbaren Ressourcen bzw. Kapazitäten durch Planvorgaben festzulegen, freizugeben, zu überwachen sowie ggf. geeignete Maßnahmen bei Abweichungen zu ergreifen (vgl. [KMO+06]).

Die operative Produktionsplanung gliedert sich nach dem konventionellen Push-Planungskonzept in die sukzessiven Phasen Produktionsprogrammplanung, Mengenplanung, Losgrößenplanung, Termin- und Kapazitätsplanung sowie Reihenfolgeplanung (vgl. [Zäpf94], [GüTe00], [Helb94]). Zu Beginn werden in der Produktionsprogrammplanung für

⁵ Nach Dangelmaier stellt dies die Aufgabe dar, „*für ein abgegrenztes Produktionssystem vorausschauend Soll-daten über die qualitative, quantitative und zeitliche Zuordnung der Vorgänge und Faktoren dieses Produktionssystems, die in sich und mit den Ausgangsdaten konsistent sind, für einen definierten, zielgerichteten Leistungserstellungsprozess festzulegen (ggf. unter Berücksichtigung gewisser Ziele), diese dem Produktionssystem vorzugeben (Auftrag, Bestellung) und ggf. festzustellen, ob Inkonsistenzen vorliegen.*“ [Dang09]

⁶ Über die genaue Anzahl dieser Planungsebenen herrscht in der Literatur kein Konsens. So differenziert Stadtler nur die mittelfristige aggregierte Planung von der kurzfristigen operativen Planung (vgl. [Stad88]). Andere Autoren, wie z.B. Schneeweise (vgl. [Schn03]) beschränken sich auf die taktische und operative Ebene.

⁷ „*Der Planungshorizont stellt den Ausschnitt aus einem Kalender dar, der in einem Planungslauf gefüllt bzw. aktualisiert wird.*“ [DaWa97]

⁸ Analog zu der Anzahl der unterschiedlichen Ebenen existieren heterogene Vorstellungen hinsichtlich der Länge der einzelnen Zeitspannen. Für einen Überblick siehe [Sill08].

⁹ In der Literatur wird üblicherweise die Produktionsplanung und -steuerung (PPS) der operativen Planung gleichgesetzt (vgl. [Dang09]).

absatzbestimmte Erzeugnisse kurzfristige Primärbedarfsmengen festgelegt. Aus dem Nettobedarf können nachfolgend in der Mengenplanung der Sekundärbedarf an Vor- und Zwischenerzeugnissen berechnet werden. Der in der Mengenplanung bestimmte Nettobedarf an Vor- und Zwischenerzeugnissen wird im Anschluss in der Losgrößenplanung in Produktions- bzw. Bestellmengenlose eingeteilt. Dem folgen die Termin- und Kapazitätsplanung. In diesen werden auf Basis der Ergebnisse der Losgrößenplanung ein terminlich sowie Kapazitäten umfassender gültiger Losgrößenplan festgelegt. Zum Abschluss erfolgen in der Reihenfolgeplanung die Festlegung der Bearbeitung der Lose und eine detaillierte zeitliche Zuordnung der Lose auf die einzelnen Produktionslinien. Ein Hauptkritikpunkt an dem dargestellten Verfahren besteht darin, dass die eingesetzten Teilplanungsverfahren nicht nach einer guten ganzheitlichen Lösung suchen, sondern ihre Lösungen isoliert entwickeln. Dadurch werden Materialien, Werkzeuge und Kapazitäten nacheinander eingeplant. Da aber jeder dieser Bereiche die Voraussetzungen für die Planung der jeweils anderen Bereiche ändert, müssen Mengen und Termine im Überfluss geplant werden, um die Unsicherheit über die Wirkung der nachfolgenden Planungsschritte im Vorfeld abzufangen. Diese Überflüsse werden durch Kapazitäts-, Zeit- und Lagerpuffer organisiert (vgl. [Sche97]). Dies bedingt jedoch, dass die Verfahren in ihren berechneten Ergebnissen häufig sehr weit von der tatsächlichen Situation entfernt sind. Teilweise lassen sich gar keine zulässigen Lösungen für das Produktionsplanungsproblem finden. Um diesen Nachteilen zu begegnen, wahrt die Simultanplanung den Interdependenzcharakter und betrachtet im Idealfall sämtliche Entscheidungsvariablen gleichzeitig. Eine optimale Lösung dieser Simultanplanungsverfahren scheitert jedoch praktisch an der Komplexität, da das zugrunde liegende Entscheidungsproblem NP-vollständig¹⁰ ist (vgl. [Ross03]).

Einen Kompromiss im Hinblick auf die zu erwartende Planungsqualität bei gleichzeitig akzeptabler Realisierungswahrscheinlichkeit stellen das Sukzessivplanungsmodell mit Rückkopplung sowie das hierarchische Modell dar (vgl. [Jäge00]). Im Rahmen der Sukzessivplanung, erweitert um eine Rückkopplung, wird das Planungsproblem (z.B. im Falle von Inkonsistenzen) an einen vorgelagerten Bereich zurückverwiesen und mit verbesserten Informationen ein erneuter Planungslauf durchgeführt (vgl. [Jäge00]). Folglich werden die einzelnen Teilbereiche mehrfach durchlaufen. Im Rahmen der hierarchischen Planung wird das Gesamtproblem in Teilprobleme aufgespalten. Die einzelnen Teilprobleme werden jedoch nicht isoliert nebeneinander betrachtet, sondern vielmehr sind die Teilprobleme modellmäßig miteinander verknüpft. Im Allgemeinen handelt es sich bei den Modellen der hierarchischen Produktionsplanung um Lösungsansätze für ganz bestimmte Problemstrukturen (vgl. [Jäge00]). Einen Überblick der grundlegenden Planungsverfahren hinsichtlich ihrer Realisierungswahrscheinlichkeit sowie der Planungsgenauigkeit gibt Abbildung 1.

¹⁰ Für weitere Informationen zur NP-Vollständigkeit siehe Kapitel 2.2.5.

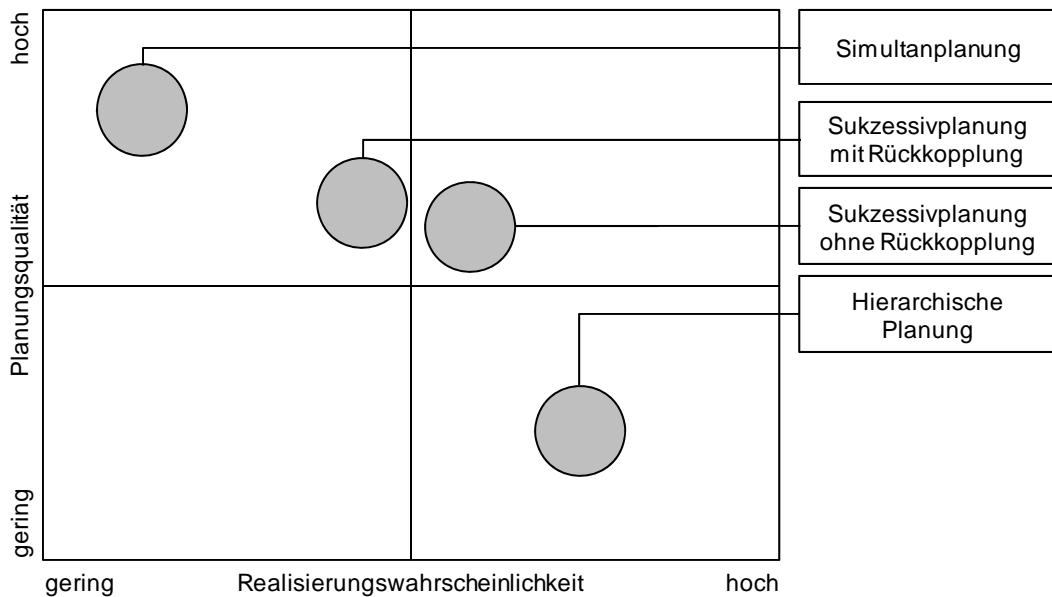


Abbildung 1: Einordnung grundlegender Planungsverfahren¹¹

2.1.2 Ablaufplanung

Die Ablaufplanung¹² „beschäftigt sich mit der zeitlichen, mengenmäßigen und räumlichen Planung des Produktionsvollzugs auf Wochen-, Tage- oder Stundenbasis“ [DoSc05]. Das Ziel ist die Generierung eines Plans, der unter Beachtung der vorhandenen Rahmenbedingungen die bei der Produktionsdurchführung entstehenden Kosten minimiert. Im Fokus stehen primär die Kosten, die bei der Fertigung (z.B. variable Lohnkosten oder Rüstkosten), der Lagerhaltung sowie im Falle einer Lieferunfähigkeit entstehen. Die Basis der Ablaufplanung stellen die Informationen der Produktionsprogramm- und Mengenplanung dar, mit denen Produktionsaufträge gebildet und eingeplant werden. Somit besteht die Gesamtaufgabe aus den Teilaufgaben der Losgrößen-, Reihenfolge- und Kapazitätsplanung.

Eine Losgröße definiert die Anzahl an „*Mengeneinheiten einer End-, Zwischen- oder Vorproduktart, die zusammengefasst werden sollen, so dass sie als ein Fertigungslos produziert beziehungsweise als eine Bestellung beschafft werden kann*“ [Kurb05]. Diese jeweilige Losgröße kann bei der Planung bereits festgelegt sein (fixierte Losgröße) oder durch die Planung determiniert werden (variable Losgröße). Die variable Losgröße kann innerhalb definierter Grenzen, die z.B. durch die vorgelagerte Produktionsstufe sowie den vorhandenen Kapazitäten vorgegeben sind, frei gewählt werden (vgl. [Puch04]).

In der Losgrößenplanung werden ein vergleichsweise grobes Zeitraster unterstellt sowie Transportvorgänge zwischen den einzelnen Arbeitsgängen und Rüstzeiten an den Maschinen in der Regel nicht explizit beachtet. In der Reihenfolgeplanung werden hingegen sämtliche zeitverbrauchenden Abläufe einbezogen. Damit besteht die Aufgabe in der Bestimmung einer Reihenfolge, „*in der N unterschiedliche Erzeugnisse auf M verschiedenen Maschinen bearbeitet werden*“ [Kurb05].

¹¹ (vgl. [Jäge00])

¹² Auch Produktionsprozess- oder Produktionsdurchführungsplanung.

beitet werden sollen, und zwar mit der Maßgabe, dass diejenige Reihenfolge gewählt wird, die einen reibungslosen, mit möglichst geringen Verlustzeiten verbundenen Produktionsablauf gewährleistet“ [Gute83]. Dabei müssen bestimmte Restriktionen, wie bspw. der Einsatz der Maschinen in der technologisch bedingten Reihenfolge, eingehalten werden. Eine besondere Relevanz kommt hierbei den Rüstzeiten zu. Werden diese nicht entsprechend berücksichtigt, können diese schnell einen hohen Anteil verfügbarer Kapazitäten in Anspruch nehmen (vgl. [KMO+06]).

Die Planung der Kapazitäten erfolgt mit der Zielsetzung, unter kapazitiven und wirtschaftlichen Kriterien die richtige Menge an Arbeitskräften zum richtigen Zeitpunkt am richtigen Ort zur Verfügung zu stellen. Dabei müssen verschiedene Zeithorizonte betrachtet werden. Im Fokus der strategischen sowie taktischen Kapazitätsplanung stehen Investitions- und Personalentscheidungen, die den langfristigen Kapazitätsrahmen für die Produktion bilden. Unter Beachtung sämtlicher Rahmenbedingungen erfolgt auf der operativen Planungsebene die kurzfristige Anpassung dieser an die jeweils aktuellen Produktionsanforderungen (auch Resourceneinsatzplanung). Das in einem Produktionsunternehmen in einem bestimmten Zeitabschnitt (z.B. für einen Monat, Woche oder Tag) zur Verfügung stehende Kapazitätsangebot wird durch ein gewähltes Schichtmodell ausgedrückt. Dieses Schichtmodell beinhaltet Informationen über arbeitsfreie Zeiten (z.B. arbeitsfreie Tage oder arbeitsfreie Zeitabschnitte je Tag) und zu einzelnen Schichten zusammengefasste Zeitabschnitte eines Tages (vgl. [Dang09]). Die Definition mehrerer Schichten an einem Tag erlaubt die Arbeit zu verschiedenen Tages- und Nachtzeiten von verschiedenen Arbeitnehmern am gleichen Arbeitsplatz ausführen zu lassen. Dadurch ist es möglich, die Kapazitäten zu erweitern, ohne die Kapazitätsstruktur technisch oder baulich zu verändern.

Bzgl. der Flexibilität werden generell zwei Arten unterschieden. Im Rahmen starrer Schichtmodelle übernehmen Arbeitnehmer nur eine bestimmte Schicht, d.h. ausschließlich in Form von Früh-, Spät- oder Nachschichten. Bei Wechselschichtmodellen (rollierende Systeme, Freischichtsysteme) werden die Arbeitnehmer hingegen in wechselnden Schichten eingesetzt. Aufgrund dieser Variationsmöglichkeit wird dies auch als ein flexibles (offenes) Arbeitssystem bezeichnet. Aus den Wechselschichtmodellen sind alle Kombinationen aus Schichten mit oder ohne Nacht- und Wochenendarbeit¹³ möglich, wie z.B. einfache unregelmäßige Schichtmodelle oder regelmäßige Schichtmodelle (wie einfache Zwei- oder Dreischichtmodelle) (vgl. [Wild91]). Erstere variieren bspw. in der Anzahl der Arbeitskräfte innerhalb einer Schicht. Die Entscheidungsgrundlage für den Personalbedarf bildet der tageszeitabhängige Arbeitsanfall.

Zur Erstellung eines Schichtmodells für einen bestimmten Zeitabschnitt müssen verschiedene Faktoren berücksichtigt werden. Diese umfassen bspw. die Bestimmung der erforderlichen Menge an Arbeitskräften zu jeweiligen Zeitabschnitten, die Bestimmung geeigneter Schichten sowie die Zuordnung von einzelnen Arbeitskräften zu den Schichten unter Beachtung von gesetzlichen Bestimmungen (z.B. einer Obergrenze erlaubter Stunden pro Woche oder Tag,

¹³ Systeme mit Nacharbeit ohne Wochenendarbeit werden als diskontinuierliche Arbeitsweise bezeichnet, wohingegen bei einer kontinuierlichen Arbeitsweise auch am Wochenende gearbeitet wird (vgl. [Wild91]).

einer maximal erlaubten Anzahl von Nachschichten oder Ruhezeiten zwischen zwei aufeinander folgenden Schichten) sowie individuellen Faktoren (z.B. der Qualifikation oder das Alter). Eine simultane Berücksichtigung aller Faktoren stellt jedoch ein sehr komplexes Problem dar, weswegen es in der Praxis häufig dekomponiert wird. So erfolgt bspw. häufig die Zuordnung von Arbeitskräften in einem separaten Schritt.

Häufig werden in der Praxis diese Teilaufgaben isoliert behandelt, wodurch reziproke Interdependenzen unberücksichtigt bleiben und ungültige oder nicht optimale Pläne entstehen. Im Folgenden werden die einzelnen Abhängigkeiten zwischen der Losgrößen-, Reihenfolge- und Kapazitätsplanung, die auf einer Produktionsstufe existieren, diskutiert:

- Losgrößenplanung ↔ Reihenfolgeplanung: Von der Wahl der Losgröße hängen eine Reihe von Kostenarten ab. Es existieren Kosten, die proportional zur Menge steigen (z.B. Lagerhaltungskosten) und jene die mit steigender Menge sinken (bspw. Rüstkosten). Um die Rüttvorgänge und die damit verbundenen Rüstkosten, wie die kostenverursachende Rüstzeit, die benötigten Arbeitskräfte, Material, usw., möglichst gering zu halten, sind entsprechend große Lose zu produzieren. Falls jedoch die Produktion der großen Lose keinen entsprechend schnellen Absatz findet, führt dies zu Kosten, z.B. für entgangene Zinsen für das gebundene Kapital im Lager, die benötigten Lagerarbeiter, etc. Somit besteht das Problem in dem Finden einer Losgröße, die unter Beachtung der konträren Kostenarten die Gesamtkosten minimiert (vgl. [KMO+06]). Die Ermittlung und Optimierung der Rüstkosten (besonders im Falle reihenfolgeabhängiger Rüstkosten) erfolgt jedoch im Rahmen der Reihenfolgeplanung. Sofern keine Rüstkostenzeiten vorliegen, kann eine Produktion nahe dem Bedarfstermin der einzelnen Aufträge realisiert werden. Eine Zusammenfassung von einzelnen Bedarfen zu größeren Losen ist dann nicht erforderlich. Für die Reihenfolgeplanung sind auf der anderen Seite Informationen über die von der Losgrößenplanung erstellten Produktionsmengen wichtig, um eine Zuordnung und eine Reihenfolge herstellen zu können.
- Losgrößenplanung ↔ Kapazitätsplanung: Die Losgröße definiert die Anzahl der zusammenhängend zu produzierenden Erzeugnisse. Diese hängt ggf. maßgeblich von der zur Verfügung stehenden Kapazität ab. Werden keine Kapazitätsrestriktionen berücksichtigt, könnten durch das Zusammenfassen von Bedarfen Lose entstehen, die nach der Zuordnung der maximal realisierbaren Kapazität eine fristgerechte Produktion aller in dem Planungshorizont vorhandenen Bedarfe verhindern. Erfolgt dagegen zuerst die Kapazitätsplanung ohne Berücksichtigung des konkreten Kapazitätsbedarfs, kann dies zu einer Kapazitätsunterdeckung oder einem Kapazitätsüberschuss führen.
- Reihenfolgeplanung ↔ Kapazitätsplanung: Zur Bestimmung der Starttermine und der damit verbundenen Reihenfolge der Lose müssen genaue Informationen über die vorhandenen Kapazitäten vorliegen. Eine Variation der Kapazitäten (Leistungsgrad¹⁴ und Arbeitszeiten) kann jedoch auch eine Variation der Bearbeitungsdauer von Arbeits- und Rüttvorgängen implizieren. Aus diesem Grund ist die Kenntnis über die Reihenfolge der Lose (besonders bei reihenfolgeabhängigen Rüstzeiten) und dessen Bedarfsterminen notwendig.

¹⁴ Der Leistungsgrad ist ein Maß der Inanspruchnahme eines outputabhängig disponierbaren Kapazitätsangebots (vgl. [Dang09]). Häufig werden „Intensität“ oder „Produktionsgeschwindigkeit“ synonym verwendet.

Neben diesen Abhängigkeiten auf einer Produktionsstufe existieren in einem mehrstufigen Produktionssystem weitere zwischen den Produktionsstufen. So dienen die Outputfaktoren einer Produktionsstufe der Nachfolgerproduktionsstufe als Input. Werden nun auf einer Produktionsstufe Änderungen vorgenommen, kann dies Auswirkungen auf alle vor- und nachgelagerten Produktionsstufen implizieren. Werden diese Interdependenzen auf einer feineren Granularitätsebene analysiert, so finden sich die beschriebenen Abhängigkeiten nicht nur auf einer Produktionsstufe, sondern jeweils auch bezogen auf sämtliche anderen Teilplanungsschritte sämtlicher Produktionsstufen, wieder.

2.1.3 Arten der Produktionsplanung

2.1.3.1 Prädiktive Produktionsplanung

In der prädiktiven Ablaufplanung wird ein Plan vorausschauend für einen bestimmten Zeitabschnitt unter der Annahme einer statischen Planungsumgebung (alle planungsrelevanten Informationen sind bekannt und ändern sich nicht im Zeitablauf) erstellt. Dabei wird unter Einhaltung vorgegebener Restriktionen eine gegebene Zielsetzung optimiert bzw. bestmöglich erfüllt. Verfolgt werden primär zeit-, ressourcen- und kostenorientierte Ziele (vgl. [Sau02]).

Die Forschung hat diesen Bereich der Ablaufplanung bereits intensiv behandelt und eine Vielzahl an Lösungsverfahren entwickelt (siehe auch Kapitel 3). Unter praxisrelevanten Bedingungen hat sich die prädiktive Ablaufplanung jedoch als ungeeignet erwiesen. So muss bei einer geänderten Ausgangslage jeweils eine vollständige oder teilweise Neuplanung erfolgen. Dies ist insbesondere bei einem mehrstufigen Produktionssystem aufgrund der Interdependenzen zwischen den einzelnen Produktionsstufen schwierig.

2.1.3.2 Reaktive Produktionsplanung

Die reaktive Produktionsplanung unterstellt eine dynamische Planungsumgebung, in der die der Planung zugrundeliegenden Daten jederzeit geändert werden können. Weicht eine solche Änderung von einem gegebenen Toleranzspektrum ab, muss der Produktionsplan an die neue Situation angepasst werden. Das Auftreten eines solchen Ereignisses wird im weiteren Verlauf dieser Arbeit als Störung bezeichnet.

Störungen können generell nach sachlichen und zeitlichen Aspekten unterschieden werden. Die sachlichen Aspekte differenzieren Störungen nach der Art, der endogenen oder exogenen Ursache und der Wirkung. Einen Überblick stellt Patig (vgl. [Pati01]) in Tabelle 1 dar.¹⁵

¹⁵ Eine ähnliche Kategorisierung von Störfaktoren wird von Aytug et. al. vorgenommen (vgl. [ALM+05]). Die Autoren differenzieren Ursache, Wirkung und Kontext.

von Abweichungen betroffene Plan-Werte	Störungswirkung	Exemplarische Störungsursache
Sachzielstörungen		
Bedarfsmenge oder Endtermin eines Fertigungsauftrags oder Anzahl von Fertigungsaufträgen	Änderung der Kundenbedarfe (Eilaufträge, Stornierungen)	- Ursachen dem Unternehmen unbekannt
	Änderung der Bedarfe unternehmensinterner Abteilungen	- Ersatzteile, Zwischen- oder Endprodukte für Testzwecke
	Nacharbeit, Ausschuss	- Fehlleistungen der Potentialfaktoren - Fehlerhafte eingesetzte Repetierfaktoren
Potentialfaktorstörungen		
Potentialfaktoren für Arbeitsgänge von Fertigungsaufträgen	Ausfall von Potentialfaktoren	- Technische (elektrische, pneumatische, mechanische usw.) Defekte von Betriebsmitteln - Krankheit von Arbeitskräften
Repetierfaktorstörungen		
Sekundärbedarfe von Fertigungsaufträgen	Bereitstellungsverzug	- Planungsfehler, Schwund, Transportprobleme, Potentialfaktorausfall
	Falschlieferung	- Organisationsmängel
	Qualitätsmängel	- Fehlerhafte Repetierfaktoren - Fehlleistungen der Potentialfaktoren
Stammdatenstörungen		
Arbeitsgänge oder Sekundärbedarfe von Fertigungsaufträgen	Arbeitsplanänderung	- Fehlerhafte Vorgabezeiten - Produktmängel
	Stücklistenänderung	- Produktmängel

Tabelle 1: Störungsklassen und -ursachen¹⁶

Die zeitlichen Aspekte betrachten den Eintritt der Störungsursache sowie das Aufkommen und die Dauer der Störungswirkung. Störungen sind Abweichungen der Ist- von den Plan-Werten innerhalb der geplanten Losgrößen, der Bearbeitungsstart- und -endtermine der Fertigungsaufträge und der auszuführenden Arbeitsschritte, denen Kapazitäten, Zeiten und Materialien zugewiesen sind. Zudem kann die Störung ein Ereignis darstellen, das eine Abweichung der Ist- von den Plan-Werten verursacht (vgl. [Pati99]). Sofern bereits vor dem Störungseintritt die alten durch neue Plan-Werte ersetzt wurden, kann dies als Störung interpretiert werden (vgl. [Pati01]). Im Allgemeinen beeinflussen Störungen den Erfolg eines Unternehmens negativ. Eine Störung kann sich jedoch auch positiv auf den Unternehmenserfolg auswirken (z.B. ein außerplanmäßiger Eilauftrag) (vgl. [Pati99]).

¹⁶ (vgl. [Pati01])

Treten in einem Unternehmen Störungen auf, sind mit der Behandlung dieser unterschiedliche ökonomische Effekte verbunden. Einen Überblick gibt Abbildung 2.

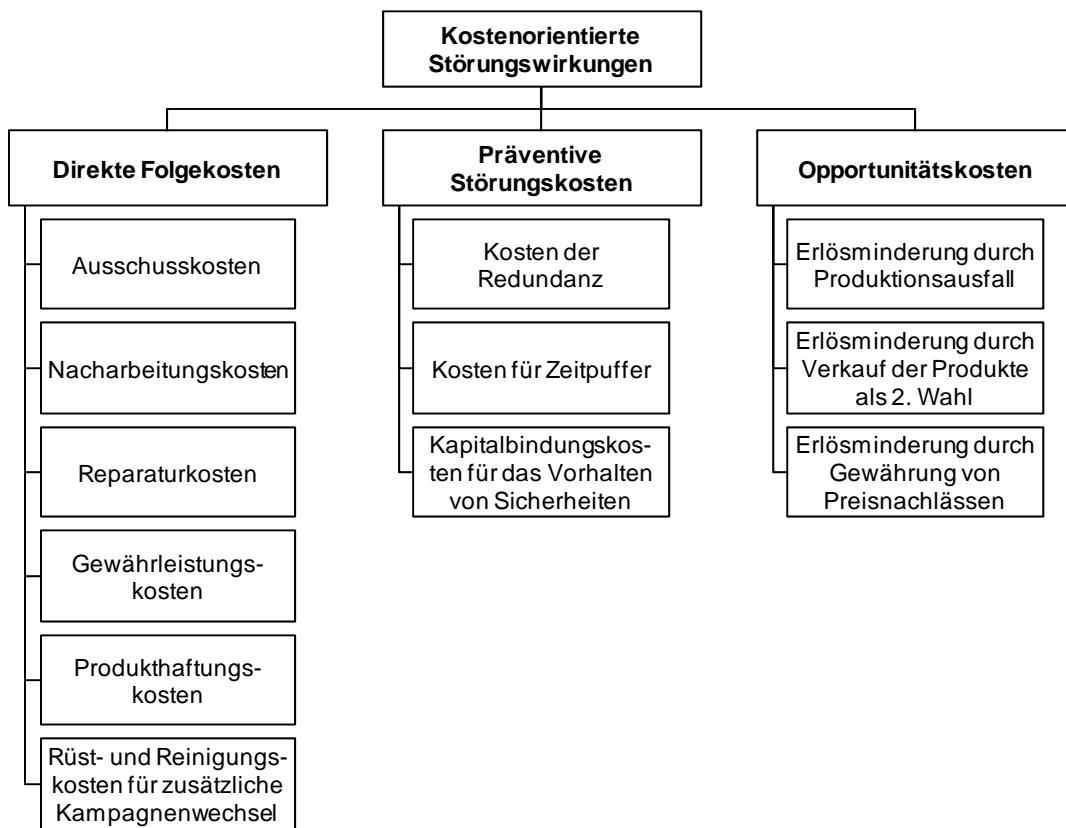


Abbildung 2: Kostenorientierte Störungswirkungen¹⁷

In Abbildung 2 wird deutlich, dass eine Vielzahl unterschiedlicher Kostenarten existieren. Besonders vor dem Hintergrund des stetigen steigenden Einsatzes kostensparender Produktionskonzepte (z.B. Just-in-Time, Just-in-Sequenz), die jedoch auch den Raum einer Störung entgegenzuwirken stark reduzieren, gewinnt ein guter Umgang mit Störungen essentiell an Bedeutung. So nennt Sauer drei Kriterien, die bei der Anpassung durch Störungen ungültig gewordener Pläne beachtet werden müssen (vgl. [Sau02]).

- Die Reaktion auf die Änderungen soll möglichst schnell erfolgen, ohne dabei jedoch ein übernervöses Verhalten zu verursachen. D.h., es muss nicht auf jede einzelne Störung sofort reagiert werden, weil sonst der Plan einer ständigen Umplanung unterworfen wäre.
- Die gesamte Güte des Plans soll sich möglichst nicht verschlechtern.
- Der bestehende Plan soll möglichst wenig geändert werden, da evtl. bereits Vorbereitungen getroffen oder Arbeitsschritte begonnen wurden.¹⁸

¹⁷ (vgl. [Frei99])

¹⁸ Dies ist insbesondere bei mehrstufigen Produktionssystemen relevant, da hier aufgrund der Interdependenzen zwischen den Produktionsstufen durch Änderungen auf einer Produktionsstufe die Pläne für die anderen Produktionsstufen ungültig werden können.

Zum Umgang mit Störungen werden generell zwei unterschiedliche Strategien unterschieden. Im Rahmen der dynamischen Planung¹⁹ werden keine längerfristigen Produktionspläne erstellt, sondern es kommen dezentralisierte Steuerungsmethoden (z.B. Prioritätsregeln) zum Einsatz, die bei Bedarf auf Basis der aktuell vorliegenden Informationen die Aufträge einplanen (vgl. [VHL03]). Im Rahmen dieser Arbeit liegt der Fokus hingegen auf der zweiten Strategie, bei der es sich um die reaktive Planung²⁰ handelt. Diese besteht aus zwei Schritten. In einem ersten Schritt wird für den Planungshorizont ein gültiger (prädikativer) Produktionsplan erstellt, der im Falle von planungsrelevanten Störungen in einem zweiten Schritt zu adaptieren ist. Das Ziel der prädiktiven Planung besteht damit in der Ermittlung eines initialen Belegungsplans, auf dessen Basis externe Aktivitäten (z.B. die Materialbeschaffung) abgestimmt werden können. Die reaktive Planung verfolgt hingegen als Zielsetzung die Gültigkeit eines aktuellen Plans, ohne eine maßgebliche Verschlechterung, wiederherzustellen (vgl. [Neuh08]). Hierzu gibt es mehrere Strategien, deren ausführliche Beschreibung in Kapitel 3.2.3 folgt.

2.2 Der Flexible Flow Shop

Die Problemklasse des Flexible Flow Shop²¹ liegt in der Praxis sehr häufig vor. Beispiele finden sich in der Automobil-, Chemie-, Kosmetik-, Elektronik-, Lebensmittel-, Papier-, Pharmazie- sowie in der Textilbranche (vgl. [Quad04]). Die konkrete Ausgestaltung eines Flexible Flow Shop ist damit vom jeweiligen Produktionssystem abhängig. Nachfolgend wird der in dieser Arbeit unterstellte Flexible Flow Shop im Detail beschrieben. Es werden der generelle Aufbau des Produktionssystems, das zugrundliegende Zeitmodell sowie die Planungspolitik erläutert. Darauf aufbauend wird anschließend das Zielkriterium definiert.

2.2.1 Aufbau des Produktionssystems

Der Aufbau des Flexible Flow Shop ist in Abbildung 3 dargestellt.

¹⁹ Von manchen Autoren auch als „on-line scheduling“ oder „real-time scheduling“ bezeichnet (vgl. [VHL03]).

²⁰ Von manchen Autoren auch als predictive-reactive scheduling bezeichnet (vgl. [VHL03]).

²¹ Ein Flexible Flow Shop stellt eine Verallgemeinerung des Flow Shop dar [BEW+01]. Es besteht wie der klassische Flow Shop aus $|PS| \geq 2$ Produktionsstufen in Serie, wobei mindestens einer Produktionsstufe ps mehr als eine Produktionslinie ($\exists PL_{ps}: |PL_{ps}| \geq 2$) zugeordnet ist. Aus diesem Grund werden für den Begriff des „Flexible Flow Shop“ in der Literatur auch „Flow Shop mit parallelen Maschinen“, „Flexible Flow Line“ oder „Hybrid Flow Shop“ verwendet.

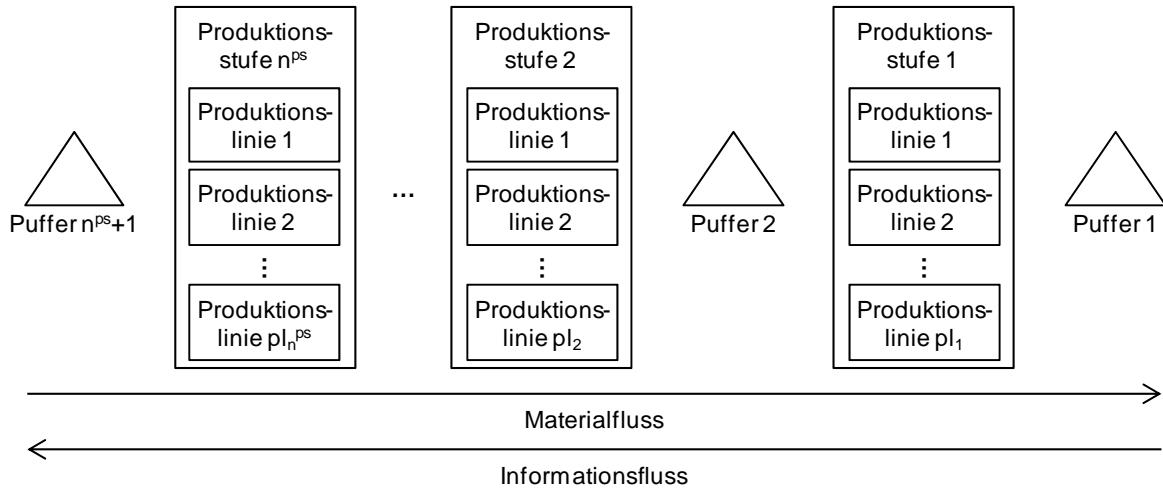


Abbildung 3: Allgemeiner Flexible Flow Shop

Das Produktionssystem besteht aus mehreren Produktionsstufen ($PS = \{1, \dots, n^{ps}\}$) in Serie²². Jede Produktionsstufe ($ps \in PS$) kann aus einer oder verschiedenartigen parallelen Produktionslinien²³ bestehen ($PL_{ps} = \{1, \dots, pl_{ps}\}$). Damit ist es möglich, dass jede dieser Produktionslinien über eine unterschiedliche qualitative Eignung zur Produktion verschiedener Erzeugnisse sowie einer quantitativen Kapazität verfügen kann. Am Anfang und am Ende des Produktionsprozesses sowie zwischen zwei aufeinanderfolgenden Produktionsstufen existieren Puffer ($PU = \{1, \dots, pu\}$) zur Lagerung der Erzeugnisse. Jeder Puffer verfügt über eine begrenzte Kapazität (pf_{pu}^{max}). Befindet sich ein Puffer zwischen zwei Produktionsstufen, kann dieser der Resequenzierung dienen. In diesem Fall wird auf der nachgelagerten Produktionsstufe in einer anderen Reihenfolge produziert als auf der vorgelagerten Produktionsstufe.

Auf diesem Produktionssystem sind eine Menge unterschiedlicher Aufträge zu bearbeiten. Ein Auftrag besteht aus n^{ps} Operationen, deren Ausführung einem fest definierten zyklenfreien

²² Die Nummerierung der Produktionsstufen folgt der Richtung des Informationsflusses. Damit ist bspw. die erste Produktionsstufe der Versand und die letzte Produktionsstufe die Teilefertigung.

²³ Es wird generell zwischen einzelnen (single) und parallelen (parallel) Produktionslinien (Maschinen und Produktionslinien werden in der Literatur synonym verwendet) unterschieden. In der erst genannten Unterscheidung müssen alle Erzeugnisse von einer Produktionslinie bearbeitet werden. Eine Verallgemeinerung stellt ein Produktionssystem mit mehreren Produktionslinien dar. Diese können nach multi und parallelen Produktionslinien differenziert werden. Im ersten Fall existiert eine feste Erzeugnis-Produktionslinie-Zuordnung, d.h., ein Erzeugnis lässt sich nur auf einer bestimmten Produktionslinie herstellen. Die Produktionslinien verfügen folglich über ein unterschiedliches qualitatives Leistungsvermögen. Im Gegensatz dazu erlauben parallele Produktionslinien einen größeren Freiraum in der Planung. Mehrere Produktionslinien, die gleiche Operationen ermöglichen, stehen für die Produktion des Erzeugnisses zur Verfügung, d.h., ein Erzeugnis kann von beliebigen zur Verfügung stehenden Produktionslinien hergestellt werden. Eine Unterscheidung kann zusätzlich hinsichtlich des quantitativen Leistungsvermögens getroffen werden. In diesem Kontext existieren identische, gleichartige und verschiedenartige parallele Produktionslinien. Erst genannte (identical machines) verrichten die gleichen Operationen in der gleichen Bearbeitungszeit. D.h., dass die Zeit, die ein Erzeugnis auf einer Produktionslinie zur Bearbeitung benötigt, unabhängig ist von der jeweiligen Produktionslinie. Unterscheiden sich die Bearbeitungszeiten der Produktionslinien voneinander, liegen gleichartige parallele Produktionslinien vor. Verschiedenartige parallele Produktionslinien stellen eine Verallgemeinerung von gleichartigen parallelen Produktionslinien dar. Bei dieser Ausprägung ist die Bearbeitungszeit abhängig von einer jeweiligen Produktionslinie sowie dem zu bearbeitenden Erzeugnis.

Prozess folgt. Dieser beginnt auf der in Materialflussrichtung ersten Produktionsstufe und endet nach der letzten Produktionsstufe. Auf einer Produktionsstufe kann erst mit einer Operation begonnen werden, wenn die Ausführung der vorherigen Operation abgeschlossen ist sowie die Vorlaufzeit (vlz_e , z.B. Transportzeiten zwischen zwei Produktionsstufen) berücksichtigt wurde. Eine zeitliche Überlappung der einzelnen Operationen ist nicht zulässig. Nachdem mit der Bearbeitung einer Operation begonnen wurde, wird diese fertig gestellt. Eine Unterbrechung von Operationen zugunsten einer anderen höher priorisierten Operation ist folglich nicht erlaubt. Das Ergebnis einer jeweiligen Operation ist ein Erzeugnis e ($e \in E = \{1, \dots, e^{ges}\}$). Demnach ist jedes Erzeugnis einer Produktionsstufe zugeordnet. Ein Erzeugnis kann von minimal einer und maximal pl_{ps} Produktionslinien bearbeitet werden. Die Stückbearbeitungszeit ($tb_{e,pl}$) entspricht der Zeit zur Fertigstellung einer Einheit des Erzeugnisses e auf der Produktionslinie pl . Findet auf einer Produktionslinie pl ein Wechsel zwischen Erzeugnissen statt, fallen für das Umrüsten von Erzeugnis i auf Erzeugnis j reihenfolge- und linienabhängige Rüstzeiten $rz_{i,j,pl}$ an. Analog zur Stückbearbeitungszeit bezieht sich die Rüstzeit auf eine Produktion mit dem Leistungsrad 1 (100%). Ein Erzeugnis ist zudem genau einem Puffer zugeordnet. Zur Berücksichtigung unterschiedlicher erzeugnisspezifischer Anforderungen (z.B. Größe, Art der Lagerung) an einen Puffer dient ein Kapazitätsfaktor pro Stück je Erzeugnis (kf_e).

2.2.2 Zeitmodell

Die zeitliche Betrachtung erfolgt durch die Definition zweier Zeitmengen mit unterschiedlicher Granularität. Die grobe zeitliche Struktur wird durch die Menge $T = \{1, \dots, n\}$ abgebildet. Diese unterteilt den Planungshorizont in n Makroperioden. Eine Makroperiode t entspricht dabei z.B. 24 Stunden. Die Verfeinerung der zeitlichen Struktur erfolgt durch die Menge $M = \{1, \dots, |T| * MM\}$. Sie teilt den Planungshorizont in $|T| * MM$ Mikroperioden ein, wobei MM die Anzahl Mikroperioden pro Makroperiode ist. Eine Mikroperiode kann z.B. eine Minute sein.

Durch die Zusammenfassung aufeinanderfolgender Mikroperioden können unterschiedliche Schichten s definiert werden. Hierbei darf die Summe der Mikroperioden aller Schichten einer Makroperiode nicht die Länge einer Makroperiode übersteigen. Zur Planung der Kapazitäten wird einer Produktionslinie für jede Makroperiode ein Schichtmodell sm zugewiesen (vgl. Abbildung 4). Ein Schichtmodell wird durch die Anzahl und Länge der eingeplanten Schichten definiert²⁴. Es gilt $sm_{pl,t} = \{s_1, \dots, s_l\}$. Die für eine Makroperiode zur Verfügung stehenden Schichtmodelle sind abhängig von einer Produktionslinie pl und der Makroperiode t . Dadurch kann ausgedrückt werden, dass an Werktagen anders gearbeitet wird als an Sonn-

²⁴ In dieser Arbeit wird von einer begrenzten Verfügbarkeit und unterschiedlichen Qualifikationen von Arbeitskräften abstrahiert. Es wird unterstellt, dass die Bereitstellung entsprechender Umfänge in der taktischen Planung sichergestellt wird bzw. Mindermengen kurzfristig durch externe Personaldienstleister ausgeglichen werden können. Die Zuordnung von Arbeitskräften zu Arbeitszeiten (Schichten) erfolgt im Rahmen einer in dieser Arbeit nicht weiter betrachteten Personaleinsatzplanung. Innerhalb einer abgeschlossenen Schicht wird zudem eine gleichbleibende Kapazität unterstellt. D.h., dass keine Kapazitätsverluste durch z.B. den Anlauf der Produktion (z.B. 20% weniger Output in den ersten x Mikroperioden) betrachtet werden.

oder Feiertagen. Ein jedes Schichtmodell umfasst genau die Anzahl enthaltender Mikroperioden. Für jede Makroperiode und jede Produktionslinie muss es ein Schichtmodell geben, das allen Mikroperioden den Leistungsgrad 0 zuweist. Das Schichtmodell repräsentiert den Fall, dass in dieser Makroperiode auf dieser Produktionslinie nicht gearbeitet wird. Die Wahl eines Schichtmodells sm für die Makroperiode t und die Produktionslinie pl verursacht Kosten $sk_{sm,pl,t}$.

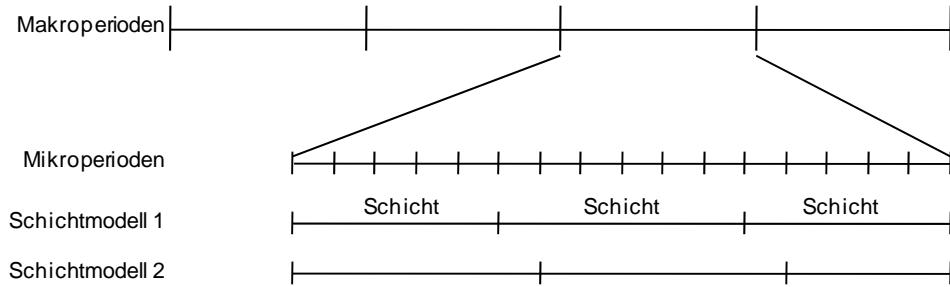


Abbildung 4: Aufbau eines Schichtmodells

Jede Mikroperiode μ eines Schichtmodells sm und einer Produktionslinie pl hat einen Leistungsgrad $lg_{sm,pl,\mu}$. Der Leistungsgrad beschreibt dabei die Arbeitsintensität im Vergleich zu einer Standardleistung und ergibt sich aus dem quantitativen Einsatz personeller Ressourcen. Im einfachsten Fall wird zwischen zwei Ausprägungen des Leistungsgrads unterschieden. Wird in einer Mikroperiode gearbeitet, verfügt diese über den Leistungsgrad 1. Ist diese Mikroperiode unproduktiv, hat sie den Leistungsgrad 0. Optional kann dieser Leistungsgrad weiter verfeinert werden. Mit abnehmendem Leistungsgrad verlängert sich entsprechend die tatsächliche Bearbeitungszeit. Ausgedrückt wird dieser Sachverhalt durch die Formel (2.1):

$$tb_{e,pl}^{temp} = \frac{tb_{e,pl}}{lg_{sm,pl,\mu}}, lg_{sm,pl,\mu} > 0 \quad (2.1)$$

Der Leistungsgrad und die Dauer der Schichten können jedoch keine beliebigen, sondern nur vorher fest definierte Ausprägungen annehmen. Es ist zu beachten, dass die Angabe eines solchen Leistungsgrads optional ist und von der konkreten Produktionslinie abhängt.

Der Aufbau der verschiedenen Schichtmodelle wird in Abbildung 5 anhand eines Beispiels veranschaulicht. In einem Unternehmen können der Produktionsstufe „Qualitätskontrolle“ Schichten mit einer Länge von 8 und 10 Stunden zugeordnet werden. Aufgrund der technischen Ausstattung und der Art der Arbeit ist es zudem möglich, den Leistungsgrad zu variieren. Wählbar sind 0, 0,5 und 1. Aufgrund von Vereinbarungen mit dem Betriebsrat sind zudem an Sonn- und Feiertagen nur 10 Stunden Schichten zulässig. Daraus ergeben sich sowohl Schichtmodelle mit nur geringer Kapazität (z.B. einer 10 Stunden Schicht mit einem Leistungsgrad von 0,5) als auch hoher (z.B. drei 8-Stunden Schichten mit 1).

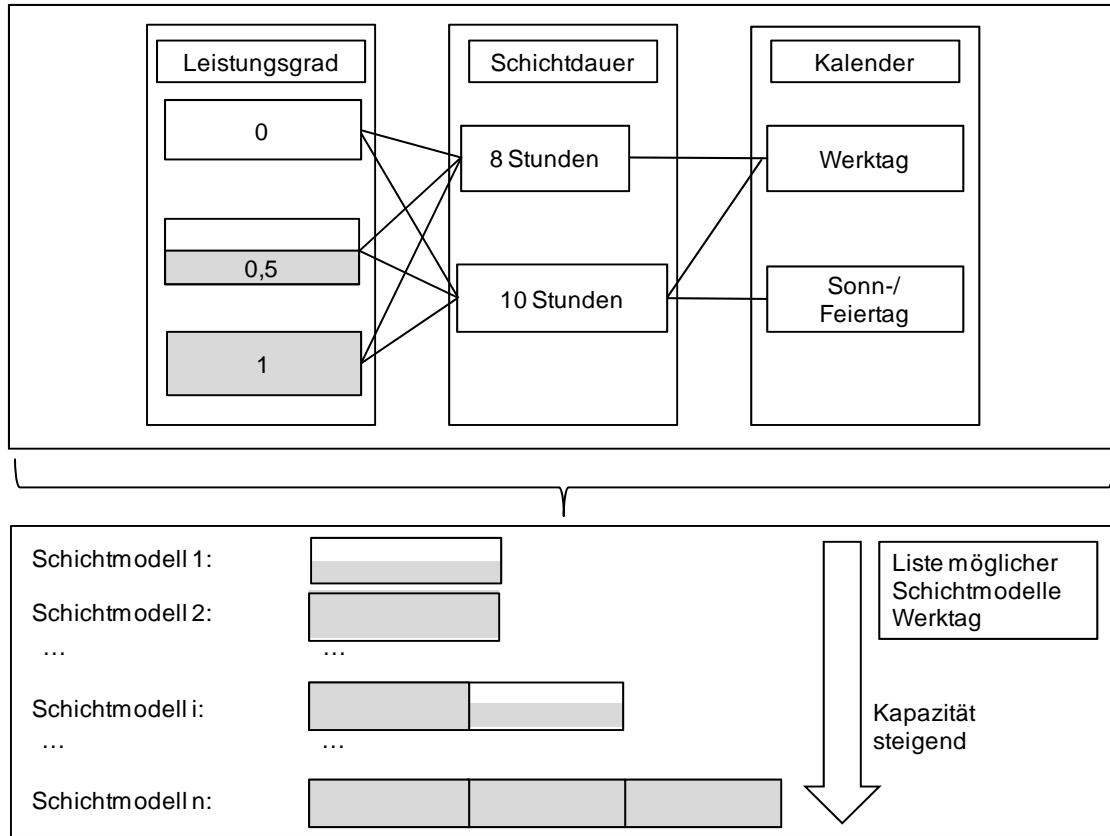


Abbildung 5: Menge aller gültigen Schichtmodelle

2.2.3 Planungspolitik

Die Fertigung der Erzeugnisse erfolgt bedarfsorientiert, wobei der Umfang zu produzierender Erzeugnisse auf externen Aufträgen (von Kunden) oder internen Aufträgen (Fertigung von Zwischenerzeugnissen) basiert. Der Bedarfstermin $b_{e,\mu}$ beschreibt die benötigte Menge eines Erzeugnisses e in der Mikroperiode μ . Damit setzt sich ein konkreter Auftrag²⁵ aus einer Kennung für ein bestimmtes Erzeugnis, dem Bedarfstermin und der Bestellmenge zusammen.

Im Kapitel 2.1.3.1 erfolgte bereits eine Unterscheidung zwischen einem statischen und dynamischen Planungsumfeld. In dem erst genannten wird für einen definierten Planungshorizont die Planung vorgenommen, wobei die Bedarfe keinerlei Veränderungen unterliegen. Diese Annahme liegt jedoch in der Praxis selten vor. So sind Veränderungen der internen oder externen Bedarfe sowie der Kapazitätsangebote (z.B. Stornierungen, Eilaufträge, Maschinenausfälle) allgegenwärtig. Dieses Problem verschärft sich besonders bei einer zunehmenden Länge des Planungshorizontes. Hierbei werden die Daten und die darauf basierenden Entscheidungen immer unsicherer. Zudem steigt der Planungsaufwand. Um diesen Aufwand zu reduzieren und eine möglichst große Aktualität der Planungsergebnisse zu gewährleisten, wird in dieser Arbeit die rollierende Planung genutzt (siehe Abbildung 6). In dieser wird nur ein begrenzter Planungshorizont beplant. In festgelegten Intervallen (jeweils einer Makroperiode) wird je-

²⁵ In dieser Arbeit werden Aufträge nur auf bestehenden internen oder externen Bedarfen erstellt. Eine stochastische Ermittlung von Bedarfen wird damit nicht integriert.

weils diese Planung fortgeschrieben und die übrigen Daten (z.B. geänderte Bedarfe) aktualisiert. Nach Recker können mit Hilfe dieser Planung „sowohl die zeitlichen Interdependenzen einer Problemstellung als auch die Unsicherheit der Problemdaten angemessen“ [Reck00] berücksichtigt werden. Innerhalb eines Planungszeitraums sind dabei nicht sämtliche Entscheidungen verbindlich zu treffen, sondern lediglich die erste Periode festzulegen, um die restlichen vorläufig planen zu können. Ein wesentliches Merkmal der rollierenden Planung besteht darin, dass die Daten umso geringere Qualitätsanforderungen stellen, je weiter die betroffene Periode in der Zukunft liegt.

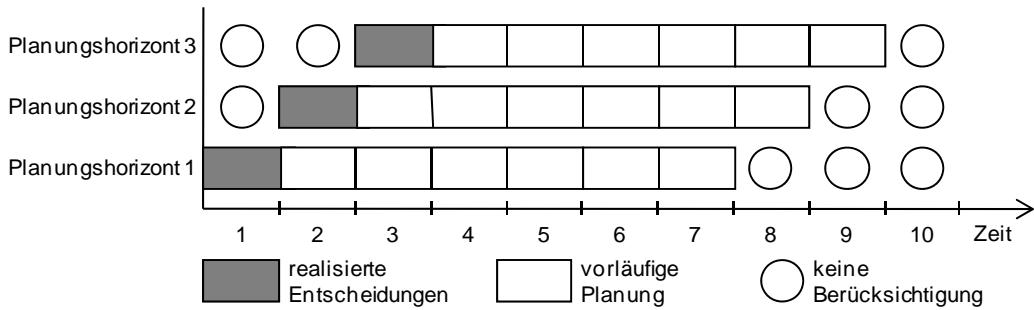


Abbildung 6: Ablauf einer rollierenden Planung

Die rollierende Planung weist jedoch auch Nachteile auf. Durch eine starre Einhaltung der Planungszeitpunkte besteht die Gefahr, dass die in einem dynamischen Planungsumfeld potentiellen Störungen (Eilauftrag, Ausfall einer Produktionslinie, etc.) verspätet berücksichtigt und dadurch nicht optimale Entscheidungen getroffen werden. Zudem können regelmäßige Planänderungen zu Akzeptanzproblemen bei den Arbeitskräften und damit verbundenen ablauforganisatorischen Schwierigkeiten, mit ggf. einer Erhöhung der Durchlaufzeiten oder Lagerbestände, führen. Trotz dieser Nachteile hat sich dieses Vorgehen in der Praxis als sinnvoll erwiesen und es wurde in mehreren Untersuchungen die Überlegenheit der rollierenden Planung z.B. gegenüber (kurzsichtigen) Prioritätsregeln nachgewiesen (vgl. [OvUz94]).

2.2.4 Zielsetzung

Bei der Planung eines Flexible Flow Shop werden zumeist zeitbezogene und monetäre Zielsetzungen verfolgt. Zeitbezogene Zielsetzungen beziehen sich typischerweise auf eine Minimierung der Gesamtdurchlaufzeit bis zur Fertigstellung aller Aufträge oder der Minimierung der mittleren Durchlaufzeit (vgl. [BDH97]). Daneben gibt es noch weitere Ziele, wie z.B. die mittlere Verspätung aller Aufträge oder die maximale Verspätung eines Auftrags (vgl. [VHL03], [Wang05]). Die zeitbezogenen Ziele spiegeln allerdings nicht die ökonomische Leistung eines Produktionssystems vollständig wider. Da diese jedoch gerade in der Praxis von entscheidender Bedeutung ist, haben verschiedene Autoren einen entsprechenden Faktor vorgeschlagen. Sie bewerten einen Produktionsplan u.a. anhand der Auftragsrentabilität, der in Bearbeitung befindlichen Erzeugnisse (work in progress) sowie der Gesamtkosten (vgl. [VHL03], [ShBr99b], [ShBr99a]).

Diese monetäre Zielsetzung wird auch in den in dieser Arbeit vorgestellten Verfahren verfolgt. In den Planungsverfahren erfolgt die Minimierung dreier Kostenarten. Dies sind Pro-

duktionskosten, Lagerkosten und Verzugskosten. Die Produktionskosten werden durch die Kosten der jeweils allokierten Schichtmodelle determiniert. Deren Wahl hängt von den einer Makroperiode zugeordneten Losen sowie der gebildeten Losreihenfolge ab. Wird eine ungünstige Reihenfolge gewählt, entsteht ein höherer Kapazitätsbedarf, der ein Schichtmodell mit höheren Kapazitäten und damit entsprechend höheren Kosten impliziert.²⁶ Zudem ist auch ein Abgleich zwischen Lagerkosten und Produktionskosten vorzunehmen. So kann es z.B. günstiger sein, für Montag früh fällige Bedarfe schon am Freitag in einer billigeren Schicht zu produzieren als am Sonntag. Werden Erzeugnisse vor dem Bedarfstermin gefertigt, entstehen Lagerkosten. Diese Kosten lk_e werden pro Erzeugnis e für eine Einheit und einer Mikroperiode beziffert. Werden hingegen Erzeugnisse verspätet gefertigt, sind Verzugskosten vk_e zu fakturieren. Diese beziehen sich auf eine Einheit des Erzeugnisses pro Mikroperiode.

Neben diesen zeitbezogenen und monetären Zielsetzungen sind im Kontext dynamischer Produktionssysteme zudem die Robustheit, Flexibilität sowie Stabilität relevant (vgl. [Neuh08]). Die Robustheit beschreibt die Unempfindlichkeit eines Plans im Falle von Störungen. Unterschieden werden die Zulässigkeits- und Optimalitätsrobustheit. Die Zulässigkeitsrobustheit beschreibt die Fähigkeit eines Plans, für jede potentielle Störung unverändert oder durch einfache Anpassungsmaßnahmen ausgeführt zu werden. Ein Plan gilt als optimalitätsrobust, wenn die Zielgröße des Plans im Falle von Störungen nicht von dem für das Szenario ermittelten optimalem Wert abweicht. Die Flexibilität kennzeichnet die Anpassungsfähigkeit eines Plans. Ein Plan gilt als flexibel, wenn er durch wenige Umplanungsaktivitäten leicht an ein verändertes Planungsumfeld angepasst werden kann (vgl. [Jens01]). Während die Flexibilität die Möglichkeit zur Plananpassung beschreibt, betrachtet die Stabilität den Umfang der notwendigen Anpassungsmaßnahmen. Ein Plan ist umso stabiler, je weniger Änderungen erfolgen müssen. In der Praxis besteht die Zielsetzung einer hohen Stabilität des Plans, da eine häufige Planänderung die Abstimmung mit anderen Unternehmensbereichen wie z.B. der Personaleinsatzplanung oder der Materialdisposition erschweren bzw. Umplanungskosten verursachen kann (vgl. [Neuh08]). Aus diesen Gründen soll auch in dieser Arbeit eine möglichst hohe Stabilität der Planungsergebnisse verfolgt werden.

2.2.5 Komplexität der Planung

Zur Generierung eines Produktionsplans für den beschriebenen Flexible Flow Shop kann in der Theorie ein ergebnisorientierter Vergleich aller erlaubten Lösungsmöglichkeiten durchgeführt werden. Diese Methode (vollständige Enumeration) resultiert jedoch schon bei kleinen Beispielen inakzeptable Rechenzeiten. Dies wird schon bei der Betrachtung einer einzigen Produktionslinie deutlich. Um mit Sicherheit eine optimale Lösung zu finden, sind sämtliche unterschiedliche Reihenfolgen zu betrachten. Bei einer Menge von L Losen ($L = \{1, \dots, l\}$), sind dies somit $l!$ Möglichkeiten (Anzahl möglicher Permutationen). Sind nun z.B. 50 Lose einzuplanen, ergeben sich folglich $50! \approx 3,04 * 10^{64}$ mögliche Permutationen. Diese Kom-

²⁶ Zu den Produktionskosten werden zudem noch Kosten für z.B. Rohmaterialien oder Hilfsmitteln subsumiert. Diese Kosten verhalten sich proportional zu der produzierten Menge und sind nur dann entscheidungsrelevant, wenn sie sich im Zeitablauf ändern. Da dies nur in den seltensten Fällen zutrifft, werden diese Kosten häufig (auch in dieser Arbeit) vernachlässigt. Sie beeinflussen nicht das Planungsergebnis, wenn die Kosten je produziertem Erzeugnis im Zeitablauf als konstant angenommen werden.

plexität steigt mit der Größe des Produktionssystems. Es sei angenommen, dass ein Flexible Flow Shop aus n^{ps} Produktionsstufen bestehe mit jeweils einer Menge $PL_{ps} = \{1, \dots, pl_{ps}\}$ identischen parallelen Produktionslinien. Jedes Los l muss auf allen Produktionsstufen von genau einer Produktionslinie bearbeitet werden. Gourgand et. al. haben gezeigt, dass es für dieses Problem $l! (\prod_{i=1}^{n^{ps}} pl_i)^l$ unterschiedliche Möglichkeiten gibt (vgl. [GGN00]). Wird das Beispiel um die Annahme erweitert, dass das Produktionssystem aus 4 Produktionsstufen mit jeweils 3 Produktionslinien je Produktionsstufe besteht, resultieren $50! (\prod_{i=1}^4 3)^{50} \approx 8,08 * 10^{159}$ unterschiedliche Permutationen²⁷.

Um das Planungsproblem in eine Klasse der Komplexitätstheorie einordnen zu können, dient Abbildung 7. So wird in der Komplexitätstheorie die Aussage diskutiert, ob eine Lösung überhaupt gefunden werden kann. D.h., in der Komplexitätstheorie wird nicht die Lösbarkeit eines Problems behandelt, sondern die Schwierigkeit eine Lösung zu finden, deren Existenz gesichert ist. Hierzu wird in der Regel das Verhältnis zwischen der Größe einer konkreten Problemrealisierung und der Zeit zur Lösung dieses Problems mit Hilfe eines Rechners im schlechtesten Fall und mit dem besten bekannten deterministischen Algorithmus gemessen.

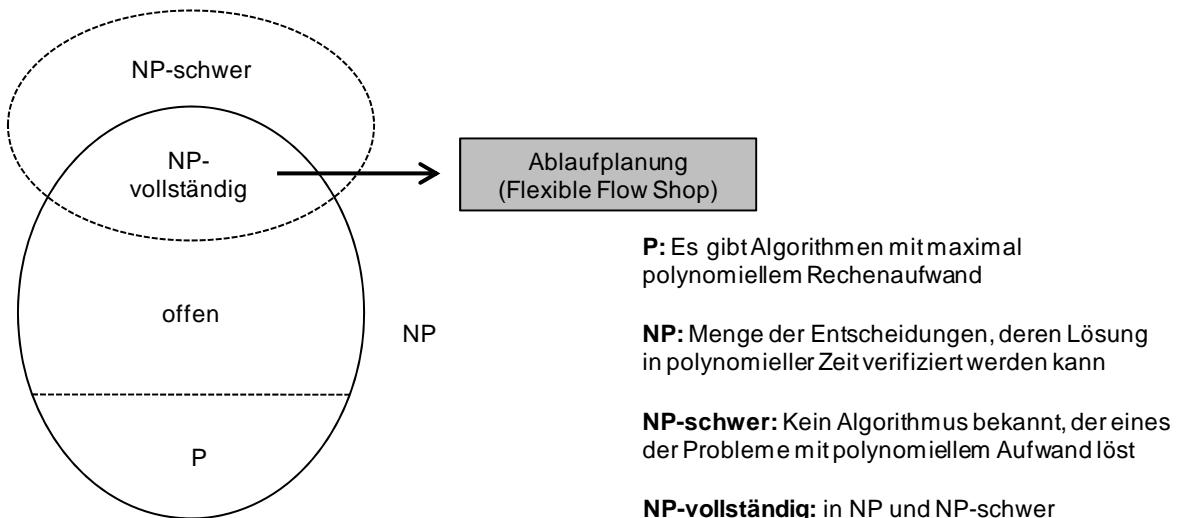


Abbildung 7: Zusammenhang der Komplexitätsklassen²⁸

Entscheidungsprobleme lassen sich hinsichtlich ihrer Komplexität in polynomiell (P), nicht-deterministisch polynomiell (NP), NP-vollständig und NP-schwer unterscheiden.²⁹ Für die Probleme der Klasse P existieren Algorithmen, die eine Lösung mit polynomiellen Rechenaufwand bestimmen können (vgl. [DSV93]). Für die NP-schweren Probleme (insb. NP-vollständigen Probleme) wird angenommen, dass dies nicht möglich ist.

²⁷ Um diese Größenordnung zu veranschaulichen, können folgende Vergleiche betrachtet werden: die Anzahl der Atome der Erde beträgt $\approx 10^{51}$, die Anzahl der Atome der Sonne $\approx 10^{57}$, die Anzahl aller Atome in unserer Galaxis $\approx 10^{67}$ und die Anzahl der gesamten Atome im Weltall (ohne dunkle Materie) $\approx 10^{77}$ (vgl. [BSW06]).

²⁸ In Anlehnung an [DSV93].

²⁹ Zur detaillierten Unterteilung siehe [GaJo79].

In der Literatur haben sich verschiedene Autoren mit der Bestimmung der Klasse für unterschiedlich geartete Probleme des Flexible Flow Shop beschäftigt. So haben bspw. Gary und Johnson sowie Sriskandarajah und Ladet gezeigt, dass unter der Zielsetzung einer minimalen Gesamtdurchlaufzeit das Problem NP-vollständig ist (vgl. [GaJo79], [SrLa86]). Diese Komplexitätsklasse konnte von Gupta sogar für zweistufige Produktionssysteme nachgewiesen werden, wenn nur eine der Produktionsstufen über parallele Maschinen verfügt (vgl. [Gupt88]). Dies impliziert, dass auch der in dieser Arbeit betrachtete Flexible Flow Shop mit der geforderten Flexibilität der Kapazitäten NP-vollständig ist. D.h., es existiert kein Verfahren, mit dem das Planungsproblem in polynomieller Zeit optimal lösbar ist. Eine Erhöhung der Eingabeparameter bedingt zugleich einen exponentiellen Zuwachs des Rechenaufwands.

2.3 Planungsverfahren für den Flexible Flow Shop

„Das Fertigungsplanungs-Verfahren ist eine festgelegte oder geeignet erzeugte Folge von zielgerichteten Transformationen der Plandaten, so daß die durch die Fertigungsplanungsaufgabe gestellten Anforderungen erfüllt werden.“ [Dang01] Zur Planung eines Flexible Flow Shop wurden in der Vergangenheit unterschiedliche Verfahrenstypen genutzt. Dies sind u.a. Optimierungsverfahren, heuristische Verfahren, Simulationsmodelle und Verfahren der künstlichen Intelligenz. Es hat sich jedoch gezeigt, dass es bislang kein universelles Verfahren zur Lösung aller Probleme gibt (vgl. [Saue02]). Stattdessen ist die Lösung eines Planungsproblems von den Umgebungsbedingungen und den zu erreichenden Zielen abhängig. Es hat sich jedoch gezeigt, dass für Ablaufplanungsprobleme besonders Optimierungsverfahren und Heuristiken gut geeignet sind (vgl. [Saue02]).

2.3.1 Optimierungsverfahren

Optimierung bezeichnet die Ermittlung derjenigen gültigen Handlungsalternative, die einem vorgegebenen Ziel von allen möglichen Alternativen bestmöglich entspricht. Eine wichtige Voraussetzung ist die Existenz eines Lösungsraums, der eine Suche nach einem Optimum in endlicher Zeit zulässt. Hierzu werden mathematische Modelle erstellt, die von geeigneten Verfahren gelöst werden. Ein solches Modell besteht aus Entscheidungsvariablen, Restriktionen und einer Zielfunktion (vgl. [SuMe06]). Die Restriktionen definieren die zugrunde liegenden Zusammenhänge des Optimierungsproblems unter Verwendung der Entscheidungsvariablen. Eine gültige Lösung liegt vor, wenn sämtliche Restriktionen für die jeweilige Belegung der Entscheidungsvariablen eingehalten werden. Neben den Restriktionen ist zudem eine Zielfunktion zu formulieren, die die Entscheidungsvariablen enthält und es so ermöglicht, die Güte der verschiedenen Lösungen zu bewerten. In Abhängigkeit der gewünschten Zielsetzung ist eine Belegung dieser Entscheidungsvariablen zu finden, die die Zielfunktion unter Berücksichtigung der Restriktionen minimiert oder maximiert. Die Optimierung setzt voraus, dass alle gültigen Handlungsalternativen bei einer bekannten Zielsetzung berücksichtigt werden. Eine Methode hierzu stellt die vollständige Enumeration dar, die jedoch mit einer enormen Komplexität verbunden und daher im Rahmen der Ablaufplanung unter praxisrelevanten Bedingungen nicht anwendbar ist (vgl. [Jung07]). Um den Rechenaufwand zu reduzieren, wurden Verfahren wie z.B. die begrenzte Enumeration und Branch & Bound entwickelt, die Lösungen ausschließen, deren Nichtoptimalität bereits bekannt ist. Die Ermittlung des

Optimums einer praxisnahen Instanz eines Problems der Ablaufplanung ist jedoch auch mit diesen allgemeinen Verfahren bislang nicht in akzeptabler Rechenzeit möglich.

2.3.2 Heuristiken

Falls eine Optimierung nicht möglich oder aufgrund der Modellstruktur und -größe zu aufwändig ist, können Heuristiken geeignet sein, möglichst gute Ergebnisse mit einem vertretbaren Rechenaufwand zu ermitteln (vgl. [SuMe06]). Dies gilt grundsätzlich für NP-schwere und NP-vollständige Probleme (siehe Kapitel 2.2.5) wie bei vielen Reihenfolge-, Zuordnungs- und Gruppierungsproblemen der kombinatorischen Optimierung.

Grundsätzlich wird unter einer Heuristik eine Vorgehensweise verstanden, mit der durch ein systematisches (sukzessives) Vorgehen eine möglichst gute Lösung bzgl. einer definierten Zielsetzung gefunden wird. Eine im mathematischen Sinne optimale Lösung oder eine Gültigkeit kann jedoch nicht garantiert werden und es ist in der Regel nicht möglich, eine Abschätzung darüber zu geben, wie weit die gefundene Lösung vom Optimum entfernt ist (vgl. [Teic98]). Durch die Aufgabe dieser zwingenden sowie meist illusorischen Optimalität verringert sich der nötige Rechenaufwand drastisch. Der Verzicht auf die Optimalität wird dadurch begründet, dass ein Optimum nur bzgl. eines Modells erreicht werden kann. Da dieses jedoch nur eine vereinfachte Abbildung der Realität darstellt, bleibt ein tatsächliches Optimum meist unmöglich. Zudem überwiegt ab einer bestimmten Lösungsgüte der zeitliche Nachteil des Rechenaufwands den Vorteil der möglichen Verbesserung.

Die klassischen Heuristiken erstellen unter Verwendung von Regeln und dem Ausnutzen bekannter Eigenschaften des zu lösenden Problems teilweise nur eine einzige, häufig jedoch gute Lösung. Moderne Heuristiken entwickeln und bewerten häufig sehr schnell sehr viele unterschiedliche Alternativen und können dadurch vielfach auch komplexe Probleme mit einer hohen Güte lösen. Eine Sonderform dieser modernen Heuristiken bilden Metaheuristiken. Eine Metaheuristik ist vor allem dadurch charakterisiert, dass sie problemunspezifisch und somit auf beliebige Probleme anwendbar ist. Dazu wird eine übergeordnete Strategie genutzt, die eine problemspezifische Heuristik steuert und modifiziert. Dadurch ist es möglich, zu besseren Ergebnissen zu kommen, als dies mit der spezifischen Heuristik allein möglich wäre (vgl. [ZäBr05]). Metaheuristiken basieren meist auf dem Grundansatz der lokalen Suche (z.B. Nachbarschaft³⁰) und/oder auf dem evolutionären/rekombinationsbasierten Prinzip. Im Rahmen der lokalen Suche erfolgt eine sukzessive Variation einer Lösung. Im Gegensatz dazu werden beim zweiten Ansatz neue Lösungskandidaten primär durch eine Kombination von Elementen mehrerer Lösungen generiert. Eine wichtige Voraussetzung von Metaheuristiken besteht darin, dass sie in einem Lösungsraum lokale Optima erkennen und auch verlassen können.

Die heuristischen Verfahren lassen sich entweder der Gruppe der Eröffnungs- oder Verbesserungsverfahren subsumieren. Eröffnungsverfahren bestimmen eine (erste) gültige Lösung

³⁰ Eine Nachbarschaft ist eine Menge von Lösungen, die sich durch kleine Modifikationen der aktuellen Basislösung bilden lassen (vgl. [DoSc05]).

eines Problems. Der Fokus liegt auf der Gültigkeit, statt auf der Güte des Ergebnisses. Die Lösung selbst wird häufig durch eine sukzessive Hinzunahme von Lösungselementen konstruiert (vgl. [AFI+08]). Bekannte Verfahren sind Prioritäts- oder Vorausschauregeln. „*Eine Prioritätsregel ist eine Vorschrift, die entsprechend den zugeordneten Zahlenwerten eine Auswahl (aus der Konfliktmenge) gestattet*“ [ZäBr05]. Sie versuchen in jedem Iterationsschritt eine maximale Verbesserung zu erzielen, ohne die möglichen Auswirkungen auf die nachfolgenden Schritte zu prognostizieren. Genau dies versuchen die Vorausschauregeln. Eine durch ein Eröffnungsverfahren generierte Lösung dient oft als Basis des Verbesserungsverfahrens. In diesem wird durch sukzessives Absuchen des Lösungsraums iterativ versucht, eine Lösung mit einer höheren Güte bzgl. der gegebenen Zielsetzung zu erzielen. Bekannte Verbesserungsverfahren sind Tabu Search, Simulated Annealing oder genetische Algorithmen.

2.4 Anforderungen an die Problemlösung

Basierend auf den vorangegangenen Beschreibungen des Problembereichs werden im Folgenden die Anforderungen an ein geeignetes Planungsverfahren sowie eine Methode zur Evaluierung dieser formuliert.

2.4.1 Anforderungen an ein Planungsverfahren

Folgende Anforderungen werden an ein Verfahren zur Ablaufplanung eines Flexible Flow Shop gestellt:

- Güte der Planungsergebnisse: Aufgrund des komplexen Planungsproblems (vgl. Kapitel 2.2) ist die simultane Planung mit optimalen Verfahren unter praxisrelevanten Bedingungen nicht möglich. Aus diesem Grund soll ein heuristisches Verfahren entwickelt werden, das im Vergleich zu einer optimalen Lösung keinen signifikanten Unterschied aufweist. Um diesen Vergleich anstellen zu können, soll für das gleiche Problem ein gemischt-ganzzahliges mathematisches Modell definiert und mit adäquaten Probleminstanzen gelöst werden.
- Schnelligkeit des Verfahrens: Das Verfahren soll mit größeren Probleminstanzen eine Lösung in zumutbarer Zeit erzielen können. Hierzu soll der Planungshorizont in einen Fein- und Grobplanungshorizont unterteilt und in Abhängigkeit der zeitlichen Lage dieser Horizonte in unterschiedlicher Granularität und Flexibilität beplant werden. Den größten Detaillierungsgrad verlangt die Planung des Feinplanungshorizontes und stellt daher die größere Herausforderung und den Schwerpunkt dieser Arbeit dar.
- Verfahren des Feinplanungshorizontes: Das zu entwickelnde Verfahren soll einen Plan erstellen, der einen über alle Produktionsstufen und Puffer gültigen Durchlauf aller Lose ermöglicht. Ein solcher Plan liegt vor, wenn zum Startzeitpunkt eines Loses, ggf. unter Beachtung der Vorlaufzeit, sämtliche Vorerzeugnisse zur Verfügung stehen. Hierzu soll das Verfahren aus zwei Stufen bestehen. In einem Eröffnungsverfahren soll eine gültige Ausgangslösung erzeugt werden. Auf deren Basis soll ein Verbesserungsverfahren nach einer bestmöglichen Lösung suchen. Die Beplanung der einzelnen Produktionslinien der jeweiligen Produktionsstufe soll auf Basis von Mikroperioden erfolgen. Dies erlaubt die Bildung einer Losreihenfolge unter Beachtung produk-

tionslinien- und reihenfolgeabhängiger Rüstzeiten. Bei Auftreten von Bedarfsänderungen sollen möglichst geringe Anpassungen des bestehenden Plans notwendig sein. Im Rahmen dieser Anpassungen soll das Verfahren die Startzeitpunkte von Losen auf der ursprünglich geplanten Produktionslinie verändern oder die Lose auf alternative Produktionslinien verschieben können. Zudem soll auch das Splitten von Losen ermöglicht werden. Eine höhere Reagibilität bei auftretenden Bedarfsänderungen soll außerdem durch die Wahl des jeweils zur Produktion am besten geeigneten Schichtmodells bestehen. Diese Schichtmodelle können sich hinsichtlich der Arbeitszeiten und Leistungsgrade unterscheiden.

- Verfahren des Grobplanungshorizontes: Für viele Unternehmensbereiche sind Informationen über einen längerfristigen Produktionsplan relevant. So erfolgt bspw. auf einem solchen Plan der Einkauf von Rohmaterialien. Aber auch für die Erstellung eines guten Plans der Feinplanung ist eine weitsichtige Betrachtung zukünftiger Bedarfe vorteilhaft. Dadurch können, bei ggf. freien Kapazitäten, Bedarfe in den Feinplanungsbereich vorgezogen werden, was eine Produktion mit Verzug vermeiden kann. Die Beplanung der einzelnen Produktionslinien der jeweiligen Produktionsstufe soll auf Basis von Makroperioden erfolgen. Hierzu werden die Bedarfsumfänge einer Makroperiode zu Losen zusammengefasst und einer Makroperiode zugeordnet. Eine Reihenfolgebildung innerhalb einer Makroperiode erfolgt nicht. Die Berücksichtigung von Rüstzeiten wird auf Basis von Durchschnittswerten vorgenommen. Aufgrund des ggf. großen Grobplanungshorizontes können die Bedarfe innerhalb des Horizontes starken Schwankungen unterliegen, so dass aus einer jeweiligen Beibehaltung des ursprünglichen Plans starke Abweichungen zu einem optimalen Plan resultieren können. Aus diesem Grund wird für den Grobplanungsbereich der jeweils alte Plan verworfen und ein neuer erzeugt. Da dies jedoch auch starke Veränderungen von Schichtmodellzuordnungen implizieren kann, wird auf Basis fester Kapazität geplant. Zur Lösung soll ein gemischt-ganzzahliges mathematisches Modell dienen. Nach einem Grobplanungslauf ist das Ergebnis hinsichtlich etwaiger Verzugsmengen zu analysieren. Liegen diese vor, sollen, sofern noch freie Kapazitäten vorhanden sind, passende Bedarfe in den Feinplanungshorizont verschoben werden. Hierbei wird die Annahme zu Grunde gelegt, dass Verzugskosten in jedem Fall teurer sind als eine vorzeitige Produktion. Zur Identifikation passender Bedarfsumfänge soll ein entsprechendes Bedarfsglättungsverfahren erstellt werden.
- Parametrisierung durch den Planer: Um das Planungsverfahren an verschiedene Situationen anpassen zu können, soll der Benutzer in den Entscheidungsprozess integriert werden. So besitzen erfahrende Planer Expertenwissen, das durch ein Planungsverfahren nicht vollständig abgedeckt werden kann. Das Verfahren soll daher für den Benutzer transparent sein und ihm die Möglichkeit bieten, verschiedene Parameter einzustellen.

2.4.2 Anforderungen an ein Benchmarkingverfahren

Zur Evaluierung der Planungsverfahren ist ein geeignetes Verfahren zu entwickeln. An dieses werden folgende Anforderungen gestellt:

- Implementierung eines Benchmarkingsystems: Die in 2.4.1 geforderten Verfahren sollen implementiert und in einem IT-System zusammengeführt werden. Dieses System soll die Möglichkeit bieten, die zur Analyse notwendigen Daten konsistent zu pflegen. Dies sind z.B. Daten über den Aufbau des Produktionssystems, Eigenschaften der einzelnen Erzeugnisse sowie planungsverfahrensspezifische Parameter. Das System soll zudem für verschiedene Szenarien eine rollierende Planung durchführen können und die jeweiligen Ergebnisse dem Benutzer transparent über geeignete Kennzahlen darstellen.
- Bedarfzahlengenerator: Um die einzelnen Verfahren objektiv zu bewerten, ist eine Methode zu entwickeln, die für einen definierten Zeithorizont eine initiale Bedarfsmenge generiert und diese entsprechend dem Konzept der rollierenden Planung sukzessiv verändert. Um spätere Analysen zu ermöglichen, sind diese Daten persistent zu speichern.
- Definition von Kriterien zum Vergleich von Planungsverfahren: Um ein aussagekräftiges, objektives und zielgerichtetes Benchmarking durchzuführen, soll das Benchmarkingverfahren über verschiedene Kriterien zur Evaluierung der Planungsverfahren verfügen. Diese Kriterien sollen sowohl monetäre Aspekte als auch Anforderungen einer dynamischen Planungsumgebung berücksichtigen.

3 Stand der Technik

„Sobald der Geist auf ein Ziel gerichtet ist,
kommt ihm vieles entgegen.“

(Johann Wolfgang von Goethe)

In diesem Kapitel werden die für diese Arbeit relevanten Ansätze diskutiert. In Kapitel 3.1 werden mehrstufige mathematische Modelle zur optimalen Lösung präsentiert. Diese stellen die Grundlagen der Modelle der Fein- sowie Grobplanung dar. Dem schließen sich ausgewählte heuristische Verfahren an (siehe Kapitel 3.2). Zum Abschluss dieses Kapitels werden in Kapitel 3.3 Verfahren zur Bewertung der Güte eingeführt.

3.1 Mehrstufige mathematische Modelle

Zur optimalen Lösung von Planungsproblemen mit Hilfe eines Optimierungssystems bedarf es geeigneter Modelle. Zu deren Erstellung ist festzulegen, welche Aspekte der Realität modelliert werden sollen. Hierzu „muss zunächst geklärt werden, was der Entscheidungsträger wirklich will (Zielfunktion), welche Freiheitsgrade (Entscheidungsvariablen) man hat und welche Nebenbedingungen (Restriktionen) berücksichtigt werden sollen“ [SuMe06]. Da ein Modell jedoch eine Abstraktion der Realität darstellt, können zu einer Problemstellung mehrere unterschiedliche Modelle existieren. Daher besteht das Ziel der Modellbildung in der Definition eines möglichst guten bzw. geeigneten Modells. Abbildung 8 illustriert den Zusammenhang zwischen der Modellierung sowie der Anwendung von Lösungsverfahren und die Notwendigkeit eines formalen Modells zur rechnergestützten Lösung des Problems.

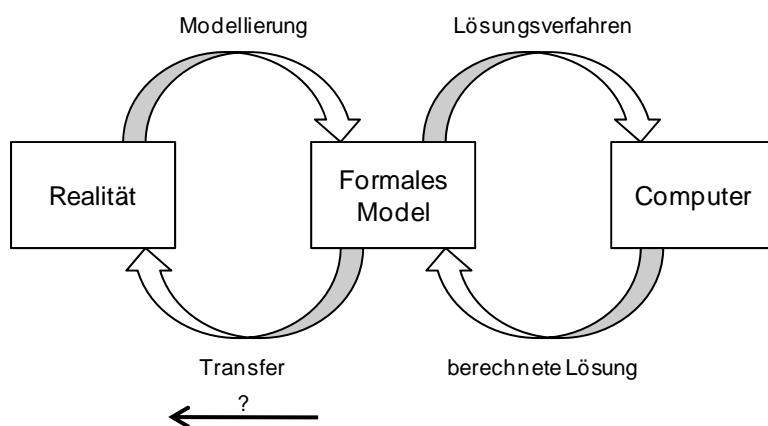


Abbildung 8: Modellierung und Lösungsverfahren³¹

Die in der Ablaufplanung verwendeten Modelle werden in Big- und Small-Bucket-Probleme differenziert. Bei einem Big-Bucket-Problem erfolgt keine Planung der Produktreihenfolge.

³¹ In Anlehnung an [SuMe06].

Die einzelnen Buckets werden sehr groß gewählt (z.B. ein Tag, eine Woche oder ein Monat) und können beliebig viele Produktwechsel beinhalten. Im Gegensatz dazu fokussieren Small-Bucket-Probleme auf Perioden mit einer Länge von Schichten, Stunden oder Minuten und erlauben höchstens ein singuläres Ereignis pro Periode, wodurch eine entsprechende Reihenfolge der Lose generiert wird. Im Vergleich zu den Big-Bucket-Modellen bedarf diese Art der Planung jedoch auch einen wesentlich höheren Ressourcenaufwand. (vgl. [Stam02])

Nachfolgend werden ein Big- und Small-Bucket-Modell im Detail vorgestellt. Anhand dieser sollen die Struktur eines mathematischen Modells sowie die Unterschiede zwischen den beiden Arten verdeutlicht werden. Da zur Berücksichtigung der beschriebenen Interdependenzen eines mehrstufigen Produktionssystems einstufige Modelle³² in der Regel ungeeignet sind, wird die Diskussion auf mehrstufige Modelle beschränkt.

3.1.1 Multi-Level Continuous Lot Sizing and Scheduling Problem

Das MLCLSP (Multi-Level Continuous Lot Sizing and Scheduling Problem) ist ein Modell des Big-Bucket-Problems und wurde von Billington et. al. (vgl. [BMT83]) definiert. Mit diesem wird eine mehrstufige Mehrproduktfertigung abgebildet, wobei der Planungshorizont begrenzt und in diskrete Perioden geteilt ist. Die externen Bedarfe liegen periodenspezifisch vor und sind fristgerecht, d.h., ohne Verzug zu befriedigen. Ein Erzeugnis auf einer Produktionsstufe kann nur auf einer Produktionslinie mit begrenzter Kapazität hergestellt werden. Um bspw. Transport- oder Trockenzeiten zu berücksichtigen, werden beim Übergang von einer Produktionsstufe zur nächsten Produktionsstufe Vorlaufzeiten berücksichtigt. Für jedes Erzeugnis, das in einer Periode hergestellt wird, muss gerüstet werden. Eine Übertragung von Rüstzuständen über mehrere Perioden ist nicht möglich (vgl. [BSH+08]). Das Modell berücksichtigt eine externe Nachfrage sowohl für Zwischen- als auch für Enderzeugnisse.³³

Zur Modellierung werden folgende Mengen, Parameter und Variablen verwendet (vgl. [Temp06]):

Definition der Mengen

$E = \{1..e\}$	Menge aller Erzeugnisse
$E_{pl} = \{1..e_{pl}\}$	Menge aller Erzeugnisse, die auf Produktionslinie pl gefertigt werden
$N_e = \{1..n_e\}$	Menge der Nachfolger des Erzeugnisses e (in der Erzeugnishierarchie direkt übergeordnete Erzeugnisse)
$PL = \{1..pl\}$	Menge der Produktionslinien
$T = \{1..t\}$	Länge des Planungshorizontes

³² Für weitere Informationen zu einstufigen Small- und Big-Bucket-Modellen sei auf [Temp06] und [Dang09] verwiesen.

³³ In der Literatur werden zahlreiche Formulierungen des MLCLSP mit verschiedenen Erweiterungen diskutiert. Eine Übersicht hierzu findet sich in [BSH+08].

Definition der Parameter

$a_{e,j} \in \mathbb{N}$	Direktbedarfskoeffizient, der angibt, wie viele Einheiten des Erzeugnisses e zur Herstellung einer Einheit des Erzeugnisses j benötigt werden
$b_{e,t} \in \mathbb{N}$	Bedarf des Erzeugnisses e in Periode t
$B \in \mathbb{N}$	Hinreichend große Zahl
$k_{pl,t} \in \mathbb{N}$	Kapazitätsangebot einer Produktionslinie pl in Periode t
$lk_e \in \mathbb{R}^0$	Lagerkostenfaktor für eine Einheit des Erzeugnisses e
$p_{e,t} \in \mathbb{R}^0$	Produktionskostensatz für Erzeugnis e in Makroperiode t
$rk_e \in \mathbb{R}^0$	Produktabhängige Rüstkosten für Erzeugnis e
$rz_e \in \mathbb{N}$	Rüstzeit für Erzeugnis e
$tb_e \in \mathbb{N}$	Stückbearbeitungszeit für Erzeugnis e
$vlz_e \in \mathbb{N}$	Vorlaufzeit von Erzeugnis e

Definition der Variablen

$I_{e,t} \in \mathbb{N}$	Lagerbestand für Erzeugnis e am Ende der Periode t
$q_{e,t} \in \mathbb{N}$	Losgröße für Erzeugnis e in Periode t
$y_{e,t} \in \{0,1\}$	Binäre Rüstvariable, die angibt, ob in der Periode t eine Umrüstung stattfindet ($y_{e,t} = 1$) oder nicht ($y_{e,t} = 0$)

3.1.1.1 Grundmodell

3.1.1.1.1 Zielfunktion

Durch die Zielfunktion (3.1) werden die über alle Erzeugnisse und Perioden aufsummierten Rüst-, Lagerhaltungs- und Produktionskosten minimiert.

$$\min z = \text{Lagerkosten} + \text{Rüstkosten} + \text{Produktionskosten} \quad (3.1)$$

Die Produktionskosten sind nur dann relevant, wenn sie sich im Zeitablauf ändern.

$$\min z = \sum_{e \in E} \sum_{t \in T} lk_e * I_{e,t} + rk_e * y_{e,t} + p_{e,t} * q_{e,t} \quad (3.2)$$

3.1.1.1.2 Restriktionen

Die Lagerbilanzgleichung (3.3) verhindert das Auftreten von Fehlmengen. Die gesamte Nachfrage eines Erzeugnisses setzt sich aus dem für dieses Erzeugnis bekannten Primärbedarf $b_{e,t}$ und aus dem durch die direkten Nachfolger verursachten Sekundärbedarfe zusammen.

Um zudem die rechtzeitige Verfügbarkeit zu garantieren, wird für die Erzeugnisse eine Vorlaufzeit vlz_e berücksichtigt.

$$I_{e,t-1} + q_{e,t-vlz_e} - \sum_{j \in N_e} a_{e,j} * q_{j,t} - I_{e,t} = b_{e,t} \quad \forall e \in E \quad \forall t \in T \quad (3.3)$$

Im Rahmen des MLCLSP wird angenommen, dass für jede Produktion eines Erzeugnisses eine Umrüstung stattfinden muss (vgl. [Stam02]). Sichergestellt wird dies durch die Restriktionen (3.4) und der binären Rüstvariable.

$$q_{e,t} - B * y_{e,t} \leq 0 \quad \forall e \in E \quad \forall t \in T \quad (3.4)$$

Die Restriktion (3.5) garantiert, dass die in einer Periode zur Verfügung stehenden Kapazitäten nicht überschritten werden.

$$\sum_{e \in E_{pl}} (tb_e * q_{e,t} + rz_e * y_{e,t}) \leq k_{pl,t} \quad \forall pl \in PL \quad \forall t \in T \quad (3.5)$$

Die Restriktion (3.6) unterstellt, dass zu Beginn und zum Ende des Planungshorizontes keine Erzeugnisse auf Lager liegen.

$$I_{e,0} = 0; I_{e,T} = 0 \quad \forall e \in E \quad (3.6)$$

Die Vorlaufzeiten werden modellseitig immer in Form ganzer Perioden beachtet. Sofern diese Vorlaufzeiten geringer sind, werden damit die Vorprodukte zu früh produziert, was höhere Lagerkosten impliziert.

3.1.1.2 Modellerweiterungen

In diesem Kapitel werden für diese Arbeit relevante Erweiterungen des vorherigen Grundmodells vorgestellt. In [Dang09] wird ein Modell präsentiert, das die Summe aus Produktions-, Rüst-, Lagerhaltungs-, Verzug- und Fehlmengenkosten minimiert. Zusätzlich zu dem Grundmodell berücksichtigt es u.a. verschiedenartige parallele Produktionslinien. Zur Transformation in ein Modell zur Lösung eines Flexible Flow Shop werden zusätzliche Bedingungen für die Parameter des Modells definiert. Hierzu werden die Menge der Erzeugnisse E und die der Produktionslinien PL entsprechend der Anzahl an Produktionsstufen PS in nicht-leere Teilmengen zerlegt. Für diese muss gelten:

$$E = \bigcup_{ps=1}^{|PS|} E_{ps}^{PS} \bigwedge_{ps=1}^{|PS|} PL = \bigcup_{ps=1}^{|PS|} PL_{ps}^{PS} \quad (3.7)$$

$$\forall 1 \leq ps_1 < ps_2 \leq |PS|: E_{ps_1}^{PS} \cap E_{ps_2}^{PS} = \emptyset = PL_{ps_1}^{PS} \cap PL_{ps_2}^{PS} \quad (3.8)$$

$$\forall i \in E_{|PS|}^{PS}: N_i = \emptyset \quad (3.9)$$

$$\forall 1 < ps \leq |PS|: E_{ps}^{PS} \subseteq \bigcup_{1 \leq \sigma < ps} \bigcup_{i \in E_{\sigma}^{PS}} N_i \quad (3.10)$$

$$\forall 1 \leq ps \leq |PS|: PL_{ps}^{PS} = \bigcup_{i \in E_{ps}} PL_i \quad (3.11)$$

Die Bedingungen (3.7) bis (3.11) definieren ein sehr generelles Problem eines Flexible Flow Shop. Sie sorgen dafür, dass sämtliche Erzeugnisse und Produktionslinien einer Produktionsstufe eindeutig zugeordnet werden. Zudem garantieren sie, dass die Eigenschaften eines Flexible Flow Shop nicht verletzt werden und der Produktionsprozess azyklisch ist. Dies verlangt, dass die Erzeugnisse der letzten Produktionsstufe PS keine Nachfolger N_i besitzen. Es wird außerdem sichergestellt, dass sich Kundenbestellungen auf sämtliche im Produktionsprozess gefertigte Erzeugnisse beziehen und nicht nur auf die der letzten Produktionsstufe. Zudem werden keine zu starren Einschränkungen hinsichtlich des Aufbaus der Erzeugnisse vorgenommen. So muss ein Erzeugnis nicht zwangsläufig auf der direkten, sondern kann auf jeder beliebigen nachfolgenden Produktionsstufe weiterverarbeitet werden.

Das Grundmodell macht keine Angaben zur Reihenfolge der Bearbeitung. Aus diesem Grund werden in jeder Periode Rüstkosten und Rüstzeiten veranschlagt. Hierbei wird aber nicht berücksichtigt, ob in der Vorperiode das gleiche Erzeugnis produziert wurde. Stattdessen werden die Rüstkosten in jeder Periode neu berechnet. Die Abbildung 9 illustriert diesen Zusammenhang. Es wird deutlich, dass die Übertragung von Rüstzuständen (im Falle einer periodenüberschreitenden Produktion) den Rüstzustand von Periode t in die Periode $t + 1$ fortzuschreiben ermöglicht und entsprechend weniger Kapazitätsbedarf besteht.

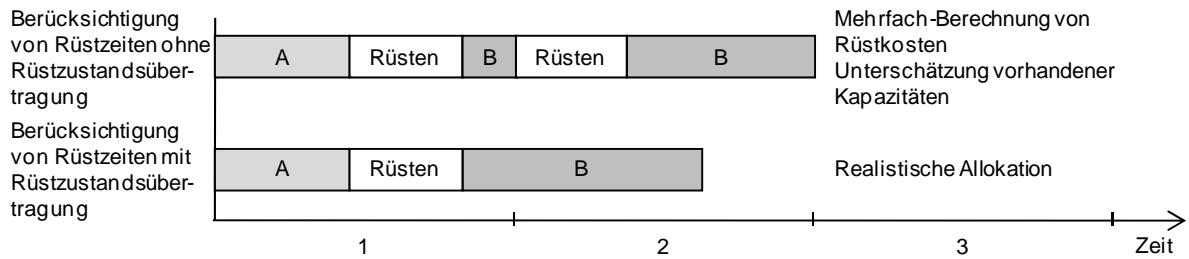


Abbildung 9: Auswirkungen der Rüstzustandsübertragung³⁴

Um nicht das Problem in ein Small-Bucket-Modell umzuwandeln, schlagen [SBT+09] die Einführung zusätzlicher Zustandsvariablen und Restriktionen vor. Diese bestimmen den Rüstzustand zu Beginn einer Periode und ein Erzeugnis, das innerhalb einer Makroperiode an den Periodengrenzen produziert wird.

3.1.2 Multi-Level Discrete Lot Sizing and Scheduling Problem

Das MLDLSP³⁵ (Multi-Level Discrete Lot Sizing and Scheduling Problem) ist ein Modell der Small-Bucket-Probleme. In diesem Modell findet eine Produktion immer nur über ganze Zeit-

³⁴ (vgl. [Suer05])

abschnitte statt. Die Produktionsmenge ist 0 oder das volle Äquivalent eines Zeitabschnitts („*Alles-oder-nichts-Produktion*“) [Flei90]. Analog zum MLCLSP kann durch eine Hinzunahme der Bedingungen (3.7) bis (3.11) die Abbildung eines Flexible Flow Shop erfolgen. Zur Modellierung dieses Problems werden folgende Mengen, Parameter und Variablen verwendet:

Definition der Mengen

$E = \{1..e\}$	Menge der Erzeugnisse
$N_e = \{1..n_e\}$	Menge der Nachfolger des Erzeugnisses e
$T = \{1..t\}$	Länge des Planungshorizontes

Definition der Parameter

$a_{e,j} \in \mathbb{N}$	Direktbedarfskoeffizient, der angibt, wie viele Einheiten des Erzeugnisses e zur Herstellung einer Einheit des Erzeugnisses j benötigt werden
$b_{e,t} \in \mathbb{N}$	Bedarf nach Erzeugnis e in Periode t
$k_t \in \mathbb{N}$	Kapazitätsangebot in Periode t
$lk_e \in \mathbb{R}^0$	Lagerkostenfaktor für eine Einheit des Erzeugnisses e
$p_{e,t} \in \mathbb{R}^0$	Produktionskostensatz für Erzeugnis e in Periode t
$rk_e \in \mathbb{R}^0$	Produktabhängige Rüstkosten für Erzeugnis e
$tb_e \in \mathbb{N}$	Stückbearbeitungszeit für Erzeugnis e

Definition der Variablen

$I_{e,t} \in \mathbb{N}$	Lagerbestand eines Erzeugnisses e in Periode t
$q_{e,t} \in \mathbb{N}$	Losgröße für Erzeugnis e in Periode t
$x_{e,t} \in \{0,1\}$	Binäre Rüstvariable, die angibt, ob in der Periode t eine Umrüstung für Erzeugnis e stattfindet ($x_{e,t} = 1$) oder nicht ($x_{e,t} = 0$)
$y_{e,t} \in \{0,1\}$	Binäre Rüstsitzungsvariable, die angibt, ob in der Periode t für Erzeugnis e gerüstet ist ($y_{e,t} = 1$) oder nicht ($y_{e,t} = 0$)

3.1.2.1 Zielfunktion

Durch die Zielfunktion (3.12) wird die Summe aus Rüst- und Lagerhaltungskosten minimiert. Die Produktionskosten sind nur dann relevant, wenn sie sich im Zeitablauf ändern.

$$\min z = \text{Rüstkosten} + \text{Lagerkosten} + \text{Produktionskosten} \quad (3.12)$$

³⁵ Ein weiteres bekanntes Small-Bucket-Modell ist das MLPLSP (Multi-Level Proportional Lot Sizing and Scheduling Problem). Das im weiteren Verlauf dieser Arbeit dargestellte Verfahren liegt jedoch der Charakteristik des MLDLSP zugrunde. Aus diesem Grund sei für weitere Informationen zum MLPLSP auf [Kimm96] verwiesen.

Rüstkosten entstehen in Abhängigkeit des in der vorherigen Periode produzierten Erzeugnisses. Lagerkosten resultieren aus dem Lagerbestand des Erzeugnisses und dessen erzeugnis-spezifischen Lagerkostensatz (siehe Formel (3.13)).

$$\min \sum_{e \in E} \sum_{t \in T} (rk_e * x_{e,t} + lk_e * I_{e,t} + p_{e,\mu} * q_{e,t}) \quad (3.13)$$

3.1.2.2 Restriktionen

Die Lagerbestandsgleichung (3.14) stellt sicher, dass der Bedarf einer Periode durch den Lagerbestand und der in dieser Periode produzierten Menge gedeckt wird. Neben den direkten Bedarfen werden zudem die resultierenden internen Bedarfe berücksichtigt. Durch diese Erweiterung gegenüber dem einstufigen DLSP wird die Mehrstufigkeit erzeugt.

$$I_{e,t-1} + q_{e,t} - \sum_{j \in N_e} (a_{e,j} * q_{e,t}) - I_{e,t} = b_{e,t} \quad \forall e \in E \quad \forall t \in T \quad (3.14)$$

Die Alles-oder-nichts-Bedingung wird durch die Formel (3.15) sichergestellt. Sie sorgt dafür, dass die gesamte Kapazität einer Periode zur Herstellung eines Erzeugnisses genutzt wird, falls es in dieser Periode auf der Produktionslinie gefertigt wird.

$$tb_e * q_{e,t} = k_t * y_{e,t} \quad \forall e \in E \quad \forall t \in T \quad (3.15)$$

Durch die Nebenbedingung (3.16) wird sichergestellt, dass in jeder Periode höchstens ein Erzeugnis hergestellt wird.

$$\sum_{e \in E} y_{e,t} \leq 1 \quad \forall t \in T \quad (3.16)$$

Die Formel (3.17) gewährleistet, dass Rüstkosten nur entstehen, wenn in der Periode $t - 1$ ein anderes Erzeugnis produziert wird, als in Periode t . Diese Formel stellt die Rüstzustandsübertragung über mehrere Perioden sicher.

$$x_{e,t} - y_{e,t} + y_{e,t-1} \geq 0 \quad \forall e \in E \quad \forall t \in T \quad (3.17)$$

Die Nebenbedingung (3.18) drückt aus, dass der Lagerbestand zu Beginn des Planungshorizontes 0 Mengeneinheiten beträgt.

$$I_{e,0} = 0 \quad \forall e \in E \quad (3.18)$$

3.1.3 Optimale Verfahren für den Flexible Flow Shop

In der Literatur werden verschiedene Arbeiten zur Generierung eines optimalen Ablaufplans für einen Flexible Flow Shop diskutiert. Im Rahmen dieses Kapitels werden einige ausgewählte Verfahren vorgestellt und anschließend hinsichtlich der Anwendbarkeit auf das in dieser Arbeit unterstellte Problem betrachtet³⁶. Einen guten Überblick über diese Verfahren wird von Wang, Quadt sowie Linn und Zhang gegeben (vgl. [Wang05], [Quad04], [LiZh99]). Diese werden im Folgenden aufgegriffen und ergänzt. Wie schon bereits im zweiten Kapitel dargestellt, ist die optimale Lösung nur für sehr kleine Instanzen möglich. Das Lösungsproblem ist NP-hart oder NP-vollständig (vgl. [GaJo79], [Pine95]). Die entwickelten Verfahren verfolgen als Ziel zumeist die Minimierung der Gesamtdurchlaufzeit bis zur Fertigstellung aller Aufträge (engl. makespan).

Die ersten Arbeiten stammen von Johnson, Arthanari und Ramamurthy sowie Salvador (vgl. [John54], [ArRa71], [Salv73]). Zur Lösung schlagen sie, wie eine Mehrheit der weiteren Arbeiten, Verfahren auf Basis von Branch & Bound vor.

Johnson betrachtet einen zweistufigen (in bestimmten Fällen dreistufigen) Flow Shop mit Rüstzeiten³⁷. Während sich auch Arthanari und Ramamurthy bei der Lösungsgröße noch auf einen zweistufigen Flexible Flow Shop begrenzen, erlaubt die Methode von Salvador ein Produktionssystem beliebiger Größe. Hierzu erzeugt das Verfahren einen Plan, in dem die Reihenfolge der Aufträge auf allen Produktionsstufen identisch ist³⁸. Kern eines solchen Vorgehens ist die Annahme, dass ein sich gegenseitiges Überholen von Aufträgen im Verlauf der Bearbeitung nicht möglich ist und somit sämtliche Aufträge in der gleichen Reihenfolge bearbeitet werden. Dadurch reduziert sich das Reihenfolgeproblem auf lediglich eine Produktionsstufe. Beiden Verfahren (von Arthanari und Ramamurthy sowie Salvador) ist jedoch gemein, dass sie weder Rüstzeiten noch Puffer zwischen den einzelnen Produktionsstufen – und folglich keine Zwischenlagerbestände – berücksichtigen. Diese Lücke schließt das Verfahren von Brah und Hunsucker (vgl. [BrHu91]). So entwickeln sie ein Verfahren, das sowohl unterschiedliche Reihenfolgen auf den Produktionsstufen erzeugen kann als auch Zwischenpuffer berücksichtigt. Zudem erlaubt das Verfahren die Berücksichtigung von Leerlaufzeiten. Ein Nachteil dieses Vorgehens liegt jedoch in der sehr hohen Lösungszeit. Rajendran und Chaudhuri entwickeln ein Verfahren, in dem die Größe des Entscheidungsbaums durch die Erzeugung eines Permutationsplans reduziert wird (vgl. [RaCh92b]). Den Entwicklungen moderner Computer wird in den Arbeiten von Brockmann et. al. (vgl. [BDH97]) und (vgl. [Broc98]) Rechnung getragen. Auf Basis von Brah und Hunsucker (vgl. [BrHu91]) erstellen die Autoren ein Verfahren, das die Rechenleistung massiv paralleler Systeme nutzt. Zur Bestimmung der unteren Schranke des Branch & Bound Verfahrens wird das Maximum aus einer Jobbasierten Schranke, einer Stufen-basierten Schranke sowie einer Maschinen-basierten Schran-

³⁶ Diese Zusammenstellung verfolgt auch das Ziel, einen kurzen Überblick über die Entstehungsgeschichte moderner Verfahren zur Planung eines Flexible Flow Shop zu geben. Aus diesem Grund werden zudem ausgewählte Arbeiten zur Planung eines „Flow Shop“ einbezogen.

³⁷ „Der Grundgedanke des Verfahrens besteht darin, dass der erste Vorgang des ersten Jobs und der letzte Vorgang des letzten Jobs in voller Länge in die Gesamtbearbeitungszeit eingehen. Je kürzer also diese Vorgänge sind, umso kürzer kann die totale Durchlaufzeit sein.“ [Dang09]

³⁸ Ein solcher Produktionsplan wird auch als „Permutationsplan“ bezeichnet.

ke gebildet. Zudem wird eine initiale obere Schranke mit Hilfe einfacher Verfahren (z.B. Erzeugung einer zufälligen gültigen Lösung) berechnet. Zur Exploration des Lösungsraums finden grundsätzlich die Strategie des Best First Branch & Bound sowie Backtracking Anwendung. Eine Weiterentwicklung nutzt zudem unterschiedliche Arten des Branchings sowie der Bestimmung unterer Schranken (vgl. [Broc98]). Zudem werden Verfügbarkeiten von Produktionslinien berücksichtigt. Ebenfalls auf den Arbeiten von Brah und Hunsucker (vgl. [BrHu91]) setzen Portmann et. al. (vgl. [PVD+98]) auf. Sie verbessern das Verfahren in zwei wesentlichen Punkten. So erfolgt die Ermittlung von oberen Schranken mit Hilfe eines genetischen Algorithmus. Außerdem erzeugen sie bessere untere Schranken, so dass Lösungen schneller gefunden werden können. Ein weiteres auf Branch & Bound basierendes Verfahren stellen Moursli und Pochet (vgl. [MoPo00]) vor. Analog zu den vorherigen Verfahren wird mit ihrem Verfahren keine Simultanplanung über alle Produktionsstufen vorgenommen. Daraus unterscheidet sich das Verfahren von Carlier und Néron (vgl. [CaNé00]). Sie kombinieren Branch & Bound mit Constraint Propagation und erzeugen den Entscheidungsbaum fortlaufend, indem sukzessiv eine Produktionsstufe um den als nächstes einzuplanenden Auftrag erweitert wird. Néron et. al. (vgl. [NBG01]) nutzen eine Branch & Bound Methode, die durch so genannte „satisfiability tests“ und „time-bound adjustments“ verbessert wurde.

Die bislang diskutierten Verfahren bedienen sich zur Lösung der bereits beschriebenen Branch & Bound Methode. Einen anderen Weg beschreiten Guinet et. al. (vgl. [GSK+96]). Sie formulieren für einen zweistufigen Flexible Flow Shop ein gemischt ganzzahliges Modell. Diesem Vorgehen bedient sich auch Sawik (vgl. [Sawi00], [Sawi02]). Er formuliert ein gemischt ganzzahliges Modell, in dem entgegen der meisten in der Literatur unterstellten Annahmen kapazitive Puffer integriert sind. Sein zweites Modell (vgl. [Sawi02]) berücksichtigt außerdem batch scheduling. Ruiz et. al. (vgl. [RŞU08]) formulieren ein gemischt ganzzahliges Modell für ein komplexeres Produktionssystem. Sie berücksichtigen u.a. verschiedenartige parallele Produktionslinien, in Abhängigkeit der Produktionslinie und Reihenfolge auftretende Rüstzeiten, Verzögerungen der Produktionslinien, Freigabezeitpunkte für Produktionslinien, Produktionslinieneignung sowie unterschiedliche Auftragsprioritäten.

Die obigen Verfahren verfolgen alle das Ziel, die Gesamtdurchlaufzeit zu minimieren. Ein Verfahren zur Minimierung der mittleren Durchlaufzeit (engl. mean flow time) wird von Rajendran und Chaudhuri (vgl. [RaCh92a]) vorgestellt. Analog zu ihrem Verfahren zur Minimierung der Gesamtdurchlaufzeit erstellen die Autoren mit diesem Verfahren einen Permutationsplan. Auf Basis ihrer Arbeit erstellen Azizoglu et. al. (vgl. [ACK01]) ein Verfahren, mit dem unterschiedliche Reihenfolgen auf den einzelnen Produktionsstufen erzeugt werden können. Die meisten der in der Literatur existierenden Verfahren betrachten lediglich die Reihenfolgebildung. Die durch die Reihenfolge- und Losgrößenbildung bedingten Abhängigkeiten werden folglich nicht berücksichtigt. Eine Ausnahme stellt die Arbeit von Quadt (vgl. [Quad04]) dar. In dieser betrachtet er in einem integrierten Lösungsansatz das Losgrößen-, Maschinenbelegungs- und Reihenfolgeproblem. Erweitert wird dieses Problem um die gleichzeitige Behandlung identischer paralleler Produktionslinien und reihenfolgeabhängiger Rüstzeiten. Basis seines Ansatzes ist das Konzept der hierarchischen Planung, mittels derer er zuerst mit einem erweiterten CLSP eine Engpassstufe beplant. Von dieser ausgehend wird anschließend der Plan auf die anderen Produktionsstufen übertragen. Als Ziel betrachtet er das

Minimum aus Lager-, Verzugs- und Rüstkosten. Als weiteres Zielkriterium nennt er eine möglichst minimale durchschnittliche Produktionszeit eines Auftrags. In [HCM04] schlagen die Autoren ein ganzzahliges Optimierungsmodell zur Lösung eines Produktionssystems mit jeweils gleichartigen parallelen Produktionslinien je Produktionsstufe vor. Als Ziel verfolgen auch die Autoren die Bestimmung optimaler Losgrößen, die primär die Minimierung der Gesamtdurchlaufzeit sowie sekundär eine optimale Anzahl an Arbeitskräften an jeder Produktionslinie erreichen.

Diese Übersicht an Verfahren verdeutlicht, dass in den vergangenen Jahrzehnten zu unterschiedlich komplexen Problemen verschiedene Arbeiten zur Generierung eines optimalen Ablaufplans entwickelt wurden. Aufgrund der großen Komplexität konnte jedoch mit keinem der Verfahren eine Lösung unter praxisrelevanten Bedingungen gefunden werden. Aus diesem Grund wurde in vielen Fällen zusätzlich für das gleiche Planungsproblem ein alternativer Lösungsansatz verfolgt und dessen Güte mit einem für geringe Testinstanzen erzeugten optimalen Ablaufplan ermittelt.

3.2 Heuristische Lösungsverfahren

Als Alternative zu den optimalen Verfahren stehen zur Lösung der beschriebenen Planungsprobleme zahlreiche Heuristiken zur Verfügung. Diese lassen sich generell in Eröffnungs- und Verbesserungsverfahren einteilen. Nach der Diskussion allgemeiner Verfahren schließt sich die Darstellung reaktiver Methoden an.

3.2.1 Eröffnungsverfahren

Das Ziel von Eröffnungsverfahren ist, eine (erste) zulässige Lösung auf direktem Weg durch die Anwendung von Regeln für das betrachtete Problem zu konstruieren (vgl. [DoDr05]). Nachdem eine solche Lösung gefunden wurde, wird diese dann jedoch von dem Eröffnungsverfahren nicht verbessert. In der Praxis finden aufgrund der leichten Handhabbarkeit häufig Verfahren auf Basis von Prioritätsregeln Anwendung. Diese weisen in jedem Verfahrensschritt den noch einzuplanenden Aufträgen Werte anhand einer Prioritätsregel zu und wählen als nächsten einzuplanenden Auftrag denjenigen mit dem minimalen bzw. maximalen Wert.

3.2.1.1 Prioritätsregeln

Im Rahmen der Produktionsplanung – auch im Kontext dynamischer Planungsumgebungen – werden häufig Prioritätsregeln eingesetzt, um Aufträge den zur Verfügung stehenden Ressourcen (z.B. Produktionslinien) zuzuweisen. Dies ist insbesondere dann wichtig, wenn mehrere Aufträge um eine Ressource konkurrieren. Nach Corsten ist eine Prioritätszahl „*als eine Vorgabe zu interpretieren, die entsprechend den zugeordneten Wertzahlen eine Auswahl aus einer Konfliktmenge erlaubt*“ [Cors04].

In der Literatur wurden in den vergangenen Jahren eine Vielzahl unterschiedlicher Regeln vorgeschlagen. In Untersuchungen zur Bewertung der Güte der einzelnen Regeln zur Planung eines Flexible Flow Shop wurde jedoch konstatiert, dass es keine Universalregel gibt. So ist

die Güte einer jeweiligen Regel vielmehr abhängig von der Zielsetzung (vgl. [RaHo99]). Im Folgenden werden einige bekannte Regeln vorgestellt. Eine Differenzierung dieser kann anhand der Veränderlichkeit der Zeit erfolgen. Bei statischen Regeln bleibt eine einmal berechnete Prioritätszahl über die Zeit unverändert. Im Gegensatz dazu erfolgt bei dynamischen Regeln während des Planungsablaufs jeweils wieder eine Neuberechnung (vgl. [FFG94]). Eine weitere Einordnung kann zudem nach dem zur Berechnung der Prioritätszahl benötigten Informationsbedarf vorgenommen werden. Hierbei werden globale und lokale Prioritätsregeln unterschieden. Während die erst genannten die Prioritätszahl basierend auf Informationen über den Zustand des gesamten Produktionssystems berechnen, ermitteln lokale Prioritätsregeln die Prioritätszahl nur über Informationen des jeweiligen Arbeitssystems (vgl. [Daub94]). Weiterhin werden elementare und kombinierte Prioritätsregeln differenziert. Während bei den elementaren Prioritätsregeln nur ein Reihenfolgekriterium genutzt wird, setzen sich die kombinierten Regeln aus mehreren elementaren Regeln zusammen. Beispiele für die elementaren Regeln sind (vgl. [Teic98], [ZäBr05]):

- FCFS-Regel (first come first served-Regel): Der Auftrag, der zuerst auf dem jeweiligen Arbeitssystem ankommt, wird als nächstes bearbeitet.
- FFT-Regel (frühester Fertigstellungstermin-Regel): Die höchste Prioritätszahl erhält der Auftrag mit dem frühesten Fertigstellungstermin.
- FIFO-Regel (first in first out-Regel): Es bekommt der Auftrag die höchste Priorität, der als erster im gesamten Produktionssystem eingetroffen ist.
- UK-Regel (Umrüstungskosten-Regel): Der Auftrag, der die geringsten Umrüstungskosten an dem Arbeitssystem verursacht erhält in der Warteschlange Priorität.
- KW-Regel (Kundenauftragsprioritäts-Regel): Die Aufträge, die für die „wichtigsten Kunden“ sind, werden bevorzugt.
- WT-Regel (Wert-Regel): Der Auftrag in der Warteschlange bekommt die höchste Priorität, der den höchsten Produktwert hat.
- GGB/KGB-Regel (größte/kürzeste Gesamtbearbeitungszeit-Regel): Der Auftrag mit der größten/kürzesten Gesamtbearbeitungszeit auf allen Arbeitssystemen wird als nächstes bearbeitet.

Durch eine Kombination der verschiedenen Regeln können die guten Eigenschaften gezielt verstärkt und die schlechten eliminiert werden (vgl. [Sieg74]). Diese Kombination erfolgt auf unterschiedlichen Arten und sollte sich an dem entsprechenden Planungsproblem mit den jeweiligen Zielsetzungen orientieren (vgl. [Daub94]). In der Literatur werden die additive, multiplikative, alternative und dominante Verknüpfung differenziert.³⁹ Insgesamt stellen Prioritätsregeln einfache, leicht verständliche, heuristische Lösungen für eine Problemstellung dar. Neben einer hohen Anwenderakzeptanz bieten sie zudem den Vorteil einer hohen Flexibilität. So ist es möglich, die jeweils für ein Arbeitssystem am besten geeignete Prioritätsregel zu verwenden. Zudem können sie in dynamischen Produktionsumgebungen herangezogen werden (vgl. [KiSt01]). Ein Nachteil stellt die kurzfristige Sichtweise dieser Verfahren dar. So werden die langfristigen Konsequenzen der jeweiligen Entscheidung auf das zukünftige Entscheidungsproblem vernachlässigt (vgl. [KiSt01]). Weiterhin existieren keine Anhaltspunkte

³⁹ Für weitere Informationen siehe [ZäBr05].

für die Auswahl passender Prioritätsregeln. Die Wirksamkeit hängt jeweils von dem vorliegenden Produktionssystem ab.

3.2.1.2 Heuristische Verfahren für den Flexible Flow Shop

In den vergangenen Jahren wurde eine große Anzahl an Arbeiten zur Planung des Flexible Flow Shop veröffentlicht. Diese unterscheiden sich stark durch das jeweils betrachtete Produktionssystem und die entsprechende Zielsetzung. Im Folgenden werden einige Ansätze dargestellt.⁴⁰ Eine wichtige Differenzierung bezieht sich auf die Größe des zu planenden Produktionssystems. So wird die Komplexität eines Problems stark durch die Anzahl betrachteter Produktionsstufen beeinflusst.

Einen zweistufigen Flexible Flow Shop betrachten Narasimhan und Panwalker (vgl. [NaPa84]). In diesem Produktionssystem begrenzen die Autoren jedoch die Anzahl jeweiliger Produktionslinien auf einer Produktionsstufe. So erlauben sie eine Produktionslinie auf der ersten und zwei gleichartige parallele Produktionslinien auf der zweiten Produktionsstufe. Narasimhan und Mangiameli [NaMa87] erweitern die Annahmen um die Möglichkeit, auch mehrere identische Produktionslinien auf der ersten Produktionsstufe zu berücksichtigen. Zur Lösung entwickelten sie zwei spezielle Prioritätsregeln (CMD-Regel - Cumulative Minimum Deviation und GCMD - Generalized Cumulative Minimum Deviation). Mit zweistufigen (Flexible) Flow Shops beschäftigen sich auch Gupta und seine Kollegen intensiv. In ihren Arbeiten, die das Ziel der Minimierung der Gesamtdurchlaufzeit verfolgen, betrachten sie unterschiedlich komplexe Probleme. Zu Beginn betrachten die Autoren einen Flexible Flow Shop (vgl. [Gupt88], [GuTu91]), der auf der zweiten Produktionsstufe identische parallele Produktionslinien erlaubt. In weiteren Arbeiten werden Rüst- und Abrüstzeiten (vgl. [GuTu94]) sowie identische parallele Produktionslinien auf der ersten Produktionsstufe (vgl. [GHP97]) berücksichtigt. Die Heuristik von Srisandarajah und Sethi (vgl. [SrSe89]) basiert auf dem Verfahren von Johnson (vgl. [John54]) und verfolgt die Minimierung der Gesamtdurchlaufzeit eines Flexible Flow Shop mit identischen parallelen Produktionslinien. Um die Lösungsgüte von Heuristiken im worst case und average case zu bestimmen, entwickeln die Autoren untere Schranken. Auf Basis von Johnson (vgl. [John54]) betrachten auch Guinet et. al. (vgl. [GSK+96]) ein Verfahren zur Minimierung der maximalen Bearbeitungszeit von Aufträgen (z.B. die Gesamtdurchlaufzeit). Im Benchmark mit verschiedenen Prioritätsregeln (shortest processing time, longest processing time) konstatieren sie gute Lösungsergebnisse.

Es gibt auch Heuristiken, die primär für eine Anwendung in der Praxis konzipiert wurden. So entwickeln bspw. Adler et. al. (vgl. [AFK+93]) ein Verfahren für ein dreistufiges Produktionssystem der Verpackungsindustrie. Das Problem beinhaltet reihenfolgeabhängige Rüstzeiten sowie unterschiedliche Auftragsprioritäten. Die Autoren stellen fest, dass den einzelnen Produktionslinien eine besondere Bedeutung zukommt. So unterstellen viele der in der Literatur betrachteten Verfahren auf einer Produktionsstufe identische parallele Produktionslinien.

⁴⁰ Diese Zusammenstellung verfolgt auch das Ziel, einen kurzen Überblick über die Entstehungsgeschichte moderner Verfahren zur Planung eines Flexible Flow Shop zu geben. Aus diesem Grund werden zudem ausgewählte Arbeiten zur Planung eines „Flow Shop“ einbezogen. Als Basis dieser Darstellung dienen die Arbeiten von [Wang05], [Quad04], [LiZh99].

Diese Annahme bestätigt sich jedoch in der Praxis nur selten. So werden in den meisten Unternehmen alte sowie ggf. neuere leistungsfähigere Produktionslinien eingesetzt. Die älteren können nun über die gleiche qualitative Kapazität verfügen, jedoch hinsichtlich der Anzahl möglicher Operationen stark von neueren abweichen. Die Planung dieses Produktionssystems erfolgt ausgehend von der Engpassstufe mit Hilfe einer Prioritätsregel. Auf Basis dieses Plans werden anschließend auf den anderen Produktionsstufen die jeweiligen Pläne erzeugt. Um auf diesen Produktionsstufen Rüstzeiten zu sparen, sind begrenzte Änderungen der Reihenfolge erlaubt. Die Zielsetzung besteht in der Minimierung der durchschnittlichen Verspätung, der in Bearbeitung befindlichen Aufträge und der Maximierung des Durchsatzes.

Für einen Flow Shop mit einer unbegrenzten Anzahl an Produktionsstufen erstellen Campbell, Dudek und Smith (vgl. [CDS70]) sowie Gupta (vgl. [Gupt71]) Verfahren auf Basis von Johnson (vgl. [John54]). Verfahren für den Flexible Flow Shop mit identischen parallelen Produktionslinien werden von Wittrock (vgl. [Witt85], [Witt88]) entwickelt. In dem ersten Verfahren erstellt der Autor einen periodischen Algorithmus, mit dem eine kleine Anzahl von Aufträgen geplant und dann die Sequenz jeweils entsprechend oft wiederholt wird. Dieses Verfahren basiert auf der problemorientierten Dekomposition⁴¹ in zwei sukzessiv zu lösende Teilprobleme. Im ersten Schritt (job loading) werden die Aufträge auf die Produktionslinien zugeordnet, auf denen sie bearbeitet werden sollen. Im zweiten Schritt (time allocating) wird die Bearbeitungsreihenfolge jeder Produktionslinie ermittelt. Für das gleiche Problem entwickelt Wittrock einen nicht-periodischen Algorithmus. Diese Lösungsfindung basiert auf der Einführung eines dritten Schrittes. So besteht sein Verfahren nun aus der Produktionslinienzuordnung (machine allocation), der Reihenfolgebestimmung der Aufträge (sequencing) sowie der Zuordnung zu einem Zeitpunkt (timing). Das Zielkriterium umfasst die Minimierung der Gesamtdurchlaufzeit sowie die längste Warteschlange eines Puffers. Auf Basis von Wittrock (vgl. [Witt85]) erstellen Kochhar und Morris (vgl. [KoMo87]) ein Verfahren, das von der Komplexität weitaus größer ist. So berücksichtigen die Autoren Rüstzeiten, begrenzte Puffer, Ausfälle von Produktionslinien, blocking⁴² und starvation⁴³ sowie Stillstandszeiten der Produktionslinien. Das Verfahren verfolgt die Minimierung der durchschnittlichen Fertigungszeit. Ding und Kittichartphayak (vgl. [DiKi94]) präsentieren drei unterschiedliche Heuristiken für einen Flexible Flow Shop mit identischen parallelen Produktionslinien. Die ersten beiden Algorithmen erweitern die Methode von Campbell et. al. (vgl. [CDS70]). Die dritte Heuristik stellt eine Erweiterung von Gupta (vgl. [Gupt72]) dar. Das Ziel all dieser Verfahren ist die Minimierung der Gesamtdurchlaufzeit. Daniels und Mazzola (vgl. [DaMa94]) integrieren in

⁴¹ Dekompositionsansätze teilen ein Planungsproblem in mehrere, weniger komplexe Planungsprobleme, welche im Anschluss einzeln betrachtet werden können. Durch diese Vereinfachung werden allerdings die Interdependenzen zwischen den Teilproblemen vernachlässigt. Daher ist eine Kernaufgabe jedes Dekompositionsansatzes, effiziente Strategien zur Berücksichtigung dieser Problematik zur Verfügung zu stellen. Es werden z.B. die auftragsorientierte, produktionsstufenorientierte und problemorientierte Dekomposition unterschieden. Für weitere Informationen siehe [Temp06].

⁴² Blocking bezeichnet einen Zustand, in dem einer Produktionslinie nachgelagerter Puffer ausgelastet ist. In diesem Fall wird die Bearbeitung auf der Produktionslinie so lange gestoppt, bis der Puffer wieder über genügend Kapazität verfügt.

⁴³ Starvation tritt ein, wenn für die Herstellung eines Erzeugnisses nicht genügend Eingangsmaterialien zur Verfügung stehen, der Materialfluss also abreißt und die Produktionslinie still steht und auf Eingangsmaterialien warten muss.

ihrem Verfahren zudem die Planung von Kapazitäten. Die Autoren betrachten einen Flow Shop, wobei eine Besonderheit ihres Verfahrens in der Berücksichtigung von Leistungsgraden besteht. So ist die Bearbeitungszeit eines Auftrags auf einer Produktionsstufe von einer nach oben beschränkten Anzahl zugeordneter Arbeitskräfte abhängig. Um zudem unterschiedliche Kapazitätsangebote zu ermöglichen, wird der Planungshorizont in einzelne Perioden unterteilt. Das Ziel ihres Verfahrens besteht in der integrierten Bestimmung einer Produktionsreihenfolge (einem Permutationsplan) sowie geeigneter Leistungsgrade (und der damit verbundenen Berechnung entsprechender Bearbeitungs- und Anfangszeiten) unter Beachtung einer minimalen Gesamtdurchlaufzeit. In ihrem ersten Verfahren unterstellen die Autoren einen beliebigen Austausch von Arbeitskräften zwischen den einzelnen Produktionsstufen. Zur Lösung schlagen die Autoren zuerst einen optimalen Lösungsansatz und dann aufgrund der großen Komplexität eine Heuristik vor. In einer späteren Arbeit integrieren Daniels et. al. (vgl. [DMS04]) zudem unterschiedliche Fähigkeiten der einzelnen Arbeitskräfte. Hierzu werden für jede Arbeitskraft Matrizen erstellt, die die Qualifikation bzgl. einer jeweiligen Produktionslinie widerspiegeln. Einen weiteren Spezialfall eines Flexible Flow Shop mit identischen parallelen Produktionslinien wird von Sawik (vgl. [Sawi95]) unterstellt. Er erlaubt in seinem Problem keine Zwischenpuffer⁴⁴. Wird ein Erzeugnis auf einer Produktionslinie fertig gestellt, blockiert es diese so lang, bis auf der Nachfolgerproduktionsstufe mit der Bearbeitung begonnen werden kann. Der von ihm vorgeschlagene Algorithmus arbeitet joborientiert. D.h., dass die Planungsaufgabe zerlegt wird, indem die Aufträge sukzessiv betrachtet werden. Dabei wird in jedem Schritt ein Auftrag auf allen Produktionsstufen genau einer Produktionslinie zugeordnet. Die Reihenfolgeentscheidungen werden also simultan getroffen. Als Zielsetzung verfolgt auch Sawik die Minimierung der Gesamtdurchlaufzeit. Eine ganzheitliche Heuristik zur Planung eines Flexible Flow Shop mit identischen parallelen Produktionslinien erstellen Agnetis et. al. (vgl. [APR+97]). Mit ihren Verfahren lösen sie ein praktisches Problem aus der Automobilindustrie. Hierzu nutzen sie unterschiedliche Prioritätsregeln. So kombinieren sie eine push⁴⁵ (Round Robin)- und eine pull⁴⁶ (Minimum Slack Time)-Regel, wodurch sie bessere Ergebnisse erhalten als mit einer jeweiligen isolierten Betrachtung. Das Verfahren verfolgt die Minimierung der Anzahl verspäteter Teile, der in Bearbeitung befindlicher Teile sowie die Verbesserung der Produktivität. Santos et. al. (vgl. [SHD95], [SHD96]) betrachten einen Flexible Flow Shop mit identischen parallelen Produktionslinien. Sie definieren eine globale untere Schranke für das Ziel der Minimierung der Gesamtdurchlaufzeit. Diese dient zur Abschätzung der Lösungsgüte einer heuristischen Lösung. Diese Schranken werden in der zweiten Arbeit genutzt, um verschiedene Heuristiken zu bewerten. Besonders in einer dynamischen Planungsumgebung stellen Prioritätsregeln eine bewährte Planungsmethodik dar. Ein Verfahren für gleichartige parallele Produktionslinien stellen Verma und Dessouky (vgl.

⁴⁴ Die meisten der in der Literatur diskutierten Verfahren unterstellen eine unbegrenzte Pufferkapazität. Siehe z.B. [DiKi94]. Da dies jedoch eine unrealistische Annahme darstellt, haben mehrere Autoren in ihren Verfahren kapazitierte Puffer integriert. Beispiele finden sich in [Norm99], [Weng00] und [YBT09].

⁴⁵ Mit Hilfe einer push-Regel wird nach der Fertigstellung von Erzeugnissen auf einer Produktionsstufe eine Produktionslinie auf der entsprechenden Nachfolgerproduktionsstufe bestimmt. Die Belegung erfolgt auf Basis des entsprechenden Erzeugnisses.

⁴⁶ Bei einer pull-Regel erfolgt die Belegung von Produktionslinien ausgehend von den einzelnen Produktionslinien. Wenn eine Bearbeitung terminiert, wird aus einem globalen Puffer das nächste zu bearbeitende Erzeugnis ausgewählt. Es wird also immer das für die Produktionslinie am besten geeignete Erzeugnis gewählt.

[VeDe99]) vor. Eine Besonderheit von deren Verfahren stellt die Art der Auftragsbetrachtung dar. So unterstellen die Autoren, dass alle Aufträge auf einer bestimmten Produktionslinie die gleiche Bearbeitungszeit haben. Um die Komplexität der Planung zu reduzieren, werden die Produktionsstufen isoliert beplant, jeweils mit dem Ziel der Minimierung der Gesamtdurchlaufzeit. Zur Planung finden verschiedene Prioritätsregeln (z.B. LSF – latest start time oder ECT – earliest completion time) Anwendung. Neben dem Ziel der Minimierung der Gesamtdurchlaufzeit verfolgen einige Autoren auch andere Zielsetzungen. So minimiert die Heuristik von Yang et. al. (vgl. [YKP00]) die durchschnittliche gewichtete Verspätung eines Produktionssystems mit jeweils identischen parallelen Produktionslinien. Die Lösung erfolgt auf Basis einer produktionsstufenorientierten Dekomposition (mit Hilfe von Prioritätsregeln), einer local-search-Methode und der Kombination beider Verfahren. Die Minimierung der maximalen Verspätung wird mit dem Verfahren von Botta-Genoulaz (vgl. [Bott00]) angestrebt. Hierzu präsentiert er verschiedene Heuristiken. Diese werden für ein System mit identischen parallelen Produktionslinien konzipiert und berücksichtigen Prioritäten, Rüst- und Transportzeiten. In [GKL+02] werden die vormals von Gupta und seinen Kollegen erstellten Arbeiten um die Möglichkeit der Planung beliebig vieler Produktionsstufen mit identischen parallelen Produktionslinien erweitert. Zudem werden keine von vornherein bestimmten Prozesszeiten unterstellt, sondern es werden diese im Planungsprozess ermittelt. Zur Lösung nutzen die Autoren Eröffnungs- und Verbesserungsverfahren. Ruiz et. al. (vgl. [RŞU08]) berücksichtigen in ihrem Verfahren neben reihenfolgeabhängigen Rüstzeiten⁴⁷ eine Fülle weiterer Kriterien. So umfasst ihr Flexible Flow Shop verschiedenartige parallele Produktionslinien, unterschiedliche Freigabezeitpunkte sowie Prioritäten von Aufträgen. Des Weiteren erweitern sie die klassische Definition eines Flexible Flow Shop in dem Sinne, dass Aufträge auch Produktionsstufen überspringen können. Zur Beplanung teilen sie das gesamte Problem entsprechend der Produktionsstufen und nutzen für jede Produktionsstufe Prioritätsregeln.

Es wird deutlich, dass in den letzten Jahrzehnten für ein breites Spektrum ganz unterschiedlicher Problemstellungen direkte Lösungsverfahren entwickelt wurden. Während zu Beginn der Forschung der Fokus auf weniger komplexen Problemen lag, nahmen in Entwicklungen der jüngeren Vergangenheit die Anzahl zu berücksichtigender Faktoren im Planungsproblem stetig zu. Neben der Größe des Planungsproblems hat sich auch im Laufe der Zeit die Anzahl möglicher Zielsetzungen erhöht. Wurden zu Beginn der Forschung primär zeitorientierte Zielsetzungen verfolgt, werden in neueren Arbeiten – wie in Kapitel 2.2.1 erwähnt – auch kostenorientierte Ziele betrachtet.

3.2.2 Verbesserungsverfahren

Verbesserungsverfahren versuchen ausgehend von einer gültigen Lösung des Problems, die entweder zufällig oder durch Anwendung eines Eröffnungsverfahrens bestimmt wird, zu einer besseren Lösung zu gelangen. Um diese zu finden, werden vielfach Metaheuristiken eingesetzt. Einen Überblick über die unterschiedlichen Arten der Metaheuristiken zeigt Abbildung 10.

⁴⁷ Weitere heuristische Verfahren, die reihenfolgeabhängige Rüstzeiten berücksichtigen, finden sich z.B. in [LiCh00], [KuAs04], [PTR+04], [JRC+08] sowie [GZA09].

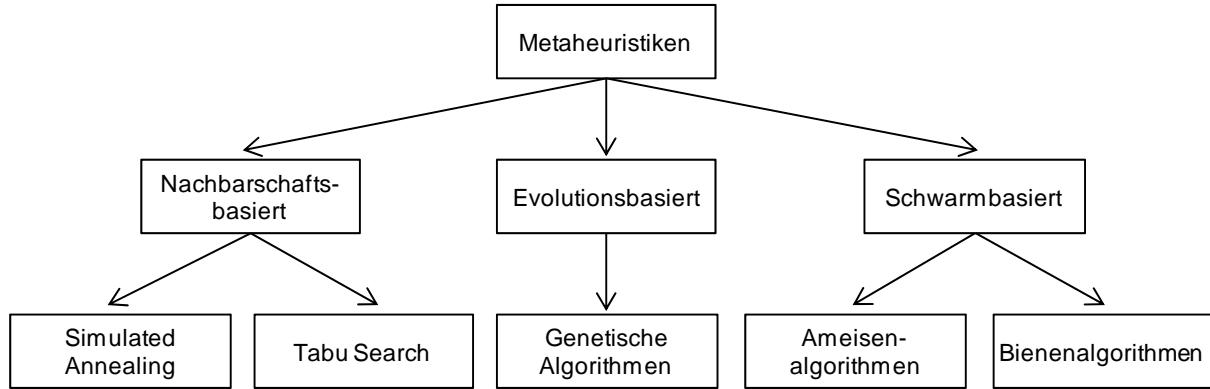


Abbildung 10: Klassifikation verbreiteter Metaheuristiken

Entsprechend dieser Klassifikation werden drei Kategorien differenziert. Nachbarschaftsbasierte Verfahren zeichnen sich dadurch aus, dass sie durch geringe Modifikationen der Ausgangslösung zu einer neuen Lösung (Nachbarlösung) gelangen. Genetische bzw. evolutionsbasierte Verfahren generieren Lösungen durch Nachahmung des Evolutionsprozesses mit der Selektion, Kreuzung sowie Mutation (vgl. [KOY95]). Durch die Selektion werden Lösungen bestimmt, die eine hohe Güte aufweisen. Aus diesen entstehen durch Kreuzung neue Lösungen, die im Nachhinein leicht zu verändern sind, um das genetische Ausdünnen zu vermeiden. Dieses Vorgehen wird auch als Mutation bezeichnet. Im Gegensatz dazu nutzen schwarmbasierte Verfahren das natürliche Kollektivverhalten einer Vielzahl von Individuen zur Lösungsfundung. Hierbei findet eine Kombination aus zwei Prinzipien Anwendung. Im Rahmen der Stigmergie ist es den Individuen mittels einer speziellen Kommunikation möglich, die Güte ihrer Einzellösungen mit zunehmender Zeit zu steigern. Dieses Vorgehen stellt einen sich selbst verstärkenden Prozess dar (das Prinzip der Autokatalyse) (vgl. [ZäBr05]).⁴⁸

In der Literatur existieren verschiedene Studien, um eine ordinale Ordnung bzgl. der Güte der einzelnen Metaheuristiken für Probleme der Ablaufplanung zu erstellen. In diesen erwiesen sich genetische Algorithmen gegenüber Simulated Annealing und Tabu Search als weniger geeignet. So bedürfen Verfahren auf Basis von genetischen Algorithmen mehr Zeit, um Lösungen nahe dem Optimum zu finden (vgl. [GKL95], [JMR98] und [YMA01]). Dies ist darin begründet, dass die nachbarschaftsbasierten Varianten pro Verbesserungsschritt tendenziell die geringsten Änderungen vornehmen. Bzgl. der Güte von Simulated Annealing und Tabu Search herrscht hingegen kein Konsens. Einige Autoren postulieren eine Überlegenheit von Simulated Annealing (vgl. [JRC+09]), andere Autoren betrachten Tabu Search als geeigneter (vgl. [NoSm98]). Gegenstand dieser Studien sind jedoch jeweils unterschiedlich komplexe Produktionssysteme sowie unterschiedliche Daten. Aus diesem Grund sind die Aussagen nicht generell übertragbar. Aufgrund der Forderung, einen Produktionsplan möglichst wenig zu ändern, kurzer Rechenzeiten sowie der intuitiven Vorgehensweise, werden jedoch im weiteren Verlauf dieser Arbeit Simulated Annealing und Tabu Search genutzt.

⁴⁸ Für weitere Informationen sei auf Zäpfel und Braune [ZäBr05] verwiesen.

3.2.2.1 Definition der Nachbarschaft

Zur Umsetzung der Tabu Search sowie des Simulated Annealing ist eine Nachbarschaft zu definieren. Sei L die Menge aller gültigen Lösungen für ein gegebenes Problem (der Lösungsraum). Existiert bereits eine gefundene, zufällige oder triviale Lösung $x_i \in L$, so ist die Nachbarschaft $N(x_i)$ definiert als die Menge „[...] aller zulässigen Lösungen, die durch genau eine (als atomar angesehene) Transformation aus Lösung x_i gebildet werden können“ [SuMe06]. Sämtliche Elemente der Nachbarschaft sind somit in der Nähe von x_i . Formal kann die Nachbarschaft als Funktion N beschrieben werden, die jedem $x_i \in L$ eine Teilmenge NL aus L zuordnet. D.h.: $N: x_i \rightarrow NL$ mit $NL \subseteq L, 0 < i \leq |L|$

Abbildung 11 illustriert das Prinzip des lokalen Suchprozesses, wobei sich die sukzessiv verbesserten Lösungen wie eine Kette durch den Lösungsraum ziehen. Zur Generierung einer neuen Nachbarlösung ist neben einer Initiallösung ein bestimmter Zug erforderlich, um die Transformation durchführen zu können. In der Literatur werden hierzu üblicherweise insert- sowie swap-Züge verwendet (vgl. [BKK+04]). Ein insert-Zug entfernt lediglich ein Element aus seiner Position und fügt es an einer alternativen Position wieder ein. Beim swap-Zug werden hingegen die Positionen der Elemente getauscht.

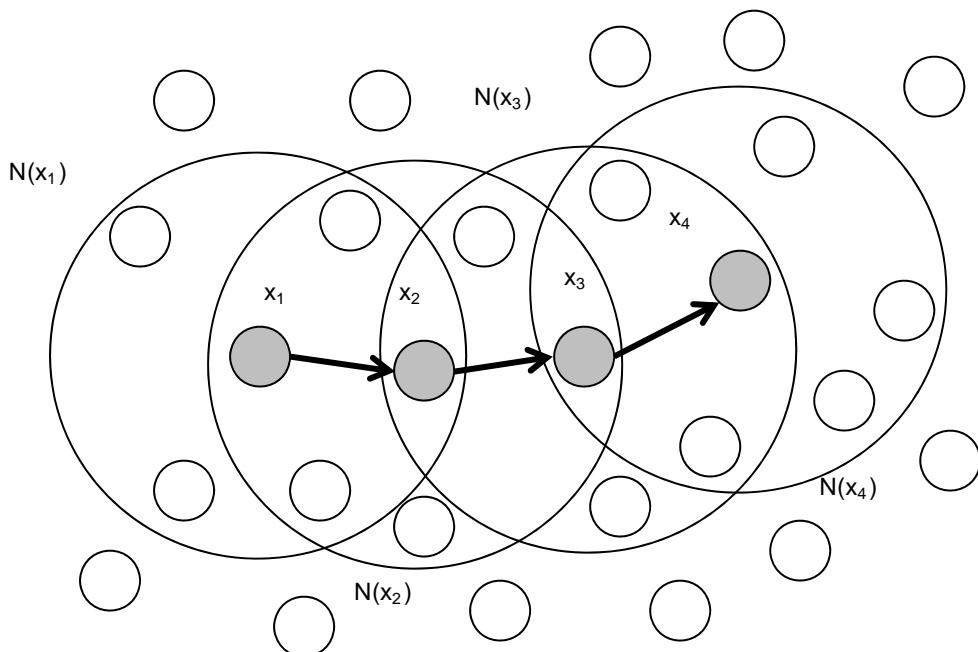


Abbildung 11: Allgemeines Prinzip der Nachbarschaftssuche⁴⁹

Das Ziel der lokalen Suche ist die beste benachbarte Lösung x' bzgl. einer aktuellen Lösung $x \in L$ zu finden, so dass für alle $x_n \in N(x)$ gilt $c(x') \leq c(x_n)$.

⁴⁹ vgl. [ZäBr05])

3.2.2.2 Simulated Annealing

3.2.2.2.1 Allgemeines

Das Verfahren des Simulated Annealing (Simulierte Abkühlung) hat seine Ursprünge in der Physik resp. Thermodynamik. Dort dient es der Simulation des Abkühlungsvorgangs, mit dem Ziel, einen Zustand minimaler Energie zu erreichen, in dem die Atome eine möglichst perfekt geordnete kristalline Struktur bilden. „*Dabei ist die Geschwindigkeit des Abkühlvorgangs entscheidend für die Anordnung der Atome. Erst wenn sich ein Gleichgewichtszustand eingestellt hat, darf die Temperatur weiter abgesenkt werden*“ [ZäBr05].

Die Grundidee des Simulated Annealing besteht darin, dass auch schlechtere Lösungen mit einer Akzeptanzwahrscheinlichkeit A angenommen werden können. Dies ermöglicht es, lokale Optima zu verlassen. Die Berechnung dieser Wahrscheinlichkeit beruht auf der Boltzmann-Verteilung. Diese verdeutlicht, dass mit geringer Wahrscheinlichkeit trotz niedriger Temperatur hochenergetische Zustände resultieren können. Für diese Wahrscheinlichkeit P gilt bei gegebener Energie E , Temperatur t und der Boltzmann-Konstante k :

$$P(E) = e^{\frac{-E}{k*t}} \quad \text{bzw.} \quad P(\Delta E) = e^{\frac{-\Delta E}{k*t}} \quad (3.19)$$

Eine Übertragung dieser Formel auf ein Optimierungsproblem ist möglich, indem der physikalische Energiebegriff durch eine Kostenfunktion ersetzt wird. Zur Vereinfachung kann die Boltzmann-Konstante vernachlässigt werden, so dass eine Akzeptanzwahrscheinlichkeit von $e^{-(c(x')-c(x))/t_k}$ resultiert. Im Gegensatz zum Metropolisalgorithmus (vgl. [MRR+53]) ist die Temperatur t_k jedoch nicht konstant, sondern wird mit einem Annealing Schedule fortgeschrieben. Diese Auswahl eines geeigneten Annealing Schedules ist entscheidend für die Performance einer Implementierung, da er über die Geschwindigkeit der Abkühlung bestimmt (vgl. [ZäBr05]).

Neben der Akzeptanzwahrscheinlichkeit $A_{ij}(t_k)$ existiert eine Generierungswahrscheinlichkeit G_{ij} . Diese stellt die Wahrscheinlichkeit für die Erzeugung einer Nachbarlösung j aus Lösung i dar. Damit resultiert für die Übergangswahrscheinlichkeit zweier Lösungen i und j : $P_{ij}(t_k) = G_{ij} * A_{ij}(t_k)$. Die Wahrscheinlichkeit einer bestimmten Lösung ist damit nur von der vorherigen Lösung abhängig. Diese „Gedächtnislosigkeit“ wird auch als Markov-Prozess bezeichnet, welcher durch die Länge der Markov-Ketten gesteuert wird.

3.2.2.2.2 Parameter

Im Folgenden werden die für den Ablauf relevanten Parameter beschrieben. Deren Ausprägung ist für die Performance des Verbesserungsverfahrens entscheidend.

- Initiallösung: Eine gültige Lösung des zugrunde liegenden Problems dient als Grundlage. Diese kann z.B. durch ein geeignetes Eröffnungsverfahren oder zufällig generiert werden.
- Initialtemperatur: Die Anfangstemperatur t_0 sollte so bestimmt werden, dass zu Beginn des Verfahrens unabhängig von der Güte der ersten ermittelten Lösungen nahezu sämtliche Lösungen mit hoher Wahrscheinlichkeit P übernommen werden und sich die Suche erst mit der Zeit einer lokalen Suche nähert. Dazu werden nach [Whit84] die Kosten einer größeren Anzahl an zufälligen Lösungen berechnet, um im Anschluss die

Standardabweichung σ bestimmen zu können. Als Initialtemperatur resultiert dann $t_0 = -3\sigma/\ln(P)$. Dabei erwies sich ein Wert von $P = 0,8$ als geeignet.

- Annealing Schedule: Der Abkühlungsverlauf während der Verbesserung wird durch den Annealing Schedule beschrieben. Hierbei werden generell zwei Typen unterschieden:
 - Fixe Verminderung: $t_k = t_{k-1} * \alpha$ mit $0 < \alpha < 1$. Um eine zu rasche Abkühlung zu vermeiden, gilt meist: $0,9 \leq \alpha \leq 0,99$
 - Adaptive Verminderung: $t_k \sim t_{k-1} / \log \psi_{k-1}$ mit ψ_{k-1} , als Funktion „eines oder mehrerer statistischer Parameter der Markov-Kette der vorhergehenden Temperaturstufe“ [ZäBr05]. In [vAL92] wird dazu die nächste Temperaturstufe als $t_{k+1} = t_k / (1 + t_k * \ln(1 + \delta) / 3\sigma_k)$ bestimmt. σ_k bezeichnet die Standardabweichung der Kostenwerte sämtlicher in Temperaturstufe k erzeugter Lösungen. Der Parameter δ gibt die Geschwindigkeit der Abkühlung an. Je kleiner δ ist, desto langsamer ist auch die Abkühlung.
- Länge der Markov-Ketten L_k : Durch diese Information wird die Anzahl anzunehmender Lösungen bestimmt, bis die Temperatur fortgeschrieben wird. Je größer die Schritte der Temperaturverminderung sind, desto länger ist die Markov-Kette. Ein üblicher Richtwert stellt die maximale Größe der Nachbarschaft dar. D.h. $L_k = |N|$.
- Abbruchkriterium: Dieses beschreibt einen beliebig zusammengesetzten Parameter der zur Terminierung des Verfahrens führt. Im Allgemeinen orientiert er sich an einer maximalen Rechenzeit und der Zeitspanne, die seit der letzten Verbesserung vergangen ist.

3.2.2.2.3 Ablauf

Der Ablauf des Simulated Annealing gliedert sich in mehrere Schritte. Die Abbildung 12 stellt dies dar.

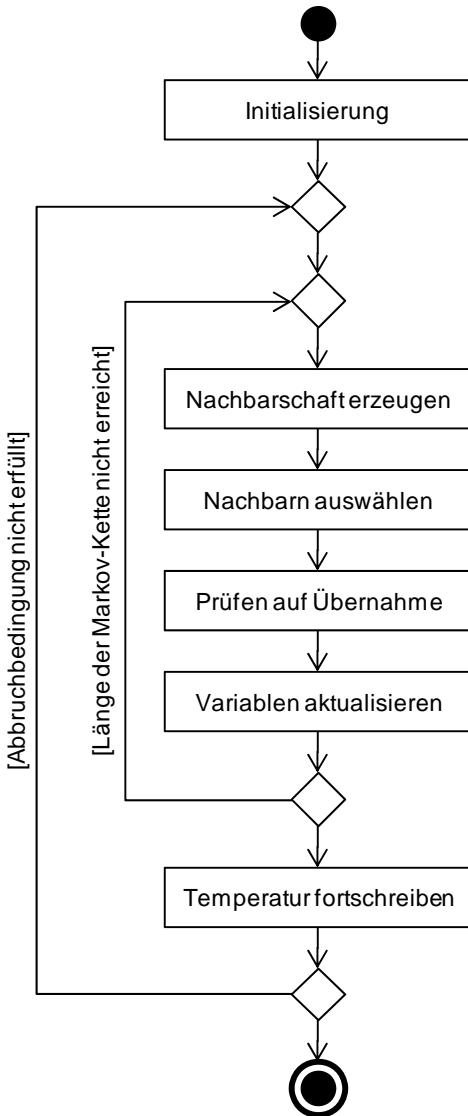


Abbildung 12: Grundsätzlicher Ablauf des Simulated Annealing

Vor dem Start erfolgt die Initialisierung der Variablen. Dies umfasst im allgemeisten Fall die Initialtemperatur, die Initiallösung sowie die Abbruchbedingung. In konkreten Implementierungen kann dies zusätzliche fallspezifische Variablen und ggf. Hilfsfunktionen beinhalten. Anschließend wird innerhalb der ersten Iteration ausgehend von der Initiallösung die Nachbarschaft berechnet (siehe Kapitel 3.2.2.1). Die einzelnen Lösungen innerhalb der Nachbarschaft werden entsprechend der Zielsetzung (Minimierung der Gesamtdurchlaufzeit, Minimierung der Gesamtkosten, etc.) durch eine Kostenfunktion bewertet.

Aus der ermittelten und bewerteten Nachbarschaft der aktuellen Lösung wird anschließend ein Lösungskandidat ausgewählt. Dazu existieren verschiedene Strategien, wie z.B. die zufällige Auswahl eines Nachbarn (random acceptance), die Auswahl des ersten Nachbarn, der besser als die aktuelle Lösung ist (first acceptance), oder den ermittelten besten Nachbarn (best acceptance)⁵⁰.

⁵⁰ Die Berechnung der Nachbarschaft verursacht in der Regel einen erheblichen Teil der Rechenzeit des Verfahrens. Aus diesem Grund ist die Wahl des besten Nachbarn häufig am sinnvollsten.

Im nächsten Schritt sind drei Fälle in Abhängigkeit der bisherigen Lösungen sowie der ausgewählten Nachbarlösung zu differenzieren:

1. Die Nachbarlösung ist besser als die bisher beste Lösung.
2. Die Nachbarlösung ist besser als die aktuelle Lösung.
3. Die Nachbarlösung ist schlechter als die aktuelle Lösung.

Für den ersten und zweiten Fall werden die Lösungen direkt übernommen. Im dritten Fall wird die schlechtere Lösung stochastisch mit der Wahrscheinlichkeit $\min\{1, e^{-(c(x')-c(x))/t_k}\}$ übernommen. Dieser Fall dient zum Verlassen von lokalen Optima.

Im letzten Schritt erfolgt eine Aktualisierung der Variablen des Verfahrens. D.h., die ggf. neue Lösung wird in entsprechenden Variablen berücksichtigt und eine neue Temperatur berechnet, falls die maximale Länge der Markov-Kette erreicht ist. Ist die Abbruchbedingung noch nicht erfüllt, startet die nächste Iteration des Verfahrens.

3.2.2.3 Tabu Search

3.2.2.3.1 Allgemeines

Tabu Search ist ein iteratives Verfahren, das analog zum Simulated Annealing auf der Nachbarschaftssuche als lokales Suchverfahren beruht [ZäBr05]. Bei der Suche nach einem Optimum wird die Historie der bisherigen Lösungsfindung einbezogen. Damit erinnert diese Vorgehensweise an das menschliche Gedächtnis, bei dem in der Entscheidungsfindung auch das in der Vergangenheit Erlebte von zentraler Bedeutung ist. Das zentrale Element dieser Metaheuristik ist die Tabu-Liste. In diese Liste wird bei jedem Zug das Komplement des letzten Zuges gespeichert und temporär für weitere Züge verboten. Dies ist notwendig, da bei diesem Verfahren die zwingende Annahme einer Nachbarschaftslösung in jeder Iteration besteht. Dabei wird nach dem Prinzip der größten Verbesserung bzw. kleinsten Verschlechterung vorgegangen (Prinzip des steepest ascent/mildest descent bei Maximierungs- bzw. steepest descent/mildest ascent bei Minimierungsproblemen). Die Wahl einer schlechteren Lösung erfolgt nur, wenn in dem Planungsschritt keine bessere Nachbarlösung existiert. Folglich wäre ohne die Tabuisierung die Gefahr von Zyklen groß, da in jeder Iteration zwingend der beste Nachbar gewählt wird. Damit sorgt diese Liste für das Verlassen eines lokalen Optimums.

Neben diesem Grundschema gibt es zudem Modifikationen zur Verbesserung der Performance. Die Aspiration erlaubt unter der Bedingung, dass der Zug zu einer Verbesserung des bisherigen Optimums führt, auch eigentlich tabuisierte Züge. Zudem existieren Ansätze, die sowohl die Intensivierung als auch die Diversifikation der Suche anstreben. Im Rahmen der Intensivierung erfolgt die Suche in vielversprechenden Regionen des Lösungsraums. Die Diversifikation verfolgt hingegen das Ziel, den Lösungsraum möglichst breit zu erkunden. Im Allgemeinen basieren diese Modifikationen auf einer speziellen Tabu-Liste. Diese speichert ähnlich zu einem Langzeitgedächtnis die Züge während der Optimierung und stellt auf diese Weise entsprechende Daten zur Analyse bereit.

3.2.2.3.2 Parameter

Bei der Tabu Search sind, unabhängig von den möglichen Modifikationen, grundsätzlich drei Parameter zu definieren.

- Art der Tabu-Liste: Es werden drei Arten des Tabu-Listen-Managements unterschieden (vgl. [DSV93]).
 - Statische, starre Tabu-Liste: Die Tabu-Liste hat eine konstante Länge und das Komplement des letzten Zuges wird ihr in jeder Iteration hinzugefügt. Die Länge sollte in Abhängigkeit der Nachbarschaftsgröße gewählt werden.
 - Statische, flexible Tabu-Liste: Die Länge der Tabu-Liste kann bei der Tabu Search zwischen einer unteren und oberen Schranke variieren.
 - Dynamische, reaktive Tabu-Liste: Die Länge der Tabu-Liste kann in jeder Iteration abhängig von der derzeitigen Situation variieren. Somit ist die Anzahl der verbotenen Züge in einer Iteration nicht vorgegeben.
- Länge der Tabu-Liste: Dies ist der entscheidende Parameter der Tabu Search. Eine zu kurze Tabu-Liste kann schnell zu Zyklen führen. Eine zu lange Tabu-Liste schränkt hingegen die Suche unnötig stark ein. Dadurch können ggf. gute Lösungen in der Nähe von bereits besuchten Lösungen nicht mehr gefunden werden [ZäBr05].
- Abbruchkriterium: Dieses beschreibt einen beliebig zusammengesetzten Parameter der zur Terminierung des Verfahrens führt. Üblicherweise orientiert er sich an einer maximalen Rechenzeit und der Zeitspanne, die seit der letzten Verbesserung vergangen ist.

3.2.2.3.3 Ablauf

Der Ablauf der Tabu Search gliedert sich in fünf Schritte. Die Abbildung 13 stellt dies dar.

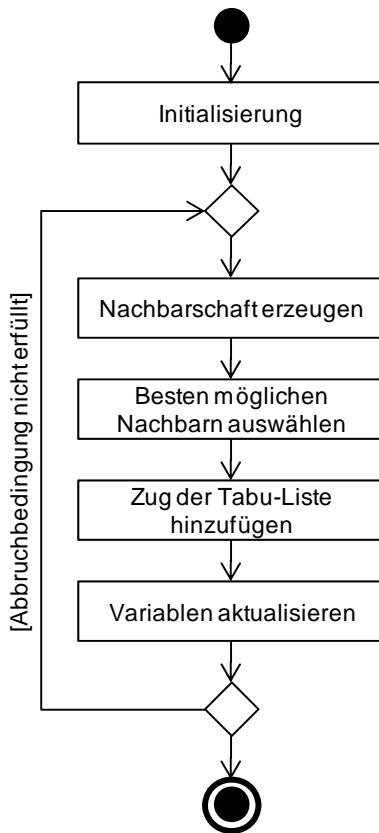


Abbildung 13: Grundsätzlicher Ablauf der Tabu Search

Analog zu dem Simulated Annealing sind in diesem Verfahren vor dem Beginn der Verbesserung die vorhandenen Variablen zu initialisieren. Dies umfasst im allgemeinsten Fall die Länge der Tabu-Liste, die Initiallösung sowie einige temporäre Zählvariablen des Verfahrens. In konkreten Implementierungen kann dies zusätzliche fallspezifische Variablen und ggf. Hilfsfunktionen beinhalten.

Ausgehend von der Initiallösung erfolgt anschließend die Generierung der Nachbarschaft (siehe Kapitel 3.2.2.1). Die Menge der durch die Nachbarschaftsfunktion erzeugten Nachbarlösungen kann in Anlehnung an [ZäBr05] in drei Teilmengen separiert werden:

1. Erlaubte Lösungen: Lösungen sind durch nicht tabuisierte Züge entstanden.
2. Verbotene, aber profitable Lösungen: Lösungen sind durch tabuisierte Züge entstanden, aber deren Güte übertrifft das bisherige Optimum.
3. Verbotene, aber nicht profitable Lösungen: Lösungen sind durch tabuisierte Züge entstanden, aber deren Güte erreicht das bisherige Optimum nicht.

Die ersten beiden Teilmengen (mit Nutzung der Aspiration) bilden zusammen die Menge der gültigen Lösungen. Diese wird nach ihrer Güte absteigend sortiert, um dann die beste Nachbarlösung zu bestimmen. Das Komplement des Zuges, das zu der gewählten besten Nachbarlösung geführt hat, wird der Tabu-Liste angehängt. Von dem gewählten Tabu-Listen-Management ist dabei abhängig, wie sich die Tabu-Liste verhält. Im z.B. einfachsten Fall (statische, starre Tabu-Liste) wird der inverse Zug dem Ende der Liste hinzugefügt. Sofern die maximale Länge der Tabu-Liste überschritten wird, wird das erste Element für die nächste Iteration freigegeben.

Im letzten Schritt erfolgt die Aktualisierung der Variablen des Verfahrens. D.h., die neue Lösung wird den Variablen der aktuellen Lösung zugewiesen. Sofern ein neues Optimum gefunden wurde, wird die jetzt aktuelle Lösung zusätzlich in den Variablen der bisher optimalen Lösung gespeichert. Nachdem auch die Zählvariablen (z.B. Iterationszähler) inkrementiert wurden, erfolgt die Analyse der Abbruchbedingung. Falls dies zutrifft, terminiert das Verfahren und liefert das aktuelle Optimum als Ergebnis. Andernfalls startet die nächste Iteration.

3.2.2.4 Metaheuristiken für den Flexible Flow Shop

In diesem Kapitel werden bekannte Verfahren zur Planung eines Flexible Flow Shop unter Nutzung von Metaheuristiken vorgestellt. Es werden primär Verfahren betrachtet, die auf Simulated Annealing und Tabu Search basieren.⁵¹

Erste Verfahren unter Nutzung von Tabu Search zur Planung eines Flow Shop werden von Widmer und Hertz (vgl. [WiHe89]) sowie Taillard (vgl. [Tail90]) entwickelt. Das Ziel dieser Verfahren ist die Suche eines Permutationsplans, der die Gesamtdurchlaufzeit minimiert. Die Überlegenheit ihrer Verfahren gegenüber einfachen Eröffnungsverfahren (z.B. dem NEH-Verfahren (vgl. [NEH83])) weisen die Autoren in mehreren experimentellen Untersuchungen nach. Für das gleiche Problem stellen Ben-Daya und Al-Fawzan (vgl. [BeAl98]) ein Verfahren vor, das eine verbesserte Generierung von Nachbarn sowie entsprechender Kandidaten für den nächsten Verbesserungsschritt umfasst. Hierzu entwickeln sie eine Strategie mit einer kombinierten Diversifikation und Intensivierung. Nowicki und Smutnicki (vgl. [NoSm98]) entwickeln ein Verfahren für einen Flexible Flow Shop mit identischen parallelen Produktionslinien. In früheren Arbeiten konstatierten die Autoren (vgl. [NoSm96b], [NoSm96a]) eine Überlegenheit von insert-Zügen, weshalb sie nur auch in dieser Arbeit zur Bildung von Nachbarn Anwendung finden. In ihrem Suchprozess nutzen sie eine kurzfristige Tabu-Liste zur Vermeidung von Zyklen und eine längerfristige Liste, um zu ausgewählten guten Lösungen zurückzukehren. Ziel von diesem Verfahren sowie deren von Norman (vgl. [Norm99]) sowie Wardono und Fathi (vgl. [WaFa04]) ist die Minimierung der Gesamtdurchlaufzeit. In dem Verfahren von Norman (vgl. [Norm99]) wird ein um begrenzte Puffer und reihenfolgeabhängige Rüstzeiten erweitertes Problem betrachtet. Begrenzte Puffer unterstellen auch Wardono und Fathi (vgl. [WaFa04]), wobei im Gegensatz zu der Arbeit von Norman zudem identische parallele Produktionslinien berücksichtigt werden. Kann ein Auftrag im Anschluss an seiner Bearbeitung auf einer Produktionsstufe mangels ausreichender Kapazität nicht dem folgenden Puffer übergeben werden, verweilt der Auftrag auf der Produktionslinie und blockiert diese. Kulcsár et. al. (vgl. [KEH07]) nutzen Tabu Search um ein Neu- und auch Änderungsplanungsproblem zu lösen. Hierbei berücksichtigen sie verschiedenartige parallele Produktionslinien, reihenfolgeabhängige Rüstzeiten, unterschiedliche Kapazitäten der Produktionslinien in bestimmten Zeitabschnitten und eine mehrdimensionale Zielsetzung. Es werden die Anzahl

⁵¹ Verfahren auf Basis genetischer Algorithmen beschreiben z.B. [Reev95], [CGT95], [JOI+02], [KuAs04], [BeBi04], [SeUl04], [RuMa06] sowie [QuKu07]. Neben diesen „reinen“ Metaheuristiken gibt es zudem hybride Verfahren. Hierbei werden die Ideen der einzelnen Verfahren kombiniert. Beispiele hierzu finden sich bei [YaRe97], [Glov98] sowie [Stüt98]. Als Basis dieser Darstellung dienen die Arbeiten von [Wang05] und [Quad04].

verspäteter Aufträge, die Summe der gesamten Verspätung, die maximale Verspätung, die Anzahl an Rüstvorgängen sowie die maximale Fertigungszeit minimiert.

Frühe Arbeiten, die sich mit den Möglichkeiten von Simulated Annealing beschäftigen, stammen von Osman und Potts (vgl. [OsPo89]) sowie Ogbu und Smith (vgl. [OgSm90], [OgSm91]). Mit diesen Verfahren können in zumutbarer Zeit Pläne erstellt werden, deren Lösungsgüte (ausgedrückt durch die Gesamtdurchlaufzeit) nahe dem Optimum liegt (vgl. [Near04]). In [OsPo89] beschreiben die Autoren für einen Flow Shop und einem Permutationsplan als Ausgangsplan vier unterschiedliche Varianten zur Nachbarschaftssuche. Sie konstatieren, dass aus insert-Zügen bessere Ergebnisse resultieren als aus swap-Zügen. Ogbu und Smith (vgl. [OgSm90], [OgSm91]) entwickeln einen neuen Ansatz zur Bestimmung der Akzeptanzwahrscheinlichkeit und der Nachbarschaft sowie ein Verfahren zur Erstellung eines guten Ausgangsplans. Für das Ziel der Minimierung der Durchlaufzeit schlagen Sridhar und Raendran (vgl. [SrRa93]) ein Verfahren vor, in dem sie drei unterschiedliche Arten der Nachbarschaftsgenerierung betrachten. Analog zu den vorherigen Verfahren werden keine parallelen Produktionslinien berücksichtigt. Ishibuchi et. al. (vgl. [IMT95]) entwickeln zwei Algorithmen mit dem Ziel der Minimierung der Gesamtdurchlaufzeit. In diesen konzentrieren sich die Autoren auf die Generierungswahrscheinlichkeit. In dem Standard-Verfahren von Simulated Annealing wird ein Nachbar von einer aktuellen Lösung zufällig generiert. D.h., dass alle Nachbarn die gleiche Wahrscheinlichkeit haben. Um jedoch möglichst schnell gute Lösungen zu erreichen, bewerten die Autoren in ihrem Verfahren in jedem Schritt die potentiellen Nachbarn und wählen den besten aus. Das gleiche Ziel wie Ishibuchi et. al. verfolgt Riane (vgl. [Rian98]). Die Nutzung von Simulated Annealing beschränkt sich bei ihm auf die erste Produktionsstufe, wobei auch Rüstzeiten Berücksichtigung finden. Die anderen Produktionsstufen werden auf Basis der entsprechenden Freigabezeitpunkte der Aufträge der Vorgängerproduktionsstufe beplant. Low (vgl. [Low05]) entwickelt ein Verfahren für einen Flexible Flow Shop mit verschiedenartigen parallelen Produktionslinien und unter Berücksichtigung von Rüstzeiten und Abrüstzeiten mit der Zielsetzung minimaler Gesamtdurchlaufzeiten. Der Fokus liegt auch bei dem Verfahren auf der Generierung guter Nachbarn und zusätzlicher Abbruchkriterien zur Beschleunigung der Lösungsfindung. Für ein System mit identischen parallelen Produktionslinien entwickeln Jin, Yang und Ito (vgl. [JYI06]) zwei Verfahren. Das Problem besteht in der Bestimmung minimaler Gesamtdurchlaufzeiten, wobei unbegrenzte Zwischenpuffer und reihenfolgeunabhängige Rüstzeiten unterstellt werden. Zu Beginn beider Verfahren wird das Planungsproblem entsprechend vorhandener Produktionsstufen in einzelne Teilprobleme zerlegt. Anschließend wird eine initiale Auftragssequenz mit Hilfe von Prioritätsregeln (beim ersten Verfahren mit FAM (first available machine) und LRPT (longest remaining processing time), beim zweiten mit LFM (last free machine) und LRPT) erstellt und danach iterativ verbessert. Tabriz et. al. (vgl. [TZV09]) betrachten ein Produktionssystem mit identischen parallelen Produktionslinien und reihenfolgeabhängigen Rüstzeiten. Zur Lösung wird auf der ersten Produktionsstufe ein Plan mit Simulated Annealing erzeugt. Die Zuweisung der Aufträge auf die anderen Produktionsstufen erfolgt durch Nutzung verschiedener Regeln.

Diese Zusammenfassung verdeutlicht, dass bereits für zahlreiche Planungsprobleme mit Tabu Search und Simulated Annealing gute Ergebnisse gefunden werden konnten. Analog zu den

Eröffnungsverfahren ist auch hier eine Zunahme der Komplexität der jeweiligen Planungsprobleme zu konstatieren. Allerdings betrachten die meisten Verfahren als Zielsetzung zeitliche Aspekte. Ein Verfahren zur Lösung der in Kapitel 2 dargestellten Problemstellung konnte nicht identifiziert werden.

3.2.3 Methoden der reaktiven Planung

In diesem Kapitel werden bekannte Strategien zum Umgang mit Störungen⁵² (zur Definition von „Störung“ siehe Kapitel 2.1.3.2) betrachtet sowie deren Einsatz im Kontext des Flexible Flow Shop analysiert.

3.2.3.1 Strategien zum Umgang mit Störungen

Zum Umgang mit Störungen werden grundsätzlich zwei Arten von Strategien differenziert. Es wird entweder ein alter Ablaufplan vollständig verworfen und ein unter Berücksichtigung aller Informationen komplett neuer Ablaufplan erstellt oder der alte Ablaufplan als Ausgangsbasis weiterer Ablaufpläne genutzt. Es ist offensichtlich, mit dem Ziel der Generierung eines optimalen Ablaufplans die Erstellung eines neuen Ablaufplans zu favorisieren. Hierbei können jedoch unterschiedliche Schwierigkeiten auftreten. So ist der Forderung nach einer optimalen Lösung praxisrelevanter Problemgrößen schwer nachzukommen (siehe Kapitel 2.2.5). Zudem kann ein häufiges Verwerfen von Ablaufplänen eine Instabilität des gesamten betrieblichen Ablaufs mit ggf. zusätzlichen Kosten bewirken. Folglich verursachen nervöse Ablaufpläne nicht nur Verwirrung im Produktionssystem, sondern können auch Nacharbeit von Versorgungsabteilungen (z.B. der Werkslogistik) verlangen (vgl. [CuPe05]). Aus diesem Grund werden im Weiteren Maßnahmen diskutiert, mit denen ein originärer Ablaufplan bei der Beseitigung von Störungen einbezogen wird. Diese Maßnahmen umfassen Strategien zur Störungsbeseitigung und zur Störungsabwehr (vgl. [Pati01]).⁵³

Im Rahmen der Störungsabwehr soll das Produktionssystem so gestaltet werden, dass die potenziellen Störfaktoren eliminiert oder aber zumindest deren Häufigkeit vermindert werden (vgl. [Pati01]). Dadurch sollen robuste Ablaufpläne entstehen, die nach Möglichkeit nicht geändert werden müssen. Durch ausschließlich präventive Maßnahmen können jedoch nicht sämtliche Störungsursachen eines Unternehmens behandelt werden. Es hat sich zwar gezeigt, dass Unternehmen, die eine ursachenorientierte Beseitigung von Störungen mit einem (möglicherweise) einmaligen, ursachenbezogenen Analyseaufwand betreiben, einen geringeren Anteil störungsbedingter Aktivitäten aufweisen (vgl. [Heil95]), aber es sind trotzdem mit dieser Strategie große Nachteile verbunden. So erfordert zum einen die Vielfalt an Störungsursachen einen hohen Analyseaufwand und zum anderen liegen manche Ursachen außerhalb des Einflussbereichs des Unternehmens (vgl. [Fisc07]). Für diese sind dann ebenfalls Maßnahmen der Störungsbeseitigung zu definieren.

⁵² In dieser Arbeit wird dies unter dem Begriff der reaktiven Planung zusammengefasst. In der Literatur synonym verwendete Begriffe sind „Entstörung(-smanagement)“, „Änderungsplanung/-management“, „Umplanung“, „Umdisposition“, „Anpassungsplanung“ sowie „Rescheduling“.

⁵³ In der Literatur gibt es keine einheitliche Kategorisierung. So unterscheidet Neuhaus (vgl. [Neuh08]) generell zwischen prädikativer, robuster und reaktiver Planung.

Das Ziel der Störungsbeseitigung ist ein adäquater Umgang mit Störungen und damit gültige und wirtschaftliche Pläne. Nach Wu und Li (vgl. [WuLi95]) gestaltet sich dieser Prozess in drei Schritten. Im ersten Schritt wird das Ausmaß einer aufgetretenen Störung bewertet. Nur wenn ein bestimmtes Toleranzspektrum überschritten ist, wird die Beseitigung der Störung verfolgt. Hierzu wird im zweiten Schritt eine geeignete Strategie bestimmt. Dies stellt nach Vieira et. al. (vgl. [VHL03]) den schwierigsten Teil dar. Im letzten Schritt wird der originäre Plan aktualisiert bzw. ein neuer Plan erstellt.

In der Störungsbeseitigung werden die Strategien Ignoranz, Antizipation und Reaktion differenziert. Die Ignoranz ist analog zur Störungsabwehr eine Strategie zur Vermeidung von Planänderungen. Hierbei werden nur als „relevant“ markierte Störungen betrachtet. Die für das jeweilige Unternehmen weniger wichtigen werden ignoriert und bedürfen keiner weiteren Maßnahme. (Ein Beispiel könnte eine Bedarfsterminverschiebung um wenige Minuten sein.)

Wird die Strategie der Antizipation verfolgt, so wird unterstellt, dass

- das Auftreten von Störungen nicht vermieden werden kann,
- der potentielle Störungsschaden niedriger sein wird als die Kosten der Prävention,
- kein deterministischer oder mit einer hohen Wahrscheinlichkeit versehener Störungszusammenhang herstellbar ist resp. dies aus wirtschaftlichen Überlegungen nicht sinnvoll erscheint (vgl. [Pati01], [Heil95]).

Im Rahmen der Antizipation erfolgt die Planung und Bereitstellung von Potentialen zur Verringerung bzw. Vermeidung von Störungswirkungen (Systemreserven wie Mengen-, Zeit- und Kapazitätspuffer werden *ex ante* eingeplant). Diese einzuplanenden Präventivpotentiale in Produktionssystemen ermöglichen die Unsicherheit zu beherrschen und sichern somit die Einhaltung von Plänen (vgl. [Pati01], [Fisc07]). Durch dieses Präventivpotential existiert eine hohe Flexibilität im Störfall, d.h., Störungswirkungen können sofort beseitigt werden. Allerdings erfolgt bei der Bestimmung des Umfangs dieser Reserven nur die Berücksichtigung der zu diesem Zeitpunkt vorliegenden Informationen. Somit können nur auf Basis dieser (unzuverlässigen) Informationen eine Antizipation der Störungswirkungen sowie die Konzeption entsprechender Präventivmaßnahmen erfolgen. Außerdem sind neben dem zufordernden relativ stabilen Informationsstand die mit dem Potentialaufbau verbundenen zusätzlichen Material-, Personal- sowie Kapazitätskosten zu berücksichtigen. Diese Kosten entstehen aufgrund der suboptimalen Planung auch dann, wenn keine Störung eintritt (vgl. [Fisc07]). Innerhalb der Strategie der Antizipation werden zudem alternative Planungsansätze betrachtet. Dabei werden die Störungen nicht innerhalb der Planungsdaten, sondern bereits in einem geeigneten Planungsverfahren berücksichtigt. Ein Beispiel für einen solchen Planungsansatz ist die flexible Planung. Bei dieser werden Pläne nicht ohne Rücksicht auf die Umweltentwicklung verbindlich festgelegt, sondern Eventualpläne (Schubladenpläne) ermittelt (vgl. [Laux05]). Dies geschieht auf der Basis von vier charakteristischen Prämissen (vgl. [Zele99]):

- Von vorherein können alle gegenwärtig und zukünftig für möglich gehaltenen Produktionssituationen als diskrete und endlich viele „Umweltzustände“ erfasst werden (Zustandsprämissen).

- Für jeden möglichen Umweltzustand kann von vornherein eine konstante Eintrittswahrscheinlichkeit zugeordnet werden (Wahrscheinlichkeitsprämisse).
- Alle Anpassungshandlungen, die aufgrund des Flexibilitätspotentials eines Produktionssystems ausgeführt werden können, lassen sich von vornherein identifizieren und als diskrete und endlich viele situationsspezifische „Aktivitäten“ darstellen (Aktivitätsprämisse).
- Jede Aktivität kann bereits im Voraus mit demjenigen Beitrag bewertet werden, den ihre Ausführung zur Erreichung des (ökonomischen) Zielsystems des Entscheidungsträgers leisten würde (Bewertungsprämisse).

Sofern diese Prämissen erfüllt sind, kann mit den Entscheidungstechniken der flexiblen Planung eine optimale Menge von Eventualplänen erstellt werden. Jeder Eventualplan empfiehlt für genau eine Produktionssituation, die von der erwarteten Entwicklung des Planungsumfelds abhängt, eine situationsspezifische, optimale Anpassungshandlung (vgl. [Zele99]). Da jedoch mit zunehmender Komplexität der Problemstellungen auch die Zahl möglicher Produktionssituationen steigt, was aufgrund der kombinatorischen Explosion einen hohen Rechenaufwand bedingt, können Eventualpläne nur für kleine Problemstellungen erstellt und analysiert werden. Es ist zudem schwierig, sämtliche potentielle Entwicklungen des Planungsumfelds bereits von vornherein vollständig und explizit einzuplanen sowie mit den entsprechenden (richtigen) Wahrscheinlichkeiten zu berücksichtigen. Das Flexibilitätspotential eines realen Produktionssystems kann daher mit der flexiblen Planung nicht vollständig erfasst werden. Aufgrund dieser Schwierigkeiten konnte sich die flexible Planung (in ihrer reinen Form) in der betrieblichen Praxis bislang nicht durchzusetzen (vgl. [Zele99]).

Die Reaktion ist die dritte Strategieoption der Störungsbeseitigung. Entstehen Abweichungen der Ist- von den Planwerten, die über einem ggf. vorgegebenen Toleranzspektrum liegen, können entweder alle oder einzelne Plan-Werte eines vormals bestimmten Ursprungsplans zur Störungsbeseitigung aktualisiert werden. In der Neuplanung werden alle Plan-Werte des Ursprungsplans aufgehoben und ein vollständig neuer Plan erstellt. In der Umplanung wird hingegen der Ursprungsplan korrigiert (vgl. [Pati01]). Diese Strategie verfolgt das Ziel, dass ein durch eine Störung ungültig gewordener Plan derart modifiziert wird, dass das mit dem ursprünglichen Plan verfolgte Ergebnis trotz Störsituation erreicht werden kann. Die Suche nach einer optimalen Lösung tritt eher in den Hintergrund. Es soll vielmehr möglichst schnell ein konsistenter Plan erstellt werden. „*Wichtig ist daher, dass man problemspezifisch geeignete Verfahren findet, die relativ häufig eine annehmbare Lösung finden*“ [Sau02]. Eine besondere Form ist das Match-Up-Scheduling (wiedereinregelnde Planung), welches in Produktionsystemen mit parallelen Produktionslinien eingesetzt werden kann (vgl. [Neuh08]). Hierbei werden unter Ausnutzung vorhandener Leerzeiten die Start- und Endzeitpunkte der Lose so geändert, dass ab einem bestimmten Zeitpunkt (dem Match-Up-Point) der alte Plan wieder verwendet werden kann.

Im Folgenden wird aufgezeigt, welche möglichen Maßnahmen zu ergreifen sind, um eine Umplanung vorzunehmen. Dabei soll zwischen Veränderungen der Kapazitätsnachfrage und Veränderungen des Kapazitätsangebots unterschieden werden. Das Kapazitätsangebot kann qualitativ oder quantitativ durch die Umplanung verändert werden, wohingegen die Kapazi-

tätsnachfrage zusätzlich auch terminlich (Anpassung der Start- und Endtermine der Lose) angepasst werden kann. (vgl. [Pati01])

Durch die Maßnahme des Right Shifting werden der Start oder das Ende der Bearbeitung von Losen verspätet ausgeführt. Dies wird durch das folgende Beispiel verdeutlicht. Abbildung 14 zeigt den Ursprungsplan von fünf Fertigungsaufträgen (A, ..., E) auf drei Produktionsstufen (Prod.St.1, ..., Prod.St.3) mit jeweils einer Produktionslinie (Prod.L.). Auf der Produktionsstufe 2 tritt eine Störung auf, die die originär geplante Ausführung von Los A der zweiten Produktionsstufe verhindert.

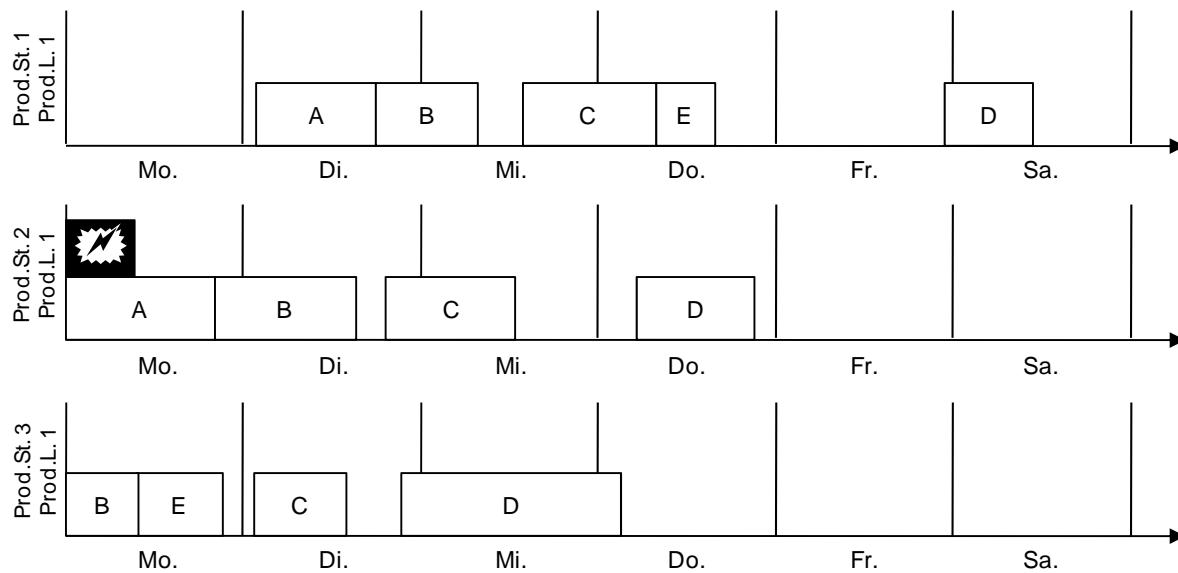


Abbildung 14: Beispiel eines Ursprungsplans

Indem nun das Los A nach rechts verschoben wird, bleibt die Ausführungsreihenfolge unverändert (Abbildung 15), aber es muss auch eine Verspätung der Lose A, B, C und E auf Produktionsstufe 1 hingenommen werden, was ein grundsätzliches Problem beim einfachen Right Shifting darstellt. Lediglich das Los D sowie die Lose der Produktionsstufe 3 (grau markiert) werden nicht beeinflusst.

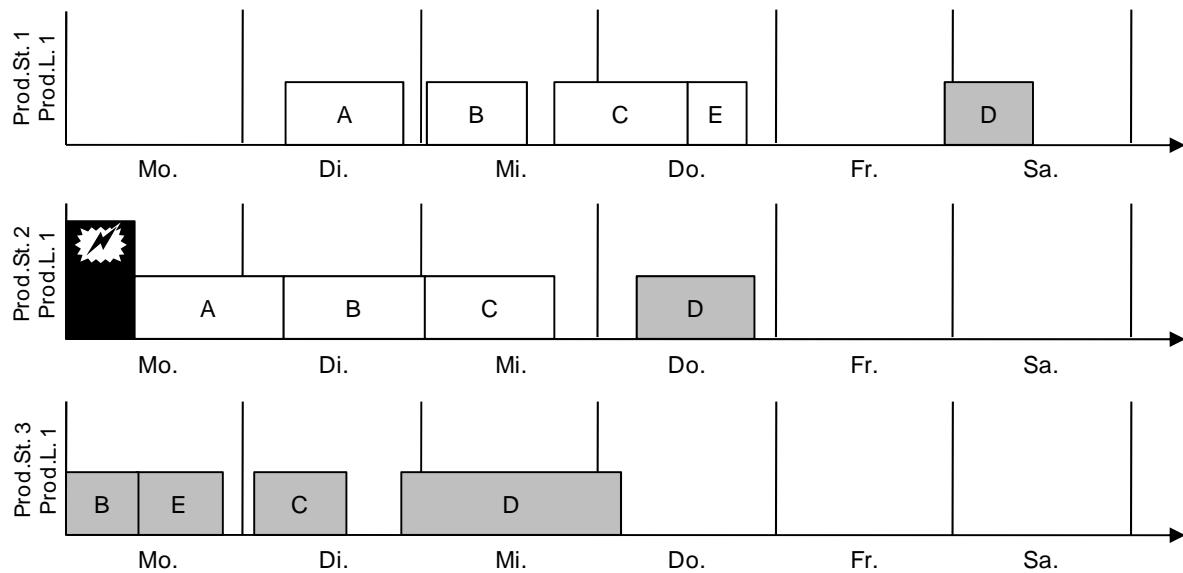


Abbildung 15: Einfaches Right Shifting

Eine weitere Maßnahme ist das Right Shifting and Jumping. Bei der einfachen Form wurde dargestellt, dass aufgrund einer einzigen Störung fast alle Lose involviert sind und sich verspäten. Das Right Shifting and Jumping berücksichtigt diesen Aspekt und verändert durch das nach hinten schieben oder springen nur diejenigen Lose, die direkt betroffen sind. Dabei springen die von der Störung betroffenen Lose auf ihrer eingeplanten Produktionslinie über diejenigen Lose, die ihnen nachfolgen. Somit soll der Ursprungsplan weitestgehend beibehalten werden. Diese Umplanung wird durch Abbildung 16 dargestellt. Bei dieser Umplanungsmaßnahme springt das Los C auf der Produktionsstufe 1 über das Los E, so dass für dieses keine neuen Zeiten ermittelt werden müssen. Durch dieses Vergehen bleibt somit mehr vom Ursprungsplan erhalten (die grau markierten Lose spiegeln den Ursprungsplan wieder).

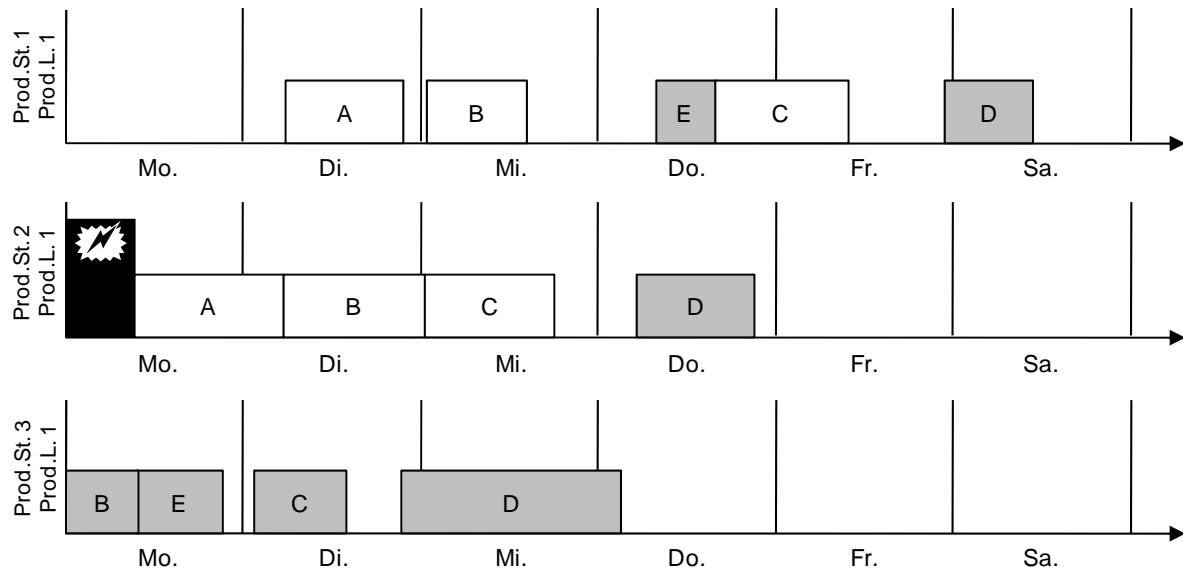


Abbildung 16: Right Shifting and Jumping

Erfolgt hingegen eine Änderung der Ausführungsreihenfolge innerhalb der feinterminierten Lose, müssen neue Plan-Termine ermittelt werden. Dadurch werden die geplanten Start- und Endtermine der jeweiligen Lose im Vergleich zum Ursprungsplan vorgezogen oder weiter nach hinten verlagert (vgl. [Pati01]). Auch diese Möglichkeit einer Umplanung wird auf Grundlage des obigen Beispiels (siehe Abbildung 14) abgebildet. Hierbei wurde das Los E auf der Produktionsstufe 1 nach ganz vorne (links) umgeplant.⁵⁴

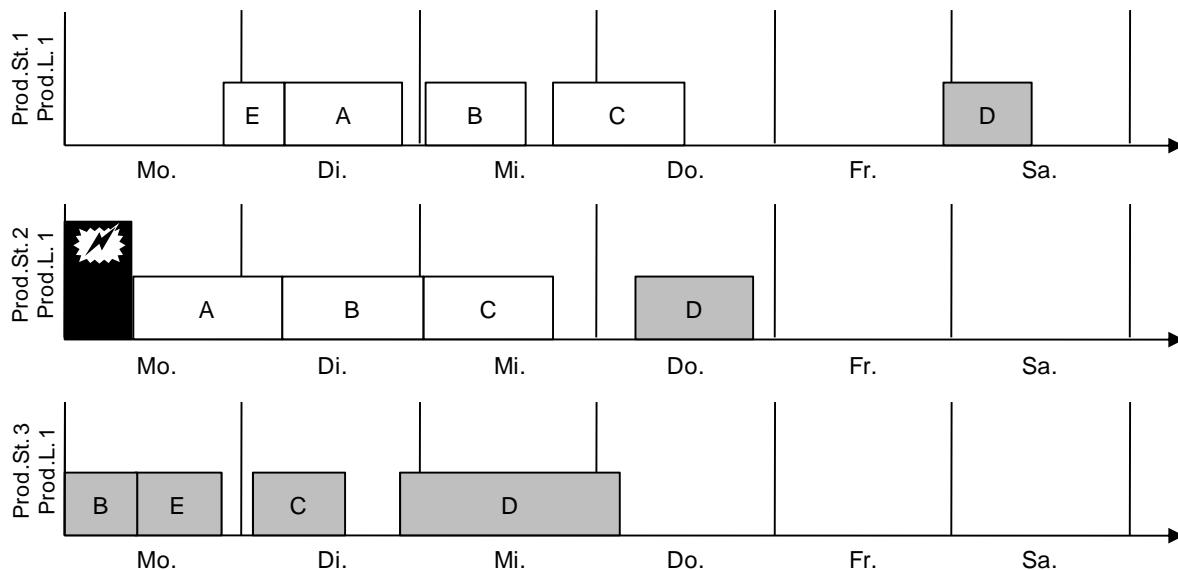


Abbildung 17: Änderung der Ausführungsreihenfolge

Eine Maßnahme zur Anpassung der qualitativen Kapazitätsnachfrage besteht – sofern technisch möglich – durch die Nutzung alternativer Fertigungsverfahren (z.B. durch die Verwendung alternativer Materialien) (vgl. [Pati01]).

Wenn die Plan-Werte hinsichtlich veränderter Bedarfsmengen gegenüber dem Ursprungsplan modifiziert werden müssen, sind dies Maßnahmen zur Veränderung der quantitativen Kapazitätsnachfrage. Als geeignete Maßnahme bietet sich hierzu u.a. die Splitting an. Hierbei werden Plan-Bedarfsmengen gegenüber der Ursprungsplanung um bestimmte Mengen reduziert, so dass diese jeweils wieder einem neu zu generierenden Fertigungsauftrag zugewiesen werden können (vgl. [Pati01]). Die Abbildung 18 zeigt ein Beispiel, bei dem ein Los, welches laut Ursprungsplan auf einer Produktionslinie gefertigt werden sollte, auf drei Produktionslinien gesplittet wird.

⁵⁴ Eine andere Möglichkeit der Umplanung ohne Änderung der Ausführungsreihenfolge besteht im Überlappen der Lose. Dabei wird die Bearbeitung einzelner Lose vor dem im Ursprungsplan vorgesehenen Endtermin der vorangehenden Lose vorgezogen (vgl. [Pati01]). Sobald eine definierte Teilmenge des Loses bereits gefertigt wurde, kann mit dem Nachfolgelos begonnen werden. Diese Maßnahme ist jedoch im Kontext eines Flexible Flow Shop unzulässig und wird aus diesem Grund nicht weiter detailliert (siehe Kapitel 2.2).

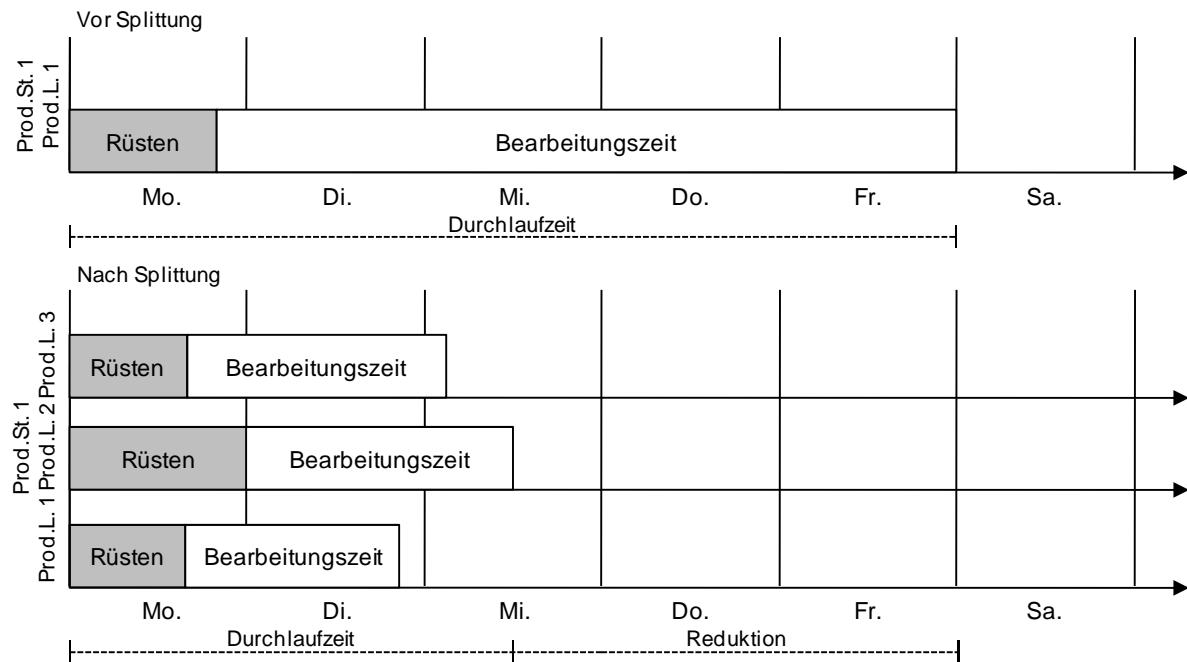


Abbildung 18: Durchlaufzeitverkürzung durch Splitting⁵⁵

In diesem Beispiel wurde durch die Splitting die gesamte Durchlaufzeit halbiert, was allerdings durch drei Produktionslinien erreicht wurde. Das Splitten erfordert jedoch mehrfaches Rüsten, was zur Folge hat, dass die verbleibende Gesamtkapazität verringert wird und zusätzlich eine Vervielfachung der Rüstkosten entsteht. Hinzu kommt, dass zusätzlicher administrativer Aufwand (z.B. seitens der Werkslogistik) erforderlich ist.

Zudem existieren Maßnahmen zur Veränderung des qualitativen und quantitativen Kapazitätsangebots (vgl. [Pati01]). Die Maßnahmen zur Veränderung des qualitativen Kapazitätsangebots sind allerdings nicht mehr in die operative PPS einzuordnen. Vielmehr handelt es sich dabei um längerfristig ausgerichtete Entscheidungen, die eher der taktischen oder der strategischen Produktionsplanung zuzuordnen sind. Hierzu zählen bspw. die Beschaffung von Produktionslinien und die Einstellung von neuem Personal. Eine Erhöhung oder Reduktion des quantitativen Kapazitätsangebots kann bspw. durch die Stilllegung oder Wiederinbetriebnahme von Produktionslinien oder einer Veränderung des jeweiligen Leistungsgrads erfolgen. Eine weitere Möglichkeit zur Steigerung des Kapazitätsangebots besteht zudem in der Fremdvergabe von Aufträgen (vgl. [AAO01]).

Neben diesen Strategien zur Auflösung gibt es unterschiedliche Herangehensweisen bzgl. der zeitlichen Reaktion auf Störungen. Hierzu werden in der Literatur drei unterschiedliche Varianten unterschieden (vgl. [VHL03], [SaBa00]). Bei periodischen Strategien werden die einzelnen Pläne in regelmäßigen Zeitabschnitten aktualisiert. Basis des jeweils neuen Plans sind der originäre Plan, aufgetretene Störungen sowie ein durch Fortschreiten des rollierenden Planungshorizontes neuer Zeitabschnitt. Nach Vieira et. al. (vgl. [VHL03]) eignet sich diese Strategie besonders in Unternehmen mit einer periodischen Betriebsdatenerfassung. Dementspre-

⁵⁵ In Anlehnung an [Kurb05].

chend besitzen die Pläne eine größere Stabilität als bei einer kontinuierlichen Planerstellung. Eine Reaktion auf eine signifikante Störung ist jedoch nicht zeitnah möglich. Bei einer ereignisorientierten Strategie wird ein Verfahren ausgelöst, wenn eine Störung eintritt. Neben den Erfordernissen einer sehr schnellen Datenerfassung und vielen Computerressourcen impliziert diese Art der Reaktion besonders bei Unternehmen mit einem hohen Störaufkommen einen sehr nervösen Plan und ggf. ein permanentes Erstellen von Plänen. Eine hybride Strategie stellt eine Kombination beider Verfahren dar. Ein Plan wird folglich in periodischen Abständen neu erstellt oder wenn eine signifikante Störung eintritt.

3.2.3.2 Reaktive Verfahren für den Flexible Flow Shop

In der Literatur existieren vereinzelte Verfahren, die Änderungen in einen bestehenden Plan eines (Flexible) Flow Shop integrierten. Guo und Nonaka (vgl. [GuNo99]) betrachten einen dreistufigen Flow Shop mit dem Ziel einer minimalen Gesamtdurchlaufzeit. Sie unterstellen, dass in einem Zeitpunkt nur eine Störung (Defekt einer Produktionslinie) auftreten kann. Ob diese Störung eine Änderungsplanung auslöst, ist abhängig von einem Trigger-Wert und der Zeit zur Behebung einer Störung. Ist letztere größer als der Trigger-Wert, wird die Änderungsplanung gestartet. Zudem führen die Autoren eine Methode ein, die aus verschiedenen Plänen den Robustesten auswählt. In dem Verfahren von Akturk und Gorgulu (vgl. [AkGo99]) wird ein möglichst schnelles Wiedereinregeln auf den Ursprungsplan und eine Minimierung verspäteter Aufträge angestrebt. Sie begründen dies damit, dass die ursprüngliche Ressourcenplanung auf Basis des Ursprungsplans erstellt wird und eine Abweichung folglich eine Instabilität impliziert. Zur Wiedereinregelung wird nach einem Ausfall einer Produktionslinie für jede Produktionslinie ein Match-Up-Punkt (dieser stellt den Zeitpunkt dar, ab dem der korrigierte Plan dem Ursprungsplan entspricht) definiert. Anschließend wird für den Zeitpunkt der Störung und dem jeweiligen Match-Up-Punkt mit Hilfe einer Branch & Bound Methode der Zeitabschnitt umgeplant. Kulcsár et. al. (vgl. [KEH07]) nutzen Tabu Search, um neben einem Neu- auch ein Änderungsplanungsproblem zu lösen. Hierbei berücksichtigen sie verschiedenartige parallele Produktionslinien, reihenfolgeabhängige Rüstzeiten, unterschiedliche Kapazitäten der Produktionslinien und eine mehrdimensionale Zielsetzung. Zur Messung der durch die Änderungsplanung verursachten Pläne führen die Autoren Qualitätsindizes ein, die einen Vergleich des Ursprungsplans mit dem jeweiligen neuen Plan erlauben. Die Indizes umfassen bspw. die Anzahl verspäteter Aufträge, die Anzahl an Aufträgen mit veränderten Bearbeitungsstationen sowie die Anzahl an Rüstvorgängen.⁵⁶

Es ist zu konstatieren, dass die Integration von Störungen in Ablaufpläne eines Flexible Flow Shop in der Forschung bislang keine große Aufmerksamkeit geschenkt wurde. Besonders die Lösung von komplexeren Planungsproblemen, wie in dieser Arbeit unterstellt, wurde weitestgehend vernachlässigt.

⁵⁶ Es existieren zudem Verfahren, die abweichend von denen in dieser Arbeit betrachteten Verfahren Änderungen integrieren. So erstellen auf Basis eines genetischen Algorithmus z.B. Gholami et. al. (vgl. [GZA09]) und auf Basis von ant colony optimization z.B. Hu et. al. (vgl. [HYY+05]) einen Plan.

3.3 Bewertung von Planungsverfahren

Benchmarking ist allgemein ein systematisches Vergleichen von Objekten (z.B. Prozessen, Erzeugnissen) anhand festgelegter Merkmale. Im Kontext der Ablaufplanung beschreibt es Methoden und Kriterien, die zur Beurteilung von Planungsverfahren benutzt werden können. Zur Bewertung von Heuristiken wird dazu häufig eine Zweiteilung in outputorientierte und inputorientierte Merkmale vorgenommen. Die inputorientierten Kriterien betrachten die einzusetzenden Ressourcen (sie werden daher auch als ressourcenorientierte Kriterien bezeichnet), während bei outputorientierten Kriterien das Ergebnis des Einsatzes einer Heuristik im Vordergrund steht (ergebnisorientierte Beurteilungskriterien) (vgl. [BDS04]). In der Literatur werden eine Vielzahl unterschiedlicher input- und outputorientierter Kriterien genannt. Es ist jedoch zu konstatieren, dass den Kriterien Rechenaufwand und Lösungsqualität die größte Beachtung geschenkt wird.⁵⁷ Zur Bewertung der Güte eines reaktiven Planungsverfahrens in einem dynamischen Planungsumfeld ist zudem die Stabilität von großer Bedeutung (vgl. [Neuh08]).

Rechenaufwand

Zur Bestimmung des Rechenaufwands können theoretische und experimentelle Untersuchungen vorgenommen werden. Im Rahmen der theoretischen Analyse wird die Komplexität anhand der O-Notation ausgedrückt. Da diese den Rechenaufwand im schlechtesten Fall (Worst-Case) beschreibt, gibt diese Abschätzung den realistisch zu erwartenden Aufwand in der Regel nicht wieder (vgl. [Heue04]). Experimentelle Untersuchungen ermitteln die Rechenzeit anhand von Testdaten. Die Ermittlung wird jedoch stark von der eingesetzten Hardware, dem verwendeten Betriebssystem und der zur Implementierung genutzten Programmiersprache beeinflusst. Ein Vergleich von Heuristiken verschiedener Autoren auf Basis der Rechenzeit ist dadurch nur eingeschränkt möglich. Für eine Analyse der Auswirkungen hinsichtlich unterschiedlicher Problemgrößen ist die Rechenzeit hingegen gut geeignet (vgl. [Schm99]).

Lösungsqualität

Die Lösungsqualität beschreibt, wie weit die Lösung einer Heuristik von einer optimalen Problemlösung abweicht. Zur Ermittlung dieser schlägt Berens die Ermittlung einer normierten Lösungsqualität vor (vgl. [Bere92]). Als Anforderung an heuristische Verfahren bzgl. ihrer Qualität nennt er eine „*möglichst gute Annäherung an den optimalen Zielfunktionswert*“ [Bere92]. Die Qualität ist definiert als das Ausmaß, in dem ein Betrachtungsobjekt die ihm

⁵⁷ Weitere in der Literatur diskutierte Kriterien fokussieren nicht die eigentliche Güte einer heuristischen Lösung. So nennen Pfohl und Hebel (vgl. [PfHe82]) außerdem Implementierungskosten (Kosten für die Schulung und Involvierung der Anwender), Programmierzeit (technische Realisierung einer Heuristik), maschinenbezogene Testzeit (zum Austesten der Heuristik benötigte Rechenzeit) sowie die Speicherplatzkomplexität (der temporär notwendige Haupt- und Festplattenspeicher). Die Autoren Golden und Stewart (vgl. [GoSt85]) nennen zudem die konzeptionelle Einfachheit (den Grad der Nachvollziehbarkeit eines heuristischen Verfahrens) als Kriterium. Newell (vgl. [Newe69]) betrachtet des Weiteren die Allgemeinheit (Menge der unterschiedlichen Probleme, zu deren Lösung die Heuristik genutzt werden kann) und Mächtigkeit (Wahrscheinlichkeit des Findens einer Lösung) einer Methode. Schlussendlich stammt von Klein (vgl. [Klei71]) die heuristische Kraft (Anteil repräsentativer Entscheidungsaufgaben, die Methoden in einer definierten Zeit unabhängig von der Güte überhaupt lösen können). Es sei angemerkt, dass aufgrund der aktuellen Entwicklung der Computer einige dieser Kriterien häufig keinen Engpass mehr darstellen.

gestellten Anforderungen erfüllt. Die Lösungsqualität r einer Heuristik normiert Berens mit ($Z_H, Z > 0$):

$$r = \frac{\text{Zielfunktionswert, der mit der Heuristik gefundenen Lösung } Z_H}{\text{Zielfunktionswert der Optimallösung } Z} \quad (3.20)$$

Durch r kann die Lösungsqualität einer Heuristik eindeutig quantifiziert und somit mit anderen Heuristiken verglichen werden. Besteht die Zielsetzung in der Maximierung ergibt sich für r ein Wert zwischen 0 und 1. Im Falle eines Minimierungsproblems ist der Reziprokwert zu bilden.

Eine Berechnung von r ist bei Vorliegen einer optimalen Lösung ex post ohne weiteres möglich. In der Praxis interessiert jedoch die Lösungsqualität ex ante bzw. die Differenz eines ermittelten Ergebnisses im Vergleich zu der optimalen Lösung. Um nun die Lösungsqualität zu bestimmen, werden drei unterschiedliche Verfahren diskutiert (vgl. [PfHe82]):

- **Worst-Case-Analyse:** Im Rahmen dieser Analyse soll auf Basis mathematisch-analytischer Überlegungen die schlechteste Ausprägung eines Kriteriums (die ungünstigste Lösungsqualität oder die maximal notwendige Rechenzeit) bei Anwendung der entsprechenden Heuristik ermittelt werden (vgl. [Schm99]). Mittels dieser Untersuchung besteht eine Information über die garantierte Mindestqualität der Heuristik und eines insofern maximal auftretenden Schadens bei der Anwendung (vgl. [BDS04]). Die Ermittlung des Worst-Case ist jedoch mit einem erheblichen analytischen Aufwand verbunden, in dem ggf. sehr enge Prämissen gesetzt werden müssen. Zudem ist eine Ermittlung besonders bei komplexeren Heuristiken oftmals nicht möglich bzw. noch nicht gelungen (vgl. [Schm99]).
- **Probabilistische Analyse:** Das Ziel der probalistischen Analyse liegt in der Ermittlung der Wahrscheinlichkeit für das Auftreten bestimmter Ausprägungen der Kriterien (durchschnittlich zu erwartende Rechenzeit bei unterschiedlichen Problemgrößen bzw. eine Dichte- oder Verteilungsfunktion der Lösungsqualitäten) (vgl. [PfHe82], [Schm99]). Hierzu finden mathematisch-statistische Methoden Anwendung. Analog zu der Worst-Case-Analyse muss bei dieser Analyse kein aufwendiges Austesten der Heuristik erfolgen. Der Analyseaufwand ist jedoch noch größer als bei der Worst-Case-Analyse, da meistens Annahmen über Verteilungsfunktionen der verwendeten Ausgangsdaten getroffen werden müssen (vgl. [PfHe82]).
- **Simulativ-empirische Analyse:** Bei dieser Analyse werden die Ausprägungen der Kriterien durch ein Austesten der entsprechenden Heuristik auf Basis einer definierten Anzahl an Testproblemen einer Problemklasse ermittelt. Die Aussagekraft dieser Analysemethode wird determiniert von der präzisen Beschreibung der untersuchten Problemklasse, der Repräsentativität der untersuchten Testprobleme in einer solchen Problemklasse sowie der Dokumentation der Heuristik und Ergebnisse (vgl. [BDS04]). Zur Erreichung einer großen Repräsentativität sind zudem eine geeignete Anzahl an Testproblemen sowie die Gestalt entsprechender Testdaten wichtig. Aus diesem Grund sollten entweder Informationen realer Probleme verwendet oder mittels Datengeneratoren erzeugt werden (vgl. [Bere92]). Das Ziel dieser Analyse ist zumeist

die Ermittlung, wie weit das mit der Heuristik ermittelte Ergebnis von einer optimalen Lösung entfernt ist. Dargestellt werden kann dieses Ergebnis als Dichte- oder Verteilungsfunktion, die dann z.B. zum Vergleich mit anderen Heuristiken, die auf dieselbe Problemklasse anwendbar sind, herangezogen werden kann. Die Bestimmung der Lösungsqualität ist aufgrund der Komplexität jedoch nur für tendenziell einfach strukturierte Problemklassen möglich. So ist die optimale Lösung „*bei realistischen Problemgrößen i.a. nicht bekannt und sie kann auch nicht mit vertretbarem Aufwand ermittelt werden, denn sonst wäre der Einsatz von Heuristiken obsolet*“ [Bere92]. Außerdem sind in der Praxis oft Entscheidungsprobleme denkbar, die verschiedene Zielfunktionen bzw. Zielkriterien haben. In diesem Fall kann eine Gewichtung in relevante und nicht-relevante Ziele erfolgen und eine Bewertung entsprechend der „Relevanz“ der Zielkriterien vorgenommen werden. Bei diesem Vorgehen geht jedoch oft die geforderte Objektivität verloren. Angesichts der mangelnden Information über die Optimallösung werden von Berens (vgl. [Bere92]) zwei Vorgehensweisen vorgeschlagen, um zu Näherungswerten für die unbekannten Optimalwerte zu gelangen. Einerseits nennt er den Einsatz von Schranken als Ober- bzw. Untergrenzen, die vom maximalen bzw. minimalen Zielfunktionswert weder über- noch unterschritten werden können. Andererseits stellt er das Schätzen von Optimalwerten als Alternative vor. Eine weitere Lösung, mit der die fehlende Kenntnis einer Optimallösung umgangen wird, besteht darin, Probleme mit bereits bekannten optimalen Lösungen zu erzeugen und diese als Testprobleme zu verwenden.

Stabilität

In der Literatur werden unterschiedliche Kriterien zur Messung der Stabilität vorgeschlagen. Cowling et. al. (vgl. [COP04]) definieren die Stabilität als die Summe der absoluten Abweichung des Fertigstellungszeitpunktes des Ausgangsplans und des geänderten Belegungsplans. Subramaniam et. al. (vgl. [SRR05]) ermittelten für ein Job Shop Problem eine normalisierte Abweichung der Startzeiten des alten und neuen Plans. Rangaswamy et. al. (vgl. [RFK04]) berechnen die Stabilität mit der absoluten Abweichung und bewerten zusätzlich die durch eine reaktive Planung früher beginnenden Aufträge mit Strafkosten. Curry und Peters (vgl. [CuPe05]) ermitteln die Nervosität eines Plans anhand des Anteils veränderter Aufträge, die einer alternativen Produktionslinie zugeordnet wurden. Einen umfangreichen Katalog an Kriterien zur Messung der Stabilität wurde zudem von Neuhaus (vgl. [Neuh08]) definiert. In diesem differenziert er lokale und globale Kennzahlen. Die lokalen Kriterien betrachten Aufträge, die im Rahmen eines Umplanungslaufs geändert wurden. Hierzu zählen:

- Zeitliche Lage: Diese Kriterien beziehen sich auf die durchschnittliche zeitliche Verschiebung aller umplanbaren Aufträge sowie den Anteil der Aufträge mit dem im Vergleich zum Ausgangsplan gleichen Fertigstellungszeitpunkt. Weiter spezifiziert werden kann das letztere Kriterium durch Angabe eines erlaubten Toleranzbereiches und der Eingrenzung, ob vor- oder zurückverlegte Aufträge ausgeschlossen werden sollen. Eine weitere Kennzahl definiert, für welchen Anteil der Aufträge aus dem Ausgangsplan ein gleichartiger Auftrag (zur Fertigung werden z.B. die gleichen Werkzeuge genutzt) mit identischem Fertigstellungstermin eingeplant wurde.
- Anlagenzuordnung: Dieses Kriterium beschreibt den Anteil der Aufträge, deren Maschinenzuordnung unverändert geblieben ist.

- Auftragsreihenfolge: Der Anteil von Auftragspaaren mit unveränderter Sequenz wird mit diesem Kriterium definiert.

Die globalen Kennzahlen beziehen sich hingegen auf alle Aufträge, die im Laufe einer Folge von Umplanungsläufen betroffen waren. Dadurch kann berücksichtigt werden, ob ein einzelner Auftrag mehrmals verschoben wurde.

- Zeitliche Lage: Dieses Kriterium betrachtet die durchschnittliche zeitliche Verschiebung aller Aufträge sowie die Umplanungshäufigkeit im Bezug auf die Fertigstellungszeitpunkte in einer Sequenz von Planungsläufen.
- Anlagenzuordnung: Dieses Kriterium beschreibt die durchschnittliche Anzahl, mit der Aufträge einer alternativen Maschine zugeordnet wurden.

Zur Durchführung eines Benchmarking können verschiedene Modelle Anwendung finden. Im Wesentlichen beinhalten diese jedoch alle die gleichen Informationen, jeweils nur in einem unterschiedlichen Abstraktionsgrad. Grundsätzlich besteht ein Benchmarking aus der Identifikation des zu analysierenden Bereichs (z.B. der Produktionsplanung eines Unternehmens), der Benchmarkingpartner (z.B. der zu vergleichenden Methoden) sowie der Informationsbeschaffung (Art der relevanten Daten sowie deren Sammlung bzw. Generierung). Zusätzlich sind Methoden zur Erzeugung und Analyse der Benchmarkingergebnisse zu bestimmen. In Abhängigkeit der Art des Benchmarking wird zudem noch eine Phase zur Umsetzung der Resultate genannt (vgl. [KaÖs94]).

4 Zu leistende Arbeit

„Gib mir nicht, was ich mir wünsche,
sondern was ich brauche.“

(Antoine de Saint-Exupéry)

Im Stand der Technik in Kapitel 3 wurden verschiedene Verfahren aufgezeigt, die zumindest eine Lösung von Teilproblemen generell ermöglichen. Keines dieser Verfahren erlaubt jedoch die ganzheitliche Betrachtung der in Kapitel 2.2 genannten Anforderungen. In diesem Kapitel wird beschrieben, wie die definierte Problemstellung unter Berücksichtigung des Stands der Technik in Kapitel 5 umgesetzt werden soll. Hierzu werden nachfolgend die einzelnen Teilprobleme dargestellt.

4.1 Entwicklung eines mathematischen Modells

Zur Lösung des vorliegenden Problems soll ein mathematisches Modell konzipiert werden. Ziel dieses Modells ist nicht die Lösung von Problemen praxisrelevanter Größen, sondern die Generierung von Vergleichsdaten zur Bewertung der Güte des heuristischen Verfahrens. Hierzu ist ein Grundmodell zu erstellen, das den beschriebenen Anforderungen genügt. Als Basis soll das in Kapitel 3.1.2 vorgestellte Modell verwendet werden. Die Ergebnisse dieses Grundmodells erlauben jedoch nur die Lösung von statischen Daten. Der geforderten rollierenden Planung wird damit nicht Rechnung getragen. Hierzu ist das Grundmodell entsprechend zu erweitern, so dass die Ergebnisse früherer Planungsläufe berücksichtigt werden können. Außerdem sollen analog zu dem heuristischen Verfahren Möglichkeiten zur Beeinflussung der Planungsergebnisse integriert werden.

4.2 Entwicklung eines heuristischen Verfahrens

Als Alternative zu der optimalen Lösung soll ein Verfahren erstellt werden, welches die Planung über alle Produktionsstufen hinweg vornimmt. Neben der initialen Generierung möglichst effizienter Produktionspläne sind auch Mechanismen zur zeitnahen Reaktion auf Störungen (wie z.B. das Eintreffen von Eilaufträgen oder die Stornierung bzw. Verschiebung von Aufträgen) erforderlich. Mit zunehmender Länge des Planungshorizontes werden die Daten und die darauf basierenden Entscheidungen immer unsicherer und darüber hinaus nimmt der Planungsaufwand untragbare Ausmaße an. Aufgrund dessen ist der Planungshorizont zu begrenzen sowie in unterschiedliche Segmente zu unterteilen. Diese sind mit unterschiedlichen Verfahren zu beplanen. Für die Planung des Feinplanungshorizontes sollen Eröffnungs- und Verbesserungsverfahren entwickelt werden. Mit Hilfe des Eröffnungsverfahrens ist bei jedem Planungslauf eine Lösung x für die Zielsetzung: $\min \gamma = \text{Produktionskosten}(x) + \text{Lagerkosten}(x) + \text{Verzugskosten}(x)$ zu suchen. Eine solche Lösung wird als gültig definiert, wenn sie jedem Auftrag A_i eine gültige Startzeit $\text{Start}[A_i]$ und Endzeit $\text{Ende}[A_i]$ zuordnet. Des Weiteren ist jeder Periode P des Planungshorizontes ein Schichtmodell $SM[P]$

zuzuweisen, um die benötigte Kapazität in diesem Zeitabschnitt bereit zu stellen. Im Gegensatz zu den meisten Verfahren im Stand der Technik sind die Zeiten für die Bearbeitung und das Rüsten von den Leistungsgraden abhängig und daher während des Planungslaufes zu ermitteln. In der Literatur wurden Prioritätsregeln für die Planung dynamischer Planungsumgebungen als gut geeignet identifiziert, weshalb sie auch innerhalb des Eröffnungsverfahrens Anwendung finden sollen. Nachdem ein gültiger Plan erstellt wurde, ist mittels eines Verbesserungsverfahrens der Lösungsraum L hinsichtlich einer Lösung zu durchsuchen, für die gilt: $x \in L, \forall x' \in L/\{x\}: \gamma(x) \leq \gamma(x')$. Um das Verharren in lokalen Optima zu vermeiden, sollen Mechanismen auf Basis von Simulated Annealing und Tabu Search genutzt werden. In der Literaturdiskussion wurde gezeigt, dass die Bildung der Nachbarschaft mit Hilfe von insert-Zügen zu besseren Ergebnissen führt als die Verwendung von swap-Zügen. Diesen Erkenntnissen wird auch in dieser Arbeit gefolgt. Die Performance von Simulated Annealing und Tabu Search kann entscheidend durch die jeweilige Parameterkonfiguration beeinflusst werden. In der Literatur erfolgten bereits unterschiedliche Analysen zu den Auswirkungen und es wurden generell geeignete Parametereinstellungen identifiziert. Diese Werte werden auch in dieser Arbeit als Standardparameter unterstellt, welche jedoch durch den Planer verändert werden können sollen. Weitere Möglichkeiten zur Parametrisierung sollen sich bspw. auf die Zielgewichtung oder die Flexibilität der Planung beziehen. Damit soll es möglich sein, das Verfahren an die jeweiligen Anforderungen anzupassen und unterschiedliche Pläne zu konstruieren. Die Planung des Grobplanungshorizontes soll im Gegensatz zu den Verfahren der Feinplanung auf Basis von Makroperioden erfolgen. Hierfür dient das in Kapitel 3.1.1 vorgestellte MLCLSP als Grundlage. Zudem ist ein Verfahren zu entwickeln, welches bei geplanten Verzugsmengen im Grobplanungshorizont und Kapazitätsreserven im Feinplanungshorizont passende Bedarfsumfänge in den Feinplanungshorizont verschiebt.

4.3 Entwicklung eines Benchmarkingverfahrens

Es ist eine Benchmarkingkomponente zu entwickeln, die es ermöglicht, Pläne von verschiedenen Verfahren zu vergleichen. Basis dieser Komponente soll die in Kapitel 3.3 vorgestellte simulativ-empirische Analyse sein. Die Bewertung dieser Verfahren soll anhand unterschiedlicher Kriterien möglich sein und mittels einer übersichtlichen und aussagekräftigen Ergebnisvisualisierung erfolgen können. Hierzu sind geeignete Benchmarkingkriterien und Untersuchungsobjekte zu definieren. Zur Generierung von Bedarfen ist ein Datengenerator zu entwickeln, der über den rollierenden Planungshorizont und dem Produktionssystem konsistente Daten erzeugt. Ein Schwerpunkt der Entwicklung wird auf einer geeigneten Ablaufsteuerung liegen (siehe Abbildung 19). Diese ist für den aktuellen, durch die rollierende Planung fortgeschriebenen, Planungshorizont erforderlich, um den Input der jeweiligen Verfahren (V_x) zu generieren, die Planungsverfahren zu starten und zu steuern sowie den Output der Verfahren einzulesen und zu interpretieren. Dabei muss auch die Möglichkeit gegeben sein, einzelne Verfahren zu einem Durchlauf zu kombinieren und deren Gesamtoutput zu analysieren.

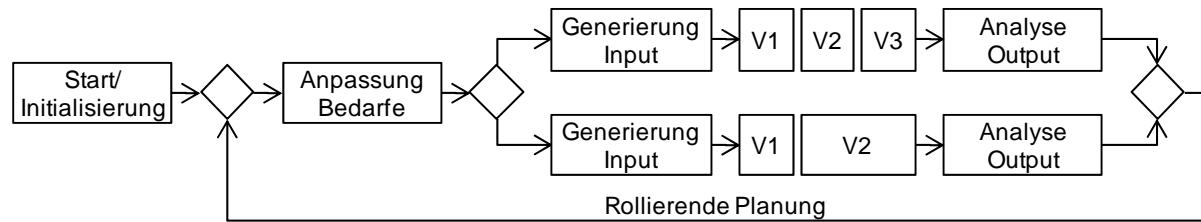


Abbildung 19: Ablaufsteuerung der Benchmarkingplattform

5 Konzeption

„Inmitten der Schwierigkeit liegt die Möglichkeit.“

(Albert Einstein)

In den vorherigen Kapiteln wurden das Problem und der aktuelle Stand der Technik vorgestellt. Wie im Stand der Technik deutlich wurde, gibt es bereits einige Verfahren zur Planung des Flexible Flow Shop. Allerdings wurde in der Literatur die Integration von Störungen bislang weitestgehend vernachlässigt (vgl. [GZA09]). Deren Berücksichtigung nehmen sich die im weiteren Verlauf dieser Arbeit vorgestellten Verfahren an.

Zu Beginn dieses Kapitels wird das zur optimalen Lösung notwendige mathematische Modell beschrieben. Neben der Vorstellung des ganzheitlichen Problems umfasst dies zudem eine Erweiterung zur Berücksichtigung vorheriger Planungsergebnisse. Auf Basis der gleichen Problemstellung wird im Kapitel 5.2 das heuristische Verfahren erklärt.

5.1 Entwicklung eines mathematischen Modells

Im Weiteren wird ausgehend von den Vorbetrachtungen ein mathematisches Modell mit einer kostenbasierten Zielfunktion und allen relevanten Restriktionen erstellt. Abschließend werden zwei Erweiterungen vorgestellt.

5.1.1 Grundmodell

Nachfolgend werden die benötigten Mengen, Parameter und Variablen definiert.

Definition der Mengen

$E = \{1..e\}$	Menge aller Erzeugnisse
$M = \{1.. T * MM\}$	Menge aller Mikroperioden
$N_e = \{1..n_e\}$	Menge der Nachfolgererzeugnisse des Erzeugnisses e
$PL = \{1..pl\}$	Menge aller Produktionslinien
$PU = \{1..pu\}$	Menge aller Puffer
$SM_{pl,t} = \{1..sm_{pl,t}\}$	Menge aller in Makroperiode t für die Produktionslinie pl verfügbaren Schichtmodelle
$T = \{1..t\}$	Menge der Makroperioden

Definition der Parameter

$a_{e,j} \in \mathbb{N}$	Direktbedarfskoeffizient, der angibt, wie viele Einheiten des Erzeugnisses e zur Herstellung einer Einheit des Erzeugnisses j benötigt werden
$b_{e,\mu} \in \mathbb{N}$	Bedarf nach Erzeugnis e in der Mikroperiode μ
$B \in \mathbb{N}$	Hinreichend große Zahl
$ep_{e,pu} \in \{0,1\}$	Lagerfähigkeit des Erzeugnisses e in Puffer pu . Der Parameter hat den Wert 1, wenn das Erzeugnis e in Puffer pu gelagert werden kann, sonst 0
$I_e^{init} \in \mathbb{N}$	Initialer Lagerbestand von Erzeugnis e
$kf_e \in \mathbb{R}^0$	Kapazitätsfaktor für die Lagerung einer Einheit von Erzeugnis e
$lg_{sm,pl,\mu} \in [0,1]$	Leistungsgrad auf der Produktionslinie pl im Schichtmodell sm für Mikroperiode μ
$lk_e \in \mathbb{R}^0$	Lagerkostenfaktor für eine Einheit des Erzeugnisses e
$MM \in \mathbb{N}$	Anzahl der Mikroperioden je Makroperiode
$mp \in \mathbb{N}$	Maximaler Abschnitt durchgehender Produktion, ausgehend von gegebenen Bedarfen
$mr_{pl} \in \mathbb{N}$	Maximale Rüstzeiten zwischen zwei Erzeugnissen auf der Produktionslinie pl
$pf_{pu}^{max} \in \mathbb{N}$	Maximale Pufferkapazität des Puffers pu
$prf_{e,pl} \in \{0,1\}$	Produktionsfähigkeit des Erzeugnisses e auf Produktionslinie pl . Der Parameter hat den Wert 1, falls das Erzeugnis e auf Produktionslinie pl produziert werden kann, sonst 0
$rz_{i,j,pl} \in \mathbb{N}$	Rüstzeit von Erzeugnis i auf Erzeugnis j auf Produktionslinie pl
$sk_{sm,pl,t} \in \mathbb{R}^0$	Kosten für Schichtmodell sm für die Makroperiode t und die Produktionslinie pl
$tb_{e,pl} \in \mathbb{N}$	Stückbearbeitungszeit für Erzeugnis e auf Produktionslinie pl
$vk_e \in \mathbb{R}^0$	Verzugskostenfaktor für eine Einheit des Erzeugnisses e
$vlz_e \in \mathbb{N}$	Vorlaufzeit von Erzeugnis e

Definition der Variablen

$h_{e,pl,\mu}^c \in \{0,1\}$	Hilfsvariable zur Linearisierung (siehe Restriktionen des virtuellen Lagers)
$h_{e,pl,\mu}^d \in \mathbb{N}$	Hilfsvariable für den Übergang des virtuellen Lagers in den realen Lagerbestand nach Beendigung eines Loses von Erzeugnis e in Mikroperiode μ der Produktionslinie pl

$h_{e,pl,\mu_1,\mu_2}^{dp} \in \mathbb{N}$	Hilfsvariable, die die produzierte Menge von Erzeugnis e , bei einer durchgehenden Produktion von Mikroperiode μ_1 bis μ_2 auf Produktionslinie pl annimmt (siehe Lager- und Verzugsmengenbilanzgleichung)
$h_{e,pl,\mu}^{I,vir,ab} \in \mathbb{N}$	Hilfsvariable zur Rundung (siehe Restriktionen des virtuellen Lagers)
$h_{e,pl,\mu_1,\mu_2}^p \in \{0,1\}$	Hilfsvariable, die angibt, ob das Erzeugnis e von Mikroperiode μ_1 bis μ_2 durchgehend auf Produktionslinie pl produziert wird (siehe Lager- und Verzugsmengenbilanzgleichung)
$h_{e,pl,\mu}^s \in \mathbb{N}$	Hilfsvariable, die den Sekundärbedarf von Erzeugnis e in Mikroperiode μ auf Produktionslinie pl angibt (siehe Lager- und Verzugsmengenbilanzgleichung)
$h_{e,pl,\mu}^{sb} \in \mathbb{N}$	Hilfsvariable zur Rundung (siehe Lager- und Verzugsmengenbilanzgleichung)
$h_{e,pl,\mu}^v \in \{0,1\}$	Hilfsvariable zur Linearisierung (siehe Restriktionen des virtuellen Lagers)
$h_{i,j,pl,\mu_1,\mu_2}^x \in \{0,1\}$	Hilfsvariable, die angibt, ob ein Rüstvorgang auf Produktionslinie pl von Erzeugnis i nach Erzeugnis j innerhalb den Mikroperioden μ_1 und μ_2 stattfindet (siehe Restriktionen zur Abbildung des Rüstens)
$I_{e,\mu} \in \mathbb{R}^0$	Lagerbestand eines Erzeugnisses e in Mikroperiode μ
$I_{e,pl,\mu}^{vir} \in \mathbb{R}^0$	virtueller Lagerbestand eines Erzeugnisses e in Mikroperiode μ auf der Produktionslinie pl
$RS_{e,\mu} \in \mathbb{R}^0$	Verzugsmenge eines Erzeugnisses e in Mikroperiode μ
$sm_{sm,pl,t} \in \{0,1\}$	Binärvariable, die angibt, ob das Schichtmodell sm in Makroperiode t für die Produktionslinie pl gewählt ist ($sm_{sm,pl,t} = 1$) oder nicht ($sm_{sm,pl,t} = 0$)
$x_{e,pl,\mu} \in \{0,1\}$	Binärvariable, die angibt, ob das Erzeugnis e auf Produktionslinie pl in Mikroperiode μ produziert wird ($x_{e,pl,\mu} = 1$) oder nicht ($x_{e,pl,\mu} = 0$)
$Z_{e,pl,\mu} \in \mathbb{N}$	Kapazität für Erzeugnis e in Mikroperiode μ auf Produktionslinie pl

5.1.1.1 Zielfunktion

Das Ziel dieses mathematischen Modells ist die Minimierung der Gesamtkosten (siehe (5.1)). Die hierbei betrachteten Kostenarten sind Verzugskosten, Produktionskosten und Lagerkosten.

$$\min z = \text{Verzugskosten} + \text{Produktionskosten} + \text{Lagerkosten} \quad (5.1)$$

Verzugskosten sind die Kosten, die veranschlagt werden, falls Bedarfe nicht fristgerecht befriedigt werden. Um diese Kosten zu bestimmen, werden zu jeder Mikroperiode μ und jedem Erzeugnis e etwaige Verzüge ermittelt und mit dem entsprechenden Verzugskostensatz vk_e multipliziert. Die gesamten Verzugskosten ergeben sich als Summe über alle Mikroperioden und Erzeugnisse.

$$Verzugskosten = \sum_{e \in E} \sum_{\mu \in M} RS_{e,\mu} * vk_e \quad (5.2)$$

Produktionskosten sind die Kosten, die sich als Summe der Kosten für die in den einzelnen Makroperioden und Produktionslinien gewählten Schichtmodellen ergeben.

$$Produktionskosten = \sum_{sm \in SM_{pl,t}} \sum_{pl \in PL} \sum_{t \in T} sm_{sm,pl,t} \cdot sk_{sm,pl,t} \quad (5.3)$$

Lagerkosten fallen an, wenn Erzeugnisse früher als benötigt produziert und eingelagert werden müssen. Hierzu werden zu jeder Mikroperiode μ für jedes Erzeugnis e die etwaigen Lagerbestände ermittelt und mit dem Lagerhaltungskostensatz lk_e multipliziert. Die gesamten Lagerkosten ergeben sich als Summe über alle Mikroperioden und Erzeugnisse.

$$Lagerkosten = \sum_{e \in E} \sum_{\mu \in M} I_{e,\mu} * lk_e \quad (5.4)$$

5.1.1.2 Restriktionen

Die Lager- und Verzugsmengenbilanzgleichung stellt sicher, dass der Bedarf gedeckt wird. Aufgrund der mit einer Flexible Flow Shop verbundenen geschlossenen Produktion (mit der Bearbeitung eines Loses darf erst begonnen werden, wenn sämtliche Vorerzeugnisse fertig gestellt wurden), ist diese Restriktion sehr komplex. Zum einfacheren Verständnis wird zuerst die nicht-linearisierte Form eingeführt.⁵⁸

⁵⁸ Durch eine Verkleinerung des Lösungsraumes vor der Initialisierung der Variablen und Restriktionen kann die Lösungszeit des Modells verringert werden. Dies kann erreicht werden, indem nur Variablen und Restriktionen initialisiert werden, die relevant und sinnvoll sind. Hierzu bietet sich der Parameter $prf_{e,pl}$ an, der den Wert 1 annimmt, wenn Erzeugnis e auf Produktionslinie pl produzierbar ist, ansonsten wird der Wert 0 angenommen. Wird die Bedingung, dass eine Variable oder Restriktion nur in das Modell aufgenommen wird, wenn $prf_{e,pl} = 1$ ist, an den geeigneten Stellen eingefügt, kann also eine beträchtliche Laufzeitsparnis erzielt werden.

$$\begin{aligned}
 I_{e,\mu-1} - RS_{e,\mu-1} + \sum_{pl \in PL} h_{e,pl,\mu}^d & \quad \forall e \in E \quad (5.5) \\
 - \sum_{pl \in PL} \sum_{j \in N_e} \left(a_{e,j} * (1 - x_{j,pl,\mu+vlz_j-1}) \right. \\
 \left. * \left| \sum_{\mu_1=\mu+vlz_j}^{|M|} \left(\frac{Z_{j,pl,\mu_1}}{tb_{j,pl}} * h_{j,pl,\mu+vlz_j,\mu_1}^p \right) \right| \right) - I_{e,\mu} + RS_{e,\mu} = b_{e,\mu}
 \end{aligned}
 \quad \forall \mu \in M$$

Die gesamte Nachfrage eines Erzeugnisses e setzt sich aus den Primärbedarfen und den durch die direkten Nachfolger verursachten Sekundärbedarfe zusammen. Die Berechnung dieses Sekundärbedarfs erfolgt mit Hilfe des Direktbedarfskoeffizienten $a_{e,j}$, der mit der Primärbedarfsmenge des übergeordneten Erzeugnisses j multipliziert wird. Um zudem die rechtzeitige Verfügbarkeit sicherzustellen, wird die Vorlaufzeit vlz_j berücksichtigt. Die in einer Periode zu produzierende Menge ergibt sich aus dem Lagerbestand der Vorperiode sowie dem angestrebten Lagerbestand der entsprechenden Periode. Falls das Kapazitätsangebot zur Deckung dieser Bedarfsumfänge nicht ausreicht, wird der Bedarf einer Mikroperiode μ reduziert und in die Nachfolgerperiode verschoben. Realisiert wird dies durch die Verzugsmengenvariable $RS_{e,\mu}$. Die in einer Mikroperiode auf einer Produktionslinie produzierbaren Umfänge eines Erzeugnisses ergeben sich aus der Division der gewählten Kapazität $Z_{j,pl,\mu}$ und der erzeugnis- und produktionslinienabhängigen Stückbearbeitungszeit $tb_{j,pl}$. Ein Los wird dem Lagerbestand nur als Ganzes zugeführt. Um dies abzubilden, wird ein virtuelles Lager eingeführt. Dessen Bestand ($h_{e,pl,\mu}^d$) wird erst zum realen Lagerbestand addiert, wenn das entsprechende Los endet. In der Restriktion (5.5) wird jedoch zur Bestimmung des Sekundärbedarfs eine Multiplikation von Entscheidungsvariablen sowie ein Abrunden vorgenommen und ist damit nicht linear. Damit ist es einem Standardsolver (z.B. CPLEX) nicht möglich, das Problem zu lösen, so dass es in eine linearisierte Formulierung überführt werden muss. Zur Linearisierung der Lager- und Verzugsmengenbilanzgleichung dienen die nachfolgenden Restriktionen ((5.6)-(5.16)).

Die Definition des Flexible Flow Shop (siehe Kapitel 2.2.1) verlangt, dass alle Vorerzeugnisse vor dem Start des jeweiligen Loses zur Verfügung stehen. Da der Bedarf eines Erzeugnisses jedoch gesplittet und damit zu unterschiedlichen Zeitpunkten und auf unterschiedlichen Produktionslinien gefertigt werden kann, muss die Höhe des jeweiligen Sekundärbedarfs ermittelt werden. Zur Bestimmung dieses Sekundärbedarfs wird ein binärer Indikator h_{e,pl,μ_1,μ_2}^p definiert, der angibt, ob ein Erzeugnis e auf einer Produktionslinie pl im Zeitraum μ_1 bis μ_2 ohne Unterbrechung produziert wurde. Zur Bestimmung dieses Indikators dienen die Restriktionen (5.6) und (5.7).⁵⁹

⁵⁹ Zur Reduktion der Lösungszeit kann die Anzahl der Restriktionen reduziert werden. Dies kann erreicht werden, indem die theoretische maximal durchgehende Produktionszeit (mp) berechnet wird und damit ausschließlich Mikroperioden mit höchstens diesem Abstand miteinander verglichen werden. Die Produktionszeit ist vor dem Aufstellen der Bedingungen zu ermitteln. Die Ermittlung erfolgt mit: $\max(Gesamtbedarf_i / (\max(Taktzeit_{i,pl}) * \min(Leistungsgeschwindigkeit_i))) \forall i \in E, pl \in PL$. Entsprechend sind die Restriktionen zu erweitern mit: $\mu_2 - \mu_1 \leq mp$.

$$1 - h_{e,pl,\mu_1,\mu_2}^p \leq \sum_{\substack{\mu_3 = \mu_1}}^{\mu_2} (1 - x_{e,pl,\mu_3}) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \leq \mu_2 \end{array} \quad (5.6)$$

$$B * (1 - h_{e,pl,\mu_1,\mu_2}^p) \geq \sum_{\substack{\mu_3 = \mu_1}}^{\mu_2} (1 - x_{e,pl,\mu_3}) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \leq \mu_2 \end{array} \quad (5.7)$$

Mit Hilfe dieses Indikators kann der entsprechende Produktionsumfang ermittelt werden (Restriktionen (5.8)-(5.10)⁵⁹). Zur Linearisierung des Ausdrucks $\frac{Z_{j,pl,\mu_1}}{tb_{j,pl}} * h_{j,pl,\mu_1,\mu_2}^p$ wird die Hilfsvariable $h_{e,pl,\mu_1,\mu_2}^{dp}$ eingefügt, die den Wert $\frac{Z_{j,pl,\mu_1}}{tb_{j,pl}}$ annimmt, wenn $h_{j,pl,\mu_1,\mu_2}^p = 1$ ist, sonst ist sie 0.

$$h_{e,pl,\mu_1,\mu_2}^{dp} \leq B * h_{j,pl,\mu_1,\mu_2}^p \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \leq \mu_2 \end{array} \quad (5.8)$$

$$h_{e,pl,\mu_1,\mu_2}^{dp} - \frac{Z_{e,pl,\mu_2}}{tb_{e,pl}} \leq B * (1 - h_{j,pl,\mu_1,\mu_2}^p) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \leq \mu_2 \end{array} \quad (5.9)$$

$$\frac{Z_{e,pl,\mu_2}}{tb_{e,pl}} - h_{e,pl,\mu_1,\mu_2}^{dp} \leq B * (1 - h_{j,pl,\mu_1,\mu_2}^p) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \leq \mu_2 \end{array} \quad (5.10)$$

In der Lager- und Verzugsmengenbilanzgleichung soll nur der Sekundärbedarf berücksichtigt werden, der auch weiterhin verarbeitet werden kann, d.h., nur ganze Erzeugnisse. Da viele Standardsolver (z.B. CPLEX) nicht über die Funktionen Floor und Ceiling verfügen, wird für den Ausdruck $\lfloor \sum_{\mu_2 \geq \mu} h_{e,pl,\mu_1,\mu_2}^{dp} \rfloor$ die Hilfsvariable $h_{e,pl,\mu}^{sb}$ definiert und die Restriktionen (5.11) und (5.12) formuliert⁶⁰.

⁶⁰ Analog zu den vorherigen Erläuterungen (siehe Fußnote 59) können auch an dieser Stelle Restriktionen ausgeschlossen werden. Hierzu kann für die Summe als Obergrenze $\mu_2 - \mu \leq mp$ gesetzt werden.

$$h_{e,pl,\mu}^{sb} \leq \sum_{\mu_2=\mu}^{|M|} h_{e,pl,\mu,\mu_2}^{dp} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.11)$$

$$h_{e,pl,\mu}^{sb} + 0.99 \geq \sum_{\mu_2=\mu}^{|M|} h_{e,pl,\mu,\mu_2}^{dp} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.12)$$

Schließlich wird zur Linearisierung von $(1 - x_{e,pl,\mu-1}) * h_{e,pl,\mu}^{sb}$ die Hilfsvariable $h_{e,pl,\mu}^s$ benutzt, um den anfallenden Sekundärbedarf einmalig zu erfassen (Restriktionen (5.13)-(5.16)). Wurde in der Vorperiode bereits produziert ($1 - x_{e,pl,\mu-1} = 0$) nimmt $h_{e,pl,\mu}^{sb}$ den Wert 0 an, so dass die Bedarfsmengen von Vorerzeugnissen einer bereits gestarteten Produktion nicht noch einmal einkalkuliert werden.

$$h_{e,pl,\mu}^s \leq B * (1 - x_{e,pl,\mu-1}) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.13)$$

$$h_{e,pl,\mu}^s - h_{e,pl,\mu}^{sb} \leq B * (1 - x_{e,pl,\mu-1}) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.14)$$

$$h_{e,pl,\mu}^{sb} - h_{e,pl,\mu}^s \leq B * (1 - x_{e,pl,\mu-1}) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.15)$$

$$h_{e,pl,\mu}^s = h_{e,pl,\mu}^{sb} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \mu = 1 \end{array} \quad (5.16)$$

Zusammenfassend ergibt sich die folgende Lager- und Verzugsmengenbilanzgleichung ((5.17)).

$$I_{e,\mu-1} - RS_{e,\mu-1} + \sum_{pl \in PL} h_{e,pl,\mu}^d - \sum_{pl \in PL} \sum_{j \in N_e} (a_{e,j} * h_{j,pl,\mu+vlz_j}^s) - I_{e,\mu} + RS_{e,\mu} = b_{e,\mu} \quad \begin{array}{l} \forall e \in E \\ \forall \mu \in M \end{array} \quad (5.17)$$

In ein virtuelles Lager werden so lange Erzeugnisse e eines Loses gelagert, bis alle Erzeugnisse des entsprechenden Loses produziert wurden. Erst dann wird der gesamte Umfang dem realen Lagerbestand zugeführt. Zur Abbildung von diesem virtuellen Lagerbestand dient die Variable $I_{e,pl,\mu}^{vir}$. Wird in einer Mikroperiode μ ein Erzeugnis e auf einer Produktionslinie pl

produziert ($x_{e,pl,\mu} = 1$), wird die jeweilige Produktionsmenge mit $I_{e,pl,\mu}^{vir} = I_{e,pl,\mu}^{vir} + \frac{Z_{e,pl,\mu}}{tb_{e,pl}}$ zum virtuellen Lagerbestand addiert. Dies wird durch die Restriktion (5.18) bis (5.20) ausgedrückt.

$$I_{e,pl,\mu}^{vir} \leq B * x_{e,pl,\mu} \quad \forall e \in E \quad (5.18)$$

$$\forall pl \in PL$$

$$\forall \mu \in M$$

$$I_{e,pl,\mu}^{vir} - \left(I_{e,pl,\mu-1}^{vir} + \frac{Z_{e,pl,\mu}}{tb_{e,pl}} \right) \leq B * (1 - x_{e,pl,\mu}) \quad \forall e \in E \quad (5.19)$$

$$\forall pl \in PL$$

$$\forall \mu: 2 \leq \mu \leq |M|$$

$$\left(I_{e,pl,\mu-1}^{vir} + \frac{Z_{e,pl,\mu}}{tb_{e,pl}} \right) - I_{e,pl,\mu}^{vir} \leq B * (1 - x_{e,pl,\mu}) \quad \forall e \in E \quad (5.20)$$

$$\forall pl \in PL$$

$$\forall \mu: 2 \leq \mu \leq |M|$$

Der aus einem virtuellen Lager in den realen Lagerbestand zu überführende Umfang kann mit $h_{e,pl,\mu}^d = \lfloor I_{e,pl,\mu-1}^{vir} \rfloor * \max(x_{e,pl,\mu-1} - x_{e,pl,\mu}, 0)$ dargestellt werden. Demnach ist das Ende eines Loses genau dann erreicht, wenn $x_{e,pl,\mu-1} = 1$ und $x_{e,pl,\mu} = 0$. In diesem Fall wird $h_{e,pl,\mu}^d$ der Umfang des virtuellen Lagers $\lfloor I_{e,pl,\mu-1}^{vir} \rfloor$ zugewiesen, so dass er zur Bedarfsdeckung verwandt werden kann. Aufgrund der Alles-oder-Nichts-Produktion-Annahme wird die Kapazität einer Periode immer vollständig zur Produktion des Erzeugnisses genutzt. Im optimalen Fall stellt die Kapazität immer ein ganzzahliges Vielfaches der Bearbeitungszeit dar, so dass immer nur vollständige Erzeugnisse produziert werden. Andernfalls kann es vorkommen, dass die Anzahl der produzierten Erzeugnisse nicht ganzzahlig ist. Ist z.B. ein Bucket 60 Minuten groß und besteht für ein Erzeugnis mit einer Bearbeitungszeit von 7 Minuten ein Bedarf von 17 Einheiten, würden innerhalb der 2 zu belegenden Perioden 17,14 Erzeugnisse produziert und im virtuellen Lager gesammelt. Da der virtuelle Lagerbedarf nicht ganzzahlig sein kann, aber nur ganze Erzeugnisse aus der Produktion überführt werden sollen, ist eine Rundung notwendig. Der obige Ausdruck muss erneut linearisiert werden. Mittels der Restriktionen (5.21) bis (5.24) wird $\max(x_{e,pl,\mu-1} - x_{e,pl,\mu}, 0)$ behandelt. Hierzu werden die Hilfsvariablen $h_{e,pl,\mu}^v = \max(x_{e,pl,\mu-1} - x_{e,pl,\mu}, 0)$ und $h_{e,pl,\mu}^c$ eingeführt. Demnach ist ein Losende erreicht, wenn $h_{e,pl,\mu}^v = 1$, ansonsten ist $h_{e,pl,\mu}^v = 0$. Zur Linearisierung von $\lfloor I_{e,pl,\mu}^{vir} \rfloor$ werden die Hilfsvariable $h_{e,pl,\mu}^{I,vir,ab}$ sowie die Restriktionen (5.25) und (5.26) definiert. Schlussendlich dienen zur Linearisierung von $h_{e,pl,\mu}^d = h_{e,pl,\mu}^{I,vir,ab} * h_{e,pl,\mu}^v$ die Restriktionen (5.27) bis (5.29). Wenn ein Los endet ($h_{e,pl,\mu}^v = 1$), ist $h_{e,pl,\mu}^d = h_{e,pl,\mu-1}^{I,vir,ab}$ sonst ($h_{e,pl,\mu}^v = 0$) ist $h_{e,pl,\mu}^d = 0$.

$$0 \leq h_{e,pl,\mu}^v \quad \forall e \in E \quad (5.21)$$

$$\forall pl \in PL$$

$$\forall \mu \in M$$

$$x_{e,pl,\mu-1} - x_{e,pl,\mu} \leq h_{e,pl,\mu}^v \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.22)$$

$$h_{e,pl,\mu}^v \leq (1 - h_{e,pl,\mu}^c) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.23)$$

$$h_{e,pl,\mu}^v - (x_{e,pl,\mu-1} - x_{e,pl,\mu}) \leq h_{e,pl,\mu}^c \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.24)$$

$$h_{e,pl,\mu}^{I,vir,ab} \leq I_{e,pl,\mu}^{vir} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.25)$$

$$h_{e,pl,\mu}^{I,vir,ab} + 0,99 \geq I_{e,pl,\mu}^{vir} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.26)$$

$$h_{e,pl,\mu}^d \leq B * h_{e,pl,\mu}^v \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.27)$$

$$h_{e,pl,\mu}^d - h_{e,pl,\mu-1}^{I,vir,ab} \leq B * (1 - h_{e,pl,\mu}^v) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.28)$$

$$h_{e,pl,\mu-1}^{I,vir,ab} - h_{e,pl,\mu}^d \leq B * (1 - h_{e,pl,\mu}^v) \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 2 \leq \mu \leq |M| \end{array} \quad (5.29)$$

Auf einer Produktionslinie pl kann in einer Makroperiode t nur genau ein Schichtmodell sm aus der Menge der in dieser Makroperiode und für diese Produktionslinie erlaubten Schichtmodelle aktiv sein. Dies wird durch die Restriktion (5.30) gewährleistet.

$$\sum_{sm \in SM_{pl,t}} sm_{sm,pl,t} = 1 \quad \begin{array}{l} \forall pl \in PL \\ \forall t \in T \end{array} \quad (5.30)$$

Zur Bildung einer Reihenfolge darf auf jeder Produktionslinie pl pro Mikroperiode μ höchstens ein Erzeugnis e produziert werden (siehe (5.31)). Dies entspricht den Annahmen des in Kapitel 3.1.2 beschriebenen MLDLSP.

$$\sum_{e \in E} x_{e,pl,\mu} \leq 1 \quad \begin{array}{l} \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.31)$$

Es muss sichergestellt sein, dass auf jeder Produktionslinie pl nur Erzeugnisse e eingeplant werden, die auf dieser Produktionslinie produziert werden können (siehe (5.32)):

$$x_{e,pl,\mu} \leq prf_{e,pl} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.32)$$

Die Pufferkapazitäten dürfen zu keinem Zeitpunkt überschritten werden. Der gesamte Pufferbestand des Puffers pu in der Mikroperiode μ ergibt sich als Summe der Bestände der einzelnen im Puffer gelagerten Erzeugnisse zu diesem Zeitpunkt mit dem entsprechenden Kapazitätsfaktor kf_e . Der Bestand für ein Erzeugnis e zum Zeitpunkt μ resultiert aus den geplanten Zugängen und den geplanten Abgängen bis zu diesem Zeitpunkt sowie dem initialen Lagerbestand (siehe (5.33)).

$$\sum_{e \in E} \left(\left(\sum_{\bar{\mu}=1}^{\mu} \sum_{pl \in PL} \frac{Z_{e,pl,\bar{\mu}}}{tb_{e,pl}} + I_e^{init} - \sum_{\bar{\mu}=1}^{\mu} \left(b_{e,\bar{\mu}} + \sum_{j \in N_e} a_{e,j} * b_{j,\bar{\mu}+v_{lz_j}} \right) \right) * kf_e * ep_{e,pu} \right) \leq pf_{pu}^{max} \quad \begin{array}{l} \forall pu \in PU \\ \forall \mu \in M \end{array} \quad (5.33)$$

Bei einem Wechsel von Erzeugnis i auf Erzeugnis j können produktionslinien- und reihenfolgeabhängige Rüstzeiten anfallen. D.h., in dieser Zeit darf auf der betroffenen Produktionslinie kein Erzeugnis zur Produktion eingeplant werden. Bei der Berechnung dieses Zeitabschnitts werden die Mikroperioden anteilig mit ihrem Leistungsgrad berücksichtigt. Der nicht-lineare Ausdruck (5.34) stellt die Einhaltung dieser Rüstzeiten sicher.

$$\sum_{\mu_1 < \mu < \mu_2} \sum_{\substack{sm \in SM \\ pl, \left[\frac{\mu}{MM} \right]}} lg_{sm,pl,\mu} * sm_{sm,pl,\left[\frac{\mu}{MM} \right]} \geq \sum_{i \in E} \sum_{j \in E} rz_{i,j,pl} * x_{i,pl,\mu_1} * x_{j,pl,\mu_2} \quad \begin{array}{l} \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \neq \mu_2 \end{array} \quad (5.34)$$

Zur Linearisierung dienen die Restriktionen (5.35)-(5.38). Die Hilfsvariable h_{i,j,pl,μ_1,μ_2}^x dient dabei zur Linearisierung des Produkts zweier Entscheidungsvariablen $x_{i,pl,\mu_1} * x_{j,pl,\mu_2}$. Die Restriktion (5.36) garantiert, dass h_{i,j,pl,μ_1,μ_2}^x gleich 0 ist, falls $x_{i,pl,\mu_1} = 0$ ist. Die zwei nächsten

Restriktionen ((5.37) und (5.38)) stellen sicher, dass falls x_{i,pl,μ_1} den Wert 1 annimmt, h_{i,j,pl,μ_1,μ_2}^x gleich x_{j,pl,μ_2} ist.⁶¹

$$\sum_{\mu_1 < \mu < \mu_2} \sum_{sm \in SM_{pl[\frac{\mu}{MM}]}} lg_{sm,pl,\mu} \cdot sm_{sm,pl[\frac{\mu}{MM}]} \geq \sum_{i \in E} \sum_{j \in E} rz_{i,j,pl} \cdot h_{i,j,pl,\mu_1,\mu_2}^x \quad \begin{array}{l} \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \neq \mu_2 \end{array} \quad (5.35)$$

$$h_{i,j,pl,\mu_1,\mu_2}^x \leq x_{i,pl,\mu_1} \quad \begin{array}{l} \forall i, j \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \neq \mu_2 \end{array} \quad (5.36)$$

$$h_{i,j,pl,\mu_1,\mu_2}^x - x_{j,pl,\mu_2} \leq 1 - x_{i,pl,\mu_1} \quad \begin{array}{l} \forall i, j \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \neq \mu_2 \end{array} \quad (5.37)$$

$$x_{j,pl,\mu_2} - h_{i,j,pl,\mu_1,\mu_2}^x \leq 1 - x_{i,pl,\mu_1} \quad \begin{array}{l} \forall i, j \in E \\ \forall pl \in PL \\ \forall \mu_1, \mu_2 \in M \\ \mu_1 \neq \mu_2 \end{array} \quad (5.38)$$

Die Variable $Z_{e,pl,\mu}$ drückt die für Erzeugnis e in der Mikroperiode μ auf Produktionslinie pl verfügbare Kapazität aus. Sofern das Erzeugnis e dieser Mikroperiode μ allokiert ist, ergibt sich die Kapazität in Abhängigkeit des gewählten Schichtmodells und dem entsprechenden Leistungsgrad. Diese Beziehung wird durch die folgende Gleichung (5.39) dargestellt:

$$Z_{e,pl,\mu} = x_{e,pl,\mu} * \sum_{sm \in SM_{pl[\frac{\mu}{MM}]}} lg_{sm,pl,\mu} * sm_{sm,pl[\frac{\mu}{MM}]} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{array} \quad (5.39)$$

Zur Linearisierung dieses Ausdrucks dienen die Restriktionen (5.40) bis (5.42). Restriktion (5.40) besagt, dass in der Mikroperiode μ auf der Produktionslinie pl nur dann Kapazität besteht, wenn das Erzeugnis e auch dieser Mikroperiode μ der Produktionslinie pl zugeordnet wurde. Falls die Binärvariable $x_{e,pl,\mu}$ 0 ist, ist $Z_{e,pl,\mu}$ auch 0. Sofern in der Mikroperiode μ das Erzeugnis produziert wird ($x_{e,pl,\mu} = 1$), ist $(1 - x_{e,pl,\mu})$ 0. Damit folgt zusammen mit den

⁶¹ Wie im Stand der Technik diskutiert wurde, steigt die Komplexität durch die Hinzunahme des reihenfolgeabhängigen Rüstens stark an. Eine Möglichkeit, um Laufzeit und Initialisierungsaufwand zu sparen, besteht darin, nur Restriktionen aufzustellen, die auch tatsächlich greifen können. Dazu werden die Restriktionen (5.35)-(5.38) erweitert um $\mu_2 - \mu_1 \leq mr_{pl}$.

Restriktionen (5.41) und (5.42) die Einhaltung der obigen Restriktion. Durch die Restriktion (5.43) wird zudem festgelegt, dass die Produktion eines Erzeugnisses nur stattfinden kann, wenn ein Minimum an Kapazität zur Verfügung steht (hier wird 0.01 als Mindestkapazität angenommen).

$$Z_{e,pl,\mu} \leq B * x_{e,pl,\mu} \quad \begin{aligned} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{aligned} \quad (5.40)$$

$$Z_{e,pl,\mu} - \sum_{\substack{sm \in SM \\ pl, [\frac{\mu}{MM}]}} \left(lg_{sm,pl,\mu} \cdot sm_{sm,pl, [\frac{\mu}{MM}]} \right) \leq B * (1 - x_{e,pl,\mu}) \quad \begin{aligned} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{aligned} \quad (5.41)$$

$$\sum_{\substack{sm \in SM \\ pl, [\frac{\mu}{MM}]}} \left(lg_{sm,pl,\mu} \cdot sm_{sm,pl, [\frac{\mu}{MM}]} \right) - Z_{e,pl,\mu} \leq B * (1 - x_{e,pl,\mu}) \quad \begin{aligned} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{aligned} \quad (5.42)$$

$$\sum_{e \in E} x_{e,pl,\mu} \leq 0.99 + \sum_{e \in E} Z_{e,pl,\mu} \quad \begin{aligned} \forall pl \in PL \\ \forall \mu \in M \end{aligned} \quad (5.43)$$

Zur korrekten Modellierung müssen noch zwei Sonderfälle berücksichtigt werden. Restriktion (5.44) verhindert die unvollständige Produktion eines einzelnen Erzeugnisses, so dass dieses durch Abrundung nicht in den Lagerbestand übergehen würde. Die Restriktion (5.45) stellt sicher, dass die Produktion zur Einsparung von Lagerkosten nicht künstlich in die Länge gezogen wird.

$$h_{e,pl,\mu}^d \geq h_{e,pl,\mu}^v \quad \begin{aligned} \forall e \in E \\ \forall pl \in PL \\ \forall \mu \in M \end{aligned} \quad (5.44)$$

$$B * (1 - h_{e,pl,\mu}^v) + h_{e,pl,\mu}^d - h_{e,pl,\mu-2}^{I,vir,ab} \geq h_{e,pl,\mu}^v \quad \begin{aligned} \forall e \in E \\ \forall pl \in PL \\ \forall \mu: 3 \leq \mu \leq |M| \end{aligned} \quad (5.45)$$

Verzugsmengen dürfen nur für Erzeugnisse anfallen, die einen Primärbedarfstermin nicht erfüllen konnten. Hierdurch wird gewährleistet, dass die Lager- und Verzugsmengenbilanzgleichung nicht derart manipuliert wird, dass durch Anpassung der Verzugsmengen angenommen wird, ein Vorerzeugnis stünde bereits zur Verfügung, obwohl es weder produziert, noch sich im Lagerbestand befindet. Indem die Obergrenze für eine Verzugsmenge mit der Verzugsmenge der vorherigen Periode zuzüglich dem Primärbedarf der betrachteten Periode definiert wird, ist nun gemeinsam mit der impliziten Untergrenze, die die Lager- und Ver-

zugsmengenbilanzgleichung vorgibt, ein eindeutiger Rahmen für die Verzugsmenge eingesetzt.

$$RS_{e,\mu} \leq RS_{e,\mu-1} + b_{e,\mu} \quad \forall e \in E \quad \forall \mu \in M \quad (5.46)$$

Der Lagerbestand und die Verzugsmengen zum Planungszeitpunkt $I_{e,0}$ bzw. $RS_{e,0}$ werden für jedes Erzeugnis e in den Restriktionen (5.47) und (5.48) auf einen Eingangswert I_e^{init} bzw. 0 gesetzt.

$$I_{e,0} = I_e^{init} \quad \forall e \in E \quad (5.47)$$

$$RS_{e,0} = 0 \quad \forall e \in E \quad (5.48)$$

Am Ende des Planungshorizontes müssen alle Verzugsmengen der Erzeugnisse ausgeglichen sein.

$$RS_{e,|M|} = 0 \quad \forall e \in E \quad (5.49)$$

Sofern für Erzeugnisse Vorlaufzeiten existieren, darf zu Beginn des Planungshorizontes entsprechend der Länge dieser Vorlaufzeiten nicht produziert werden (Restriktion (5.50)). Zudem kann in der ersten Mikroperiode kein Bestand vom virtuellen ins reale Lager überführt werden (Restriktion (5.51)) und die Produktion in der letzten Mikroperiode wird ebenfalls ausgeschlossen (Restriktion (5.52)), da die Fertigstellung innerhalb des Planungshorizontes nicht mehr möglich ist.

$$x_{e,pl,\mu} = 0 \quad \forall e \in E \quad \forall pl \in PL \quad \forall \mu: 1 \leq \mu < v_{lze} \quad (5.50)$$

$$h_{e,pl,\mu}^d = 0 \quad \forall e \in E \quad \forall pl \in PL \quad \mu = 1 \quad (5.51)$$

$$x_{e,pl,\mu} = 0 \quad \forall e \in E \quad \forall pl \in PL \quad \mu = |M| \quad (5.52)$$

5.1.2 Erweiterungen zur reaktiven Planung

Das im vorherigen Kapitel dargestellte Modell ist ein Neuplanungsverfahren. Bei jeder Lösung des Problems wird immer ein komplett neuer auf den aktuellen Daten basierender optimaler Plan erstellt. Sofern die einzelnen Horizonte der verschiedenen Planungen disjunkt sind und eine statische Planungsumgebung unterstellt wird, ist eine solche Planung möglich. Wie in Kapitel 2 bereits diskutiert, trifft eine solche Annahme jedoch für die industrielle Praxis selten zu. Um den Anforderungen einer dynamischen Planungsumgebung Rechnung zu tragen, werden im Folgenden Erweiterungen eingeführt, die eine rollierende Planung unterstützen. Eine rollierende Neuplanung kann jedoch große Planänderungen implizieren und eine große Flexibilität der innerbetrieblichen Abläufe verlangen. So kann bspw. die Werkslogistik vom Produktionsplan abhängig sein und muss dann entsprechend angepasst werden. Aus diesem Grund wird nachfolgend das Modell um den Kostenfaktor Umplanungskosten erweitert, mit dem diese Planänderungen reduziert werden können. Diese Umplanungskosten entstehen durch eine Veränderung des Start- bzw. Endtermins eines Loses und werden entsprechend mit Kosten belastet.

Zur Modellierung wird das Grundmodell erweitert um:

Definition der Mengen

$FSZ = \{1..fsz\}$	Menge der Mikroperioden, in denen eine Fixierung der Schichtmodelle vorgenommen wird
$FZ = \{1..fz\}$	Menge der Mikroperioden, die der eingefrorenen Zone zugeordnet sind und in denen eine Fixierung der Produktion vorgenommen wird

Definition der Parameter

$b_e^g \in \mathbb{N}$	Die Menge des Bedarfs von Erzeugnis e , der am Anfang des Planungshorizontes bereits gedeckt ist
$h_{e,pl,\mu} \in \{0,1\}$	Indikator für die rollierende Planung, der anzeigt, ob das Erzeugnis e im Ursprungsplan in Mikroperiode μ auf der Produktionslinie pl eingeplant ist (vgl. [Neuh08])
$f_{e,pl} \in \mathbb{N}$	Der Rüstzeitfortschritt von Erzeugnis e , der am Anfang des Planungshorizontes bereits gerüstet wurde
$I_{e,pl,\mu}^{vir,r} \in \mathbb{N}$	Die Größe des virtuellen Lagers eines Loses für das Erzeugnis e auf Produktionslinie pl zum Zeitpunkt μ , wenn ein Los bei der rollierenden Planung makroperiodenübergreifend produziert wird
$rp_{pl} \in \mathbb{N}$	Der Rüstzustand einer Produktionslinie pl , der am Anfang des Planungshorizontes vorliegt
$sms_{sm,pl,t} \in \{0,1\}$	Indikator für die Schichtmodellauswahl, der anzeigt, ob das Schichtmodell sm im Ursprungsplan in Makroperiode t auf der Produktionslinie pl eingeplant ist

$$uk_e \in \mathbb{R}^0 \quad \text{Umlanungskosten, die für die Umlanung eines Erzeugnisses } e \text{ anfallen}$$

Damit ergibt sich folgende Zielfunktion (5.53):

$$\min z = \text{Verzugskosten} + \text{Produktionskosten} + \text{Lagerkosten} + \text{Umlanungskosten} \quad (5.53)$$

Die Umlanungskosten errechnen sich mit (5.54):

$$\text{Umlanungskosten} = \sum_{e \in E} \sum_{pl \in PL} \sum_{\mu \in M} (|x_{e,pl,\mu} - h_{e,pl,\mu}|) * uk_e \quad (5.54)$$

Unabhängig von der Art der Planung, d.h., Neu- oder Änderungsplanung, können innerhalb der rollierenden Planung weitere Bedingungen integriert werden. Im Rahmen dieser Arbeit sind dies die eingefrorene Zone – also ein Planungsbereich, in dem keine Änderungen der Planung vorgenommen werden – oder ein fixierter Schichthorizont, in dem keine Adaption der Schichtmodelle erlaubt ist. Zur Modellierung werden $h_{e,pl,\mu}$ und $sms_{sm,pl,t}$ definiert, die die Planvorgaben des vorherigen Planungsschritts abbilden. Die Restriktionen (5.55) und (5.56) weisen den Variablen in der eingefrorenen Zone die Werte zu. Dadurch sind die Werte während der Planung fest vorgegeben und können nicht geändert werden. Außerdem soll keine Produktionskapazität zur Verfügung stehen, wenn die Planvorgaben in der eingefrorenen Zone keine Produktion vorsehen. Es soll $Z_{e,pl,\mu} = 0$ gelten, wenn $x_{e,pl,\mu} = 0$. Analog dazu werden die Schichtmodelle des fixierten Schichthorizontes mit der Restriktion (5.57) dargestellt. Hierbei ist bis zu dem angegebenen Zeitpunkt keine Adaption des Schichtmodells erlaubt.

$$x_{e,pl,\mu} = h_{e,pl,\mu} \quad \forall e \in E \quad (5.55) \\ \forall pl \in PL \\ \forall \mu \in FZ$$

$$Z_{e,pl,\mu} \leq B * h_{e,pl,\mu} \quad \forall e \in E \quad (5.56) \\ \forall pl \in PL \\ \forall \mu \in M$$

$$sms_{sm,pl,t} = sms_{sm,pl,t} \quad \forall sm \in SM \quad (5.57) \\ \forall pl \in PL \\ \forall t \in FSZ$$

Mehrere Spezialfälle können bei der rollierenden Planung auftreten, die gesondert behandelt werden müssen. Sobald mit der Bearbeitung eines Loses begonnen wird, erfolgt eine Aktualisierung des Bestands der benötigten Vorerzeugnisse entsprechend der Größe des gesamten

Loses. Erstreckt sich nun die Länge des zu produzierenden Loses über die erste Makroperiode hinaus, muss beim nächsten Planungslauf die Information übergeben werden, dass für das Los dessen benötigter Bedarf an Vorerzeugnissen bereits in der Vorperiode gedeckt wurde. Hierzu wird der Parameter b_e^g definiert, der in den Anfangslagerbestand der nachfolgenden Planung eingeht. Dieser Zusammenhang wird durch die Restriktion (5.58) dargestellt.

$$\sum_{j \in N_e} \sum_{p \in PL} a_{e,j} * h_{j,pl,0}^s \leq I_e^{init} + b_e^g \quad \forall e \in E \quad (5.58)$$

Es wurde bereits beschrieben, dass die produzierten Einheiten eines nicht-abgeschlossenen Loses in einem virtuellen Lager gespeichert werden. Erstreckt sich nun die Produktion eines Loses über die erste Makroperiode hinaus, muss zudem dieser Wert übergeben werden. Dazu wird der Parameter $I_{e,pl,\mu}^{vir,r}$ herangezogen und innerhalb der eingefrorenen Zone durch die Gleichung (5.59) dem virtuellen Lager zugewiesen.

$$I_{e,pl,\mu}^{vir} = I_{e,pl,\mu}^{vir,r} \quad \forall e \in E \quad (5.59)$$

$$\forall pl \in PL$$

$$\forall \mu \in FZ$$

Beim rollierenden Fortschreiten des Planungshorizontes muss außerdem der Rüstzustand einer Produktionslinie berücksichtigt werden. Dieser Zustand wird mit Hilfe des Parameters rp_{pl} abgebildet, der den jeweils letzten vollständigen Rüstzustand aufnimmt. Wurde in der Vorperiode bereits mit dem Rüsten auf ein Nachfolgeerzeugnis begonnen, wird dieser Fortschritt dem Parameter $f_{e,pl}$ zugewiesen und die verbleibende Rüstzeit entsprechend reduziert. Hierzu dient die Restriktion (5.60).

$$\sum_{\mu < \mu_1} \sum_{sm \in SM_{pl}[\frac{\mu}{MM}]} lg_{sm,pl,\mu} \cdot sm_{sm,pl,[\frac{\mu}{MM}]} \geq \sum_{e \in E} (rz_{rp_{pl},e,pl} - f_{e,pl}) * x_{e,pl,\mu_1} \quad f_{e,pl} > 0 \quad (5.60)$$

$$\forall pl \in PL$$

$$\forall \mu_1: 1 \leq \mu_1 < mr_{pl}$$

Zur korrekten Variablenbelegung sind zudem die Restriktionen für das virtuelle Lager entsprechend der Ungleichungen (5.61) und (5.62) zu erweitern.

$$I_{e,pl,\mu}^{vir} - \left(\frac{Z_{e,pl,\mu}}{tb_{e,pl}} \right) \leq B * (1 - x_{e,pl,\mu}) \quad \forall e \in E \quad (5.61)$$

$$\forall pl \in PL$$

$$\mu = 1$$

$$I_{e,pl,\mu}^{vir,r} = 0$$

$$\begin{aligned}
 I_{e,pl,\mu}^{vir} - I_{e,pl,\mu}^{vir,r} &\leq B * (1 - x_{e,pl,\mu}) & \forall e \in E & (5.62) \\
 & & \forall pl \in PL \\
 & & \mu = 1 \\
 & & I_{e,pl,\mu}^{vir,r} > 0
 \end{aligned}$$

Neben diesen zusätzlich notwendigen Restriktionen entfällt zur rollierenden Planung die Restriktion (5.50).

5.1.3 Gewichtete Zielfunktion

Zur Erstellung spezieller Zielausrichtungen ist es möglich, die Bestandteile der Zielfunktion unterschiedlich zu gewichten. Dadurch verliert die Zielfunktion jedoch den realen Kostenbezug, die einzelnen Bestandteile erhalten eher den Charakter eines Steuerparameters. Dies ermöglicht bspw., durch eine sehr hohe Gewichtung der Lagerkosten und eine sehr geringe Gewichtung der Produktionskosten die Optimierung auf kurze Durchlaufzeiten auszurichten.

Zur Gewichtung wird ein nicht-negativer Gewichtungsfaktor g_x einem Zielkriterium zugeordnet, wobei die Verhältnisse unter den Zielgewichtungsfaktoren die Grenzwerte der Substitution der Zielsetzungen ausdrücken (vgl. [BDS04]). Dadurch resultieren die in (5.63) und (5.64) dargestellten Zielfunktionen.

$$\min z = g_v * Verzugskosten + g_p * Produktionskosten + g_l * Lagerkosten \quad (5.63)$$

$$\min z = g_v * Verzugskosten + g_p * Produktionskosten + g_l * Lagerkosten + g_u * Umlanungskosten \quad (5.64)$$

5.2 Entwicklung eines heuristischen Verfahrens

In diesem Kapitel wird zur Problemlösung ein heuristischer Lösungsweg dargestellt. Zu Beginn wird der generelle Aufbau des gesamten Verfahrens erläutert. Anschließend werden die einzelnen Bestandteile im Detail betrachtet.

5.2.1 Aufbau des Planungsverfahrens

In den vorherigen Kapiteln wurde deutlich, dass die detaillierte Planung eines Flexible Flow Shop einer hohen Laufzeit bedarf. Aufgrund der dynamischen Planungsumgebung können jedoch um so geringere Ansprüche an den Detaillierungsgrad der Planung eines Zeitabschnittes gestellt werden, desto ferner diese Periode in der Zukunft liegt. Um diesen Zusammenhang abzubilden, wird der Planungshorizont in verschiedene Abschnitte eingeteilt. Dies sind der Feinplanungshorizont, der u.a. die eingefrorene Zone und den fixierten Schichthorizont beinhaltet, sowie der Grobplanungshorizont (siehe Abbildung 20).

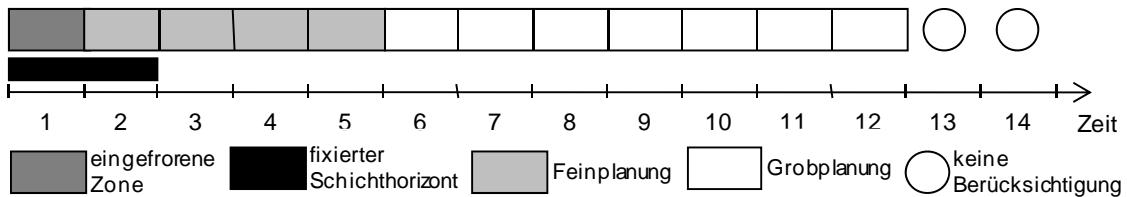


Abbildung 20: Horizonte der Planung

Im Feinplanungshorizont erfolgt eine sehr detaillierte Planung bei T_0 beginnend für x Makroperioden. Dies erlaubt z.B. Aussagen über die Reihenfolge von Losen oder aggregierter Schichtmodelle. Grundlage der Feinplanung ist damit ein Zeitmodell, in dem die Planung auf Mikroperioden basiert. Um in der Feinplanung unterschiedliche Flexibilität abzubilden, wird dieser Planungsbereich weiter unterteilt. So ist in der eingefrorenen Zone eine Anpassung des Produktionsplans mittels eines Planungsverfahrens unzulässig. D.h., dass eine Planänderung nur manuell erfolgen kann und generell nur durch außerplanmäßige Ereignisse wie z.B. einem Maschinenausfall initiiert wird. Die eingefrorene Zone ($T_{ps}^{b,eZ}$) startet zu Beginn des Planungshorizontes T_0 und umfasst mindestens 1 Makroperiode ($T_{ps}^{e,eZ}$). Zudem unterliegt die flexible Einplanung der oben diskutierten Schichtmodelle häufig gewissen Einschränkungen. So können z.B. Vereinbarungen mit dem Betriebsrat oder externen Personaldienstleistern einen fixierten Schichthorizont verlangen. In diesem Fall darf keine Anpassung der Kapazitätsangebote erfolgen. Alle weiteren Anpassungen, z.B. eine Veränderung der Ausführungsreihenfolge der Lose, sind jedoch in dem fixierten Schichthorizont möglich. Hierbei gilt $T_{ps}^{e,eZ} \leq T_{ps}^{e,FS}$. Der letzte Abschnitt ist der Grobplanungsbereich ($T_{ps}^{e,FP} < T_{ps}^{b,GP}$), dessen Ziel eine grobe Bestimmung zukünftiger Produktionspläne ist. Diese Bestimmung erlaubt, notwendige Rohmaterialien und Zwischenerzeugnisse zu beschaffen sowie ggf. bereits im Vorfeld adäquate Maßnahmen zur Vermeidung späterer Konflikte (z.B. Bedarfsspitzen als Folge von Werbeaktionen) zu ergreifen. Die Planung in dem Grobplanungshorizont erfolgt auf Basis von Makroperioden.

Entsprechend dieses zugrundeliegenden Zeitmodells untergliedert sich auch das im Weiteren vorgestellte Verfahren in ein Fein- und Grobplanungsverfahren. Grundlage dieser Verfahren ist eine vom Planer vorzunehmende Parametrisierung (siehe 5.2.2). Diese findet generell nur einmal statt und wird danach nur in Ausnahmefällen verändert. Der weitere Aufbau und Ablauf des Planungsverfahrens ist in der Abbildung 21 dargestellt. Demnach unterteilt sich das Feinplanungsverfahren in ein Eröffnungs- und ein Verbesserungsverfahren. Das Ziel des Eröffnungsverfahrens besteht darin, für die jeweils neue Planungssituation eine erste gültige Lösung auf Basis des Ergebnisses des vorherigen Planungslaufs zu finden. Ein Plan wird als gültig bezeichnet, wenn er sämtliche unterstellten Restriktionen (z.B. Kapazitätsbegrenzungen oder zu beachtende Vorlaufzeiten zwischen den einzelnen Produktionsstufen) einhält. Nach Abschluss des Eröffnungsverfahrens liegt ein realisierbarer Plan vor, der jedoch wahrscheinlich nicht der Forderung nach Optimalität genügt. Der Forderung nach einer möglichst wirtschaftlichen Lösung wird im Rahmen eines Verbesserungsverfahrens Rechnung getragen. In diesem wird unter Beachtung einer vorher festzulegenden maximalen Anzahl an Iterationen ein bestmöglicher Plan gesucht. Dazu wird der von dem Eröffnungsverfahren erzeugte Plan im Rahmen erlaubter Bereiche stufenweise verbessert. Der Input des Feinplanungsverfahrens

besteht damit zum Planungszeitpunkt $PZ(f)$ aus dem für diese Perioden vormals ermittelten Plan ($PZ(f, \dots, g-1)$), dessen etwaigen Bedarfsveränderungen ($PZ(\Delta f, \dots, \Delta g-1)$) sowie den Bedarfsumfängen für die neue Makroperiode $PZ(g)$. Letztere basieren auf den in der Grobplanung ermittelten Produktionsterminen. Zur Ermittlung dieser dienen ebenfalls zwei Verfahren. Nach dem Import neuer Bedarfe für $PZ(t+n+1)$ sowie der Aktualisierung der Bedarfe für $PZ(g, \dots, t+n)$ erfolgt die Planung mittels eines erweiterten MLCLSP (siehe 3.1.1). Wenn den vorhandenen Bedarfen nicht ausreichend Kapazitäten gegenüberstehen, können die Bedarfe nur mit Verzug oder ggf. gar nicht eingeplant werden. Zur Berücksichtigung etwaiger freier Kapazitäten in der Feinplanung werden in diesem Fall Bedarfe in diesen Planungsbereich verschoben. Zur Identifikation entsprechender Bedarfe dient das Bedarfsglättungsverfahren.⁶²

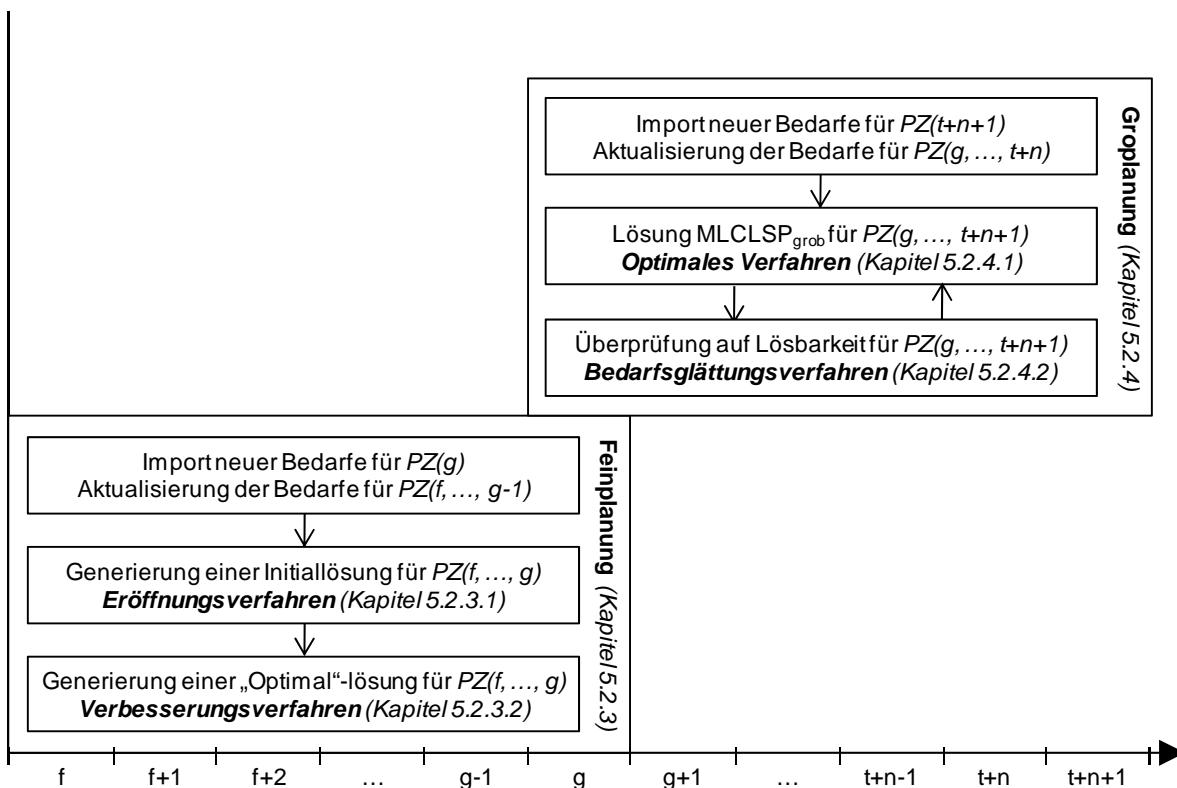


Abbildung 21: Gesamtablauf des heuristischen Verfahrens

5.2.2 Parametrisierung

Das Planungsverfahren verfügt über Parameter, mit denen die Planungskomplexität und die Zielsetzung beeinflusst werden können. Dies ermöglicht dem Planer, das Verfahren unterschiedlichen Bedingungen anzupassen oder die Zielsetzung durch die Gewichtung von Steuerparametern zu variieren.

⁶² Dieses Vorgehen bedingt, dass der $PZ(g)$ in einem Planungsdurchlauf zwei Mal geplant wird. Dadurch wird gewährleistet, dass für diese Makroperiode in der Feinplanung aktuelle Werte vorliegen.

Die folgenden Parameter bestimmen den Aufbau und Ablauf der Planungsverfahren.

- Länge des Feinplanungshorizontes: Die Länge der Planungshorizonte stellt einen entscheidenden Faktor für die Anwendbarkeit der Produktionspläne dar. Ist er zu kurz gewählt, kann nicht rechtzeitig auf große Bedarfsanforderungen reagiert werden, während ein zu langer Zeitraum die Komplexität erhöht und somit die Laufzeit des Planungsverfahrens negativ beeinträchtigt. Es kann jedoch kein allgemeingültiger Wert definiert werden, da dieser stark von der jeweiligen Bestellstruktur (inkl. Fristen, Verbindlichkeit der Bestellung, etc.) als auch von der Fertigungsstruktur (insbesondere der Durchlaufzeit eines kompletten Erzeugnisses inkl. aller Vorerzeugnisse) abhängt.
- Länge der eingefrorenen Zone: Dieser Parameter definiert den Aufsetzpunkt für die Planung.
- Länge der fixierten Schichten: In diesem Bereich ist eine Anpassung der Kapazitätsangebote durch das heuristische Verfahren nicht möglich. Wenn dieser Bereich sehr groß gewählt wird, reduziert sich entsprechend die Flexibilität der Planung. Sollte dieser Wert gleich der Länge des Feinplanungshorizontes sein, wird eine Planung unter Nutzung einer festzulegenden Standardkapazität vorgenommen.
- Länge des Grobplanungshorizontes: Analog zur Länge des Feinplanungshorizontes wird auch die Länge des Grobplanungshorizontes festgelegt.
- Planen mit Verzug: Mit der Einstellung von diesem Parameter wird eine Planung mit Verzug erlaubt oder ausgeschlossen. Wird das Planen mit Verzug zugelassen, wird die beste Kombination aus Produktions-, Lager- und Verzugskosten gesucht, ansonsten aus Produktions- und Lagerkosten. Damit wird auch zugelassen, dass ein Bedarf trotz der Möglichkeit zur termingerechten Fertigstellung ggf. verspätet eingeplant wird. Dies ist dann der Fall, wenn die Summe aus Produktions- und Verzugskosten kleiner als die Summe aus Produktions- und Lagerkosten ist.
- Maximale Anzahl an Splittungen: Ist in dem Eröffnungsverfahren die Einplanung eines Bedarfs nicht möglich, wird dieser gesplittet. Je höher aber diese Anzahl an Splittungen ist, desto mehr Lose – mit einer entsprechenden höheren Laufzeit – sind zu berücksichtigen. Aus diesem Grund kann eine obere Grenze definiert werden.
- Flexibilität der Schichtmodelle: Bei einer großen Flexibilität kann es zu einer großen Nervosität der Schichtmodelle kommen, da bei Bedarfsänderungen stark mit Schwankungen der Arbeitszeiten reagiert werden kann. Diese Schwankungen können über diesen Parameter gesteuert werden.
- Maximale Reichweite zu verschiebender Lose in x Zeiteinheiten: Dieser Parameter begrenzt die Reichweite eines zu verschiebenden Loses. So kann bspw. die Startposition eines Loses um nur 2 Makroperioden in die Zukunft oder Vergangenheit verschoben werden. Ein Grund für die Beschränkung können bspw. Absprachen mit der Werkslogistik oder mit externen Lieferanten sein.
- Abkühlungsfaktor für das Verbesserungsverfahren Simulated Annealing: Mit diesem Parameter wird die Abbruchbedingung spezifiziert. (siehe Kapitel 3.2.2.2)
- Initialtemperatur für das Verbesserungsverfahren Simulated Annealing: Dieser Parameter beschreibt die Anfangstemperatur. (siehe Kapitel 3.2.2.2)

- Länge der Markov-Kette für das Verbesserungsverfahren Simulated Annealing: Dieser Parameter stellt das Abbruchkriterium für die innere Schleife des Verfahrens dar (siehe Abbildung 12).
- Länge der Tabu-Liste für das Verbesserungsverfahren Tabu Search: Zur Vermeidung von Zyklen speichert die Tabu-Liste Komplemente der durchgeföhrten Züge. Während eine zu lange Liste die Suche stark einschränkt, führt eine zu kurze Liste zu Zyklen. (siehe Kapitel 3.2.2.3)
- Maximale Anzahl an Iterationen der Verbesserungsverfahren: Als weiteres Abbruchkriterium gilt eine maximale Anzahl an Durchläufen. Ist dieser Wert erreicht, beendet das Verbesserungsverfahren die Suche nach alternativen Lösungen. Dieses Abbruchkriterium ist notwendig, da es keine Garantie dafür gibt, mit dem Verbesserungsverfahren die optimale Lösung zu finden, bei der das Verfahren terminiert.
- Maximale Rechenzeit der Verbesserungsverfahren: Dieses Abbruchkriterium beschränkt die Laufzeit eines Verfahrensschrittes. Wird diese Zeit erreicht, wird die bis dahin gefundene beste Lösung für den nächsten Schritt verwendet.

Die Zielausrichtung des heuristischen Verfahrens ist prinzipiell die Minimierung der Gesamtproduktionskosten, wobei sich diese aus mehreren Faktoren zusammensetzen. Um spezielle Zielausrichtungen verfolgen zu können, kann jeder Faktor mit einem separaten Gewicht beeinflusst werden. Es folgt eine Auflistung der möglichen Planungsziele und der hierfür relevanten Parameter.

- Minimierung der Lagerkosten: Dieses Zielkriterium befindet darüber, wie wichtig eine Einplanung von Losen nahe am tatsächlichen Bedarfstermin ist. Sofern bei einem Unternehmen häufig Stornierungen eingehen, könnte dadurch die vorzeitige Produktion später nicht mehr benötigter Erzeugnisse vermieden werden. Damit hat dieses Zielkriterium auch Einfluss auf die Gesamtdurchlaufzeit. Wird dieser Steuerparameter sehr hoch angesetzt, werden die Erzeugnisse nahe am Bedarfstermin produziert und auch auf den Zwischenstufen nicht lange zwischengelagert.
- Minimierung der Verzugskosten: Sofern Verzugskosten nicht quantifiziert werden können, werden häufig fiktive Werte verwendet (eine Ausnahme bilden Konventionalstrafen). Aus diesem Grund ist es nicht immer zweckdienlich, die Verzugskosten mit diesen Werten zu berücksichtigen. Durch eine bspw. hohe Gewichtung dieses Parameters könnte jedoch in einer kritischen Situation eine Abwanderung eines Kunden zur Konkurrenz vermieden werden.
- Minimierung der Produktionskosten: Hierbei wird jeweils ein besonderes Gewicht auf die Auswahl günstiger Schichtmodelle gelegt. Dies kann Folgen für die gesamte Durchlaufzeit und die intendierte Lieferfähigkeit haben.

5.2.3 Entwicklung eines Verfahrens für den Horizont der Feinplanung

Wie der Abbildung 21 zu entnehmen ist, besteht das Feinplanungsverfahren aus einem Eröffnungs- und einem Verbesserungsverfahren. Diese werden nachfolgend beschrieben.⁶³

5.2.3.1 Eröffnungsverfahren

In der Realität kann es aufgrund vielfältiger Gründe (siehe Kapitel 2.1.3.2) notwendig sein, die Planung anzupassen. Häufig ist es nicht zweckmäßig, jedes Mal eine vollständig neue Planung durchzuführen. Dieses resultiert zum einen aus dem häufig großen Rechenbedarf der Produktionsplanungsverfahren (siehe Kapitel 3) und zum anderen aus den jeweils mit der Planung verbundenen Abhängigkeiten. Insofern besteht in dieser Arbeit das Ziel in dem möglichst guten Einarbeiten auftretender Bedarfsveränderungen bei gleichzeitig nur geringen Anpassungen des Fertigungsprogramms.

Im Stand der Technik (siehe Kapitel 3) wurde deutlich, dass eine simultane Betrachtung aller Restriktionen von großer Komplexität und unter praxisrelevanten Bedingungen auch mit heutiger Technik nicht in adäquater Zeit lösbar ist. Aus diesem Grund ist das im Weiteren vorgestellte Eröffnungsverfahren nach dem Konzept der Sukzessivplanung mit Rückkopplung aufgebaut. In diesem beginnt die Planung mit der ersten Produktionsstufe ($ps = 1$) und wird entgegen der Materialflussrichtung bis zur letzten Produktionsstufe ($ps = n^{ps}$) durchgeführt. Zur Berücksichtigung von Interdependenzen zwischen Produktionsstufen erfolgt ein Abgleich durch integrierte Planungsschleifen. Entsprechend dieses Konzeptes werden bei der Einplanung von Bedarfen immer zeitliche Abhängigkeiten zwischen zwei Produktionsstufen berücksichtigt. Nur wenn alle Lose zweier benachbarter Produktionsstufen konfliktfrei eingeplant wurden, startet auf der nächsten Produktionsstufe der Prozess von vorne. Sofern auf weiter vorgelagerten Produktionsstufen Probleme auftreten, werden diese soweit nach vorne gerichtet, bis die Probleme komplett gelöst sind. Sofern das nicht realisiert werden kann, werden die Bedarfe in einer Liste zur manuellen Einplanung gespeichert. Die Zielsetzung entspricht der des mathematischen Modells (siehe (5.63)).

Nach diesem kurzen Überblick werden nun die einzelnen Bausteine des Verfahrens im Detail betrachtet. Um den Zusammenhang und den zeitlichen Ablauf des Prozesses übersichtlich darzustellen, dient Abbildung 22.

⁶³ Zur Vermeidung von Redundanz, werden – sofern möglich – für beide Verfahren die gleichen Funktionen verwendet. Damit sind dies faktisch nicht zwei getrennte Verfahren. Zur Erhöhung der Lesbarkeit werden die beiden Verfahrenstypen jedoch getrennt voneinander beschrieben. An entsprechenden Stellen werden Verweise auf die Funktionsbeschreibungen gegeben.

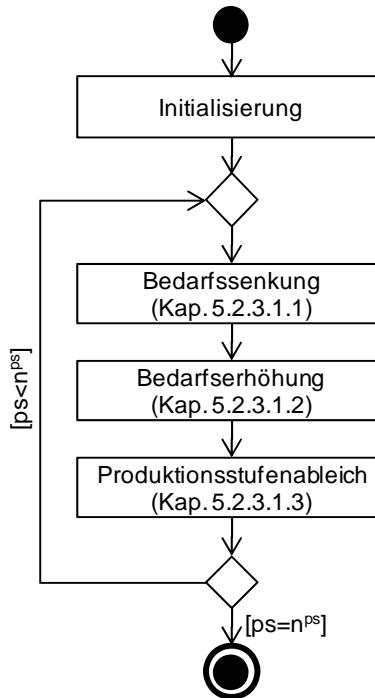


Abbildung 22: Gesamtaufbau des Eröffnungsverfahrens

Zu Beginn des Verfahrens sind verschiedene Vorarbeiten nötig. Ein Beispiel stellt die Initialisierung dar, in der der zur Ausführung benötigte Speicherplatz für das Programm reserviert und mit Startwerten gefüllt wird. Da dies jedoch nichts Spezifisches des Verfahrens darstellt, soll dieser Vorgang hier nicht weiter betrachtet werden. Im Rahmen der drei Hauptfunktionen kommen verschiedene Hilfsfunktionen zum Einsatz. Die umfangreicheren Hilfsfunktionen werden in Kapitel 5.2.3.1.4 erläutert.

5.2.3.1.1 Bedarfssenkung

Wie der Abbildung 22 zu entnehmen ist, werden generell zwei Bedarfstypen unterschieden⁶⁴. Dies sind die Bedarfssenkungen (BS) und die Bedarfserhöhungen (BE). Der Ablauf einer Bedarfssenkung ist in Abbildung 23 dargestellt.

⁶⁴ Die jeweiligen Umfänge sind zu Beginn eines jeden Planungslaufs zur Verfügung zu stellen und können bspw. aus dem im Unternehmen installierten ERP-System in die Planungskomponente importiert werden.

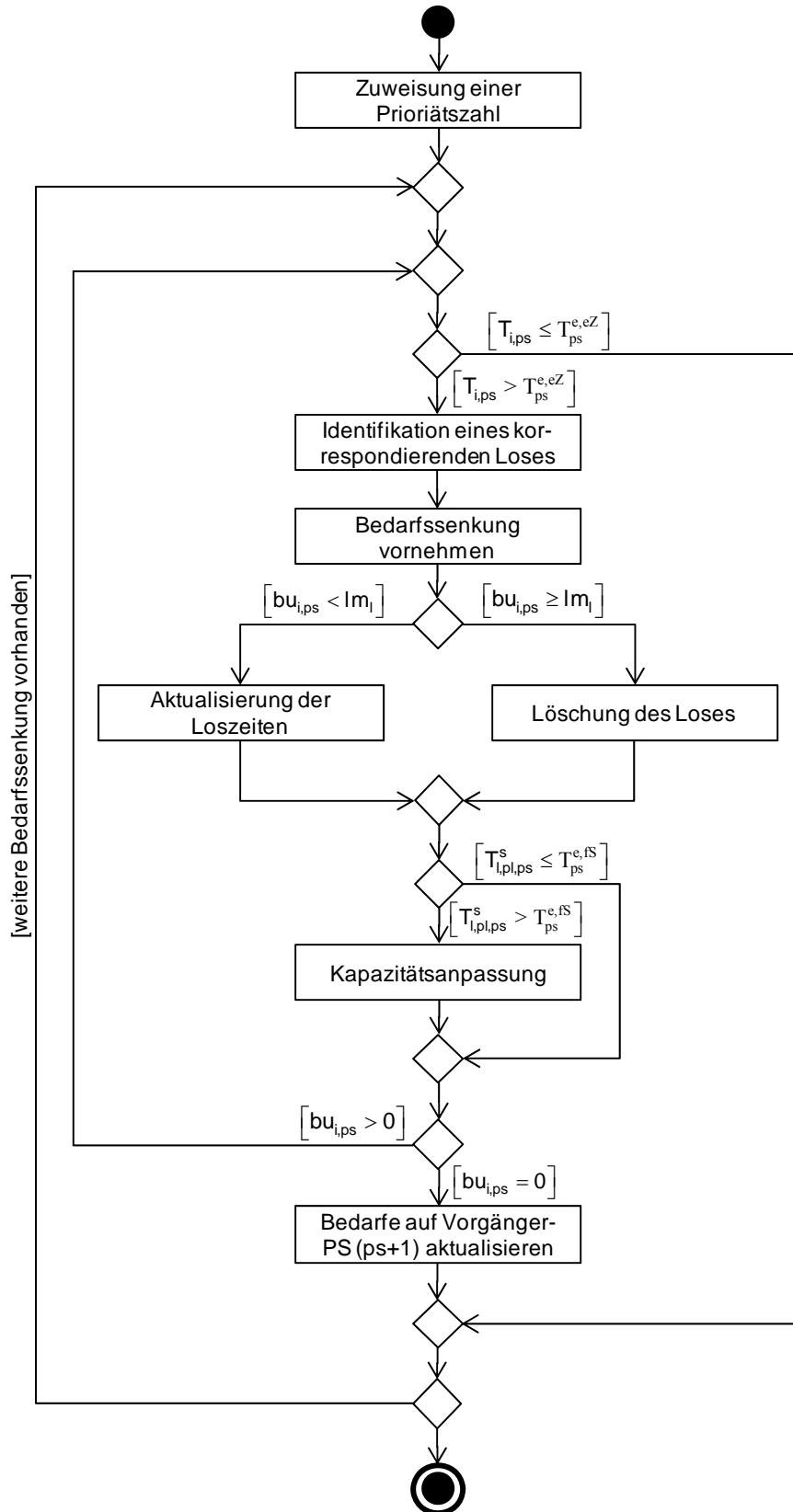


Abbildung 23: Ablauf einer Bedarfssenkung

Zu Beginn des Verfahrens wird jeder Bedarfssenkung entsprechend der FFT-Regel (siehe Kapitel 3.2.1.1) eine Prioritätszahl zugewiesen. Demnach werden die Bedarfstermine ($T_{i,ps}$)

einer Bedarfssenkung i so geordnet, dass sie einer zeitlich aufsteigenden Sortierung folgen ($T_{i,ps} \leq T_{i+1,ps}$). Der Bedarfstermin ergibt sich in Abhängigkeit zu der Produktionsstufe. Auf der ersten Produktionsstufe leitet er sich direkt aus dem entsprechenden Kundenauftrag ab. Auf allen anderen Produktionsstufen resultiert er aus der Startzeit des korrespondierenden Loses l auf der nachgelagerten Produktionsstufe ($ps - 1$) sowie einer etwaigen Vorlaufzeit. Als Informationen zur Bedarfssenkung stehen das Erzeugnis, der Bedarfstermin und die Anzahl, um die der Auftrag verringert werden soll, zur Verfügung. Bedarfssenkungen, die zeitlich in den Bereich der eingefrorenen Zone fallen ($T_{i,ps} \leq T_{ps}^{e,EZ}$), werden herausgefiltert und in einer Liste zur manuellen Planung gespeichert. Wie mit diesen Bedarfssenkungen verfahren wird, muss fallbasiert vom verantwortlichen Planer entschieden werden. So kann er bspw. Aufträge gleichen Erzeugnisses im Bereich außerhalb der eingefrorenen Zone suchen und entsprechend reduzieren – natürlich mit dem Nachteil zusätzlicher Lagerkosten für die in der eingefrorenen Zone produzierten Menge – oder die Menge in der eingefrorenen Zone reduzieren, bzw. durch ein anderes geeignetes Erzeugnis ersetzen. Für alle weiteren Bedarfssenkungen sind die zugehörigen Lose zu finden und zu reduzieren⁶⁵. Zur Identifizierung des passenden Loses erfolgt ein Vergleich von Erzeugnis und Auftragsnummer. Nachdem das passende Los l identifiziert wurde, erfolgt die Berechnung der verbleibenden Losgröße. Wenn der Umfang der Bedarfssenkung ($bu_{i,ps}$) kleiner ist als das identifizierte Los ($bu_{i,ps} < lm_l$), wird das Los entsprechend reduziert und dessen Startzeit neu berechnet (*KalkStartzeit*⁶⁶, siehe Kapitel 5.2.3.1.4.1). Ist hingegen die Bedarfssenkung genauso groß oder größer als das identifizierte Los ($bu_{i,ps} \geq lm_l$), wird das gesamte Los gelöscht und der Umfang der Bedarfssenkung ($bu_{i,ps} = bu_{i,ps} - lm_l$) angepasst. Durch eine Bedarfssenkung besteht in der entsprechenden Makroperiode und Produktionslinie ein verminderter Kapazitätsbedarf. Wird das zu dem Bedarf korrespondierende Los gelöscht, muss neben der Bearbeitungszeit auch die Differenz (rz^{diff}) der Rüstzeiten (rz) der betroffenen Lose (Vorgängerlos, Nachfolgerlos und zu lösches Los) berücksichtigt werden. Zur Ermittlung dient die Formel (5.65).

$$rz^{diff} = rz_{l-1,l,pl} + rz_{l,l+1,pl} - rz_{l-1,l+1,pl} \quad (5.65)$$

Auf Basis dieser Information ist zu prüfen, ob durch die Reduktion ein günstigeres Schichtmodell gewählt werden kann. Dies ist jedoch nur im Bereich außerhalb des fixierten Schichthorizontes ($> T_{ps}^{e,fS}$) möglich. Erfolgt eine Reduktion innerhalb des Bereiches, entstehen Lücken, die durch das Einplanen oder im Zuge des Verbesserungsverfahrens geschlossen werden müssen. Zur Anpassung der Kapazität wird die Methode *DekremSchichtmodell* (siehe Kapitel 5.2.3.1.4.2) aufgerufen. Nachdem eine Bedarfssenkung abgeschlossen ist, werden mengen- und terminmäßige Bedarfsänderungen für die vorgelagerte Produktionsstufe ($> n^{ps}$) erzeugt. Anschließend wird – sofern vorhanden – die nächste Bedarfssenkung betrachtet.

⁶⁵ Jede Planänderung – sowohl im Eröffnungs- als auch im Verbesserungsverfahren – wird auf Basis einer Kopie des Ursprungsplans vorgenommen. Resultiert aus einer Planänderung ein ungültiger irreparabler Plan, wird diese Kopie verworfen und auf Basis des originären Plans nach einer Alternative gesucht.

⁶⁶ Im Anhang A befindet sich der gesamte Pseudo-Code des Eröffnungs- und Verbesserungsverfahrens. Funktionen des Verfahrens werden im Weiteren *kursiv* dargestellt.

Der Ablauf einer Bedarfssenkung wird in Abbildung 24 dargestellt. Vor der Reduktion ist ein Schichtmodell mit zwei Schichten sowie jeweils einer Schichtlänge von zehn Stunden ausgewählt. Nach der Reduktion des Loses liegt eine Lücke mit nicht genutzter Kapazität vor. Um diese zu schließen, wird der Makroperiode ein Schichtmodell mit geringerer Kapazität zugewiesen. In diesem Beispiel ist das ein Schichtmodell mit zwei Schichten mit jeweils einer Länge von 8 Stunden. Bei den Schichtmodellen wird ein identischer Leistungsgrad der Schichten unterstellt. Die Start- und Endzeitpunkte der Lose müssen für das geänderte Schichtmodell neu berechnet werden. Die Reihenfolge der Lose bleibt unverändert.

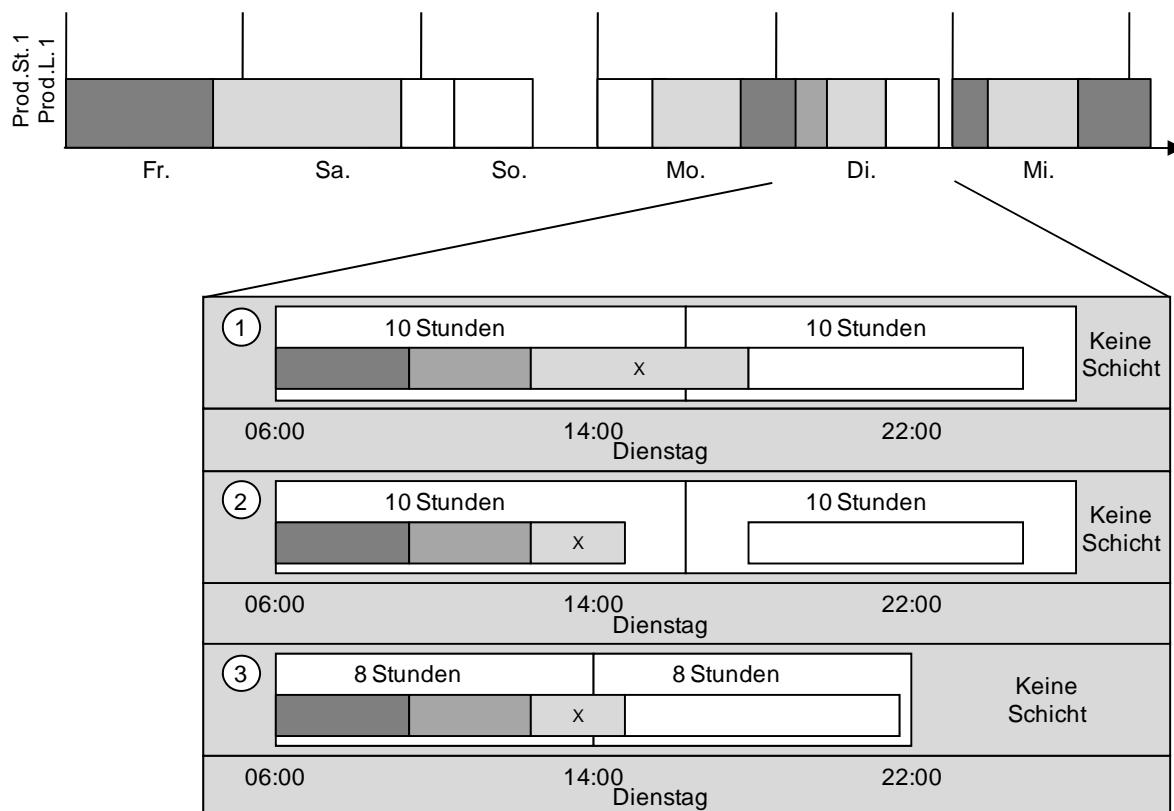


Abbildung 24: Beispiel einer Bedarfssenkung

5.2.3.1.2 Bedarfserhöhung

Die zweite Liste stellt Bedarfserhöhungen dar. Der Umfang der Bedarfserhöhungen ist eine Zusammenfassung von Kunden-Bedarfserhöhungen, neuen Bestellungen im vormalen betrachteten Planungszeitraum, im vorherigen Planungslauf nicht einplanbare Bedarfe und Bedarfe, die aus dem Fortschreiten des Feinplanungshorizontes alle x Zeiteinheiten (z.B. 24 Stunden) resultieren.

Im Kapitel 3.2.3 wurden unterschiedliche Maßnahmen zur Änderungsplanung aufgezeigt. Zur Integration von Bedarfserhöhungen finden im Folgenden das Right Shifting, Left Shifting⁶⁷ sowie die Splittung Anwendung. Basis für die Einplanung dieser Bedarfserhöhung ist der

⁶⁷ Analog zu dem Right Shifting (siehe Kapitel 3.2.3.1) verhält sich das Left Shifting, nur dass die Lose zur Be seitigung einer Konfliktsituation zeitlich in Richtung der Heute-Linie verschoben werden.

vormals reduzierte Plan (siehe Kapitel 5.2.3.1.1). Bei der Erweiterung dieses Plans ist auf eine wirtschaftliche Nutzung der Kapazitäten zu achten. Solange die neuen Lose außerhalb des fixierten Schichthorizontes ($> T_{ps}^{e,fS}$) liegen, können die Kapazitätsangebote adaptiert werden. Bei der Einplanung jeden Loses wird die Position auf jeder Produktionslinie gesucht, auf der die Summe aus Produktionskosten, Lagerkosten und Verzugskosten am geringsten ist. Der Ablauf dieser Funktion ist in Abbildung 25 dargestellt.

Zu Beginn des Verfahrens werden die Bedarfserhöhungen zeitlich aufsteigend sortiert (FFT-Regel, siehe Kapitel 3.2.1.1). Auf der Basis dieser Reihenfolge erfolgt anschließend die Prüfung, ob der Bedarf zum Bedarfstermin mit etwaigen Lagerbeständen gedeckt werden kann. Sofern Lagerbestände genutzt werden können, wird der entsprechende Bedarfsumfang reduziert, bzw. der Bedarf als bearbeitet markiert. Im Anschluss daran erfolgt die Analyse, ob bereits in einem vorherigen Planungslauf eingeplante Lose den gleichen Lagerbestand berücksichtigt haben. In dem Fall ist in Höhe des neu genutzten Lagerbestands ein neues Los mit dem Bedarfstermin des später eingeplanten Loses zu erzeugen. Nachfolgend werden die noch ungeplanten Bedarfserhöhungen bearbeitet. Hierzu werden diese zeitlich absteigend sortiert und analysiert. Liegt der Bedarfstermin in der eingefrorenen Zone ($\leq T_{ps}^{e,fS}$), ist keine automatische Planung möglich. Analog zur Bedarfssenkung ist dieser Bedarf dann der Liste nicht einplanbarer Bedarfe hinzuzufügen. Entsprechend der Sortierung werden die Bedarfe beginnend mit der spätesten Bedarfserhöhung iterativ abgearbeitet. Hierzu werden die Methoden *KalkLosdetails* – zur Erzeugung des im Weiteren zu spezifizierenden Loses sowie der Bedarfsmengen der Vorgänger-PS ($ps + 1$) – und *KalkLosposition* (siehe Kapitel 5.2.3.1.4.7) aufgerufen. Diese durchsucht den Lösungsraum nach der günstigsten gültigen Kombination von Zeit, Produktionslinie, Reihenfolge und Schichtmodell.

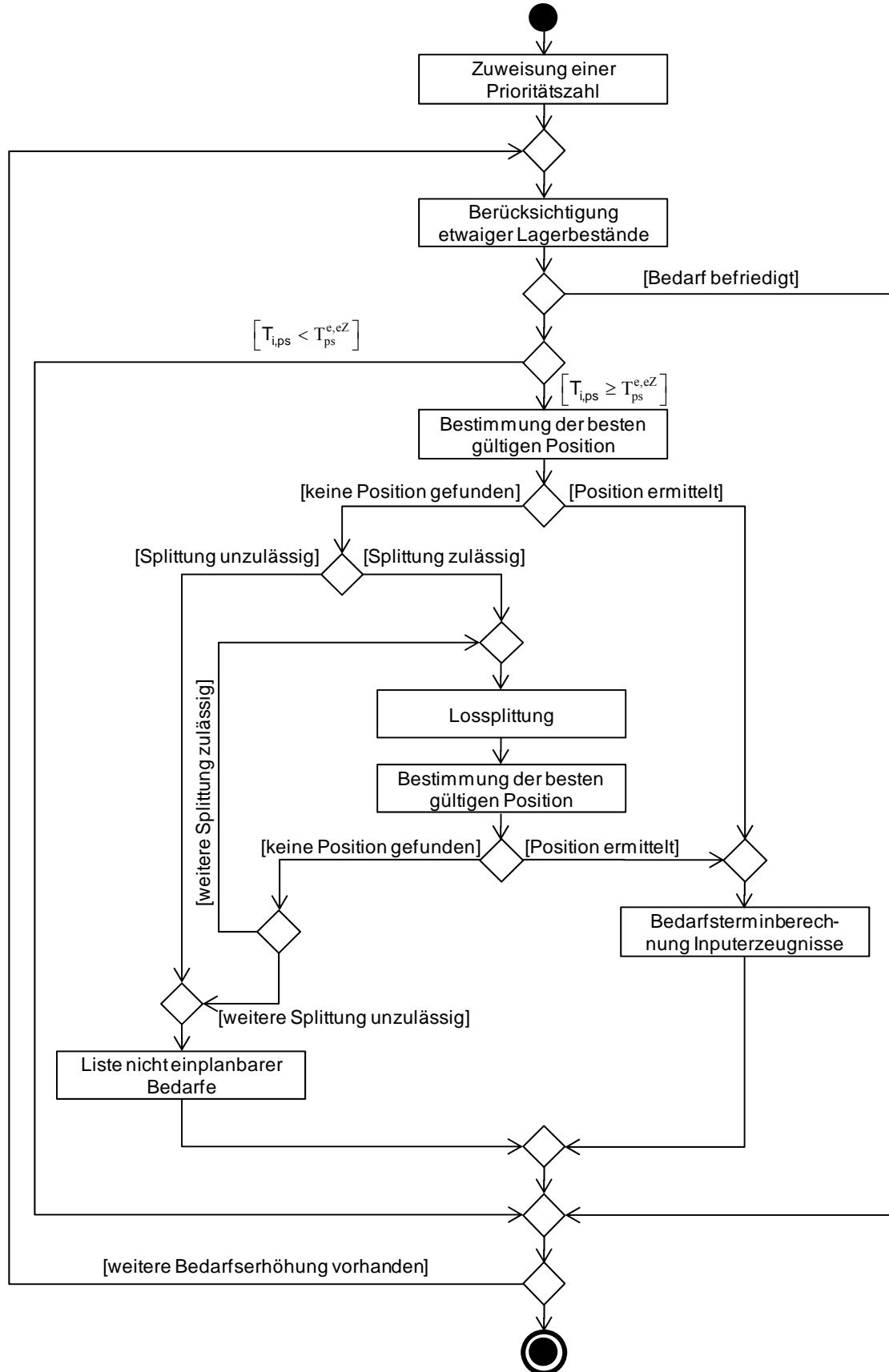


Abbildung 25: Ablauf der Bedarfserhöhung

Wird bei dieser Suche keine Position für die Einplanung des gesamten Bedarfs gefunden, erfolgt, sofern durch die Parametrisierung erlaubt, die Suche mit gesplitteten Bedarfen⁶⁸. Hierzu wird der Bedarf in zwei gleich große Teile gesplittet und versucht einzuplanen (hierzu dienen abermals die Methoden *KalkLosdetails* sowie *KalkLosposition* (siehe Kapitel 5.2.3.1.4.7)). Zur Begrenzung der Rechenzeit und zur Vermeidung zu kleiner Lose wird der Bedarf markiert und die Anzahl der Splittversuche begrenzt (siehe Parameter „Maximale Anzahl an Splittungen“ in Kapitel 5.2.2).⁶⁹ Wird nach dem Durchlauf der Funktion festgestellt, dass es keine gültige Einfügeposition gibt, wird dieser Teil über alle Produktionsstufen aus dem Produktionsplan entfernt und der Liste nicht einplanbarer Bedarfe hinzugefügt.

Dieses Vorgehen wird solange wiederholt, bis die Liste der Bedarfserhöhungen abgearbeitet ist. Wenn Lose nicht fristgerecht eingeplant werden, können auf der nachgelagerten Produktionsstufe die Lose nicht wie geplant beginnen. Dies hat zur Folge, dass der bereits erstellte Plan für die nachgelagerte Produktionsstufe ungültig wird. Um die Gültigkeit des Plans wieder herzustellen, erfolgt die Neueinplanung der betroffenen Lose auf der nachgelagerten Produktionsstufe (siehe Kapitel 5.2.3.1.3). Hierdurch werden die Pläne der einzelnen Produktionsstufen aufeinander abgestimmt und die Konsistenz des Gesamtplans wird sichergestellt.

Das Bestimmen der günstigsten Einfügeposition für eine Makroperiode und Produktionslinie ist in Abbildung 26 exemplarisch dargestellt. Im ersten Schritt werden alle in der Makroperiode möglichen Alternativen hinsichtlich der besten Position zum Einfügen des Loses untersucht. In der Abbildung 26 sind beispielhaft drei verschiedene Positionen dargestellt und bewertet. In dieser variieren die Rüstzeiten in Abhängigkeit von der gewählten Position, da sich durch die Änderung der Einfügeposition auch die Reihenfolge ändert. Zum Schluss wird aus allen möglichen Positionen die günstigste für diese Makroperiode und Produktionslinie gewählt. In diesem Beispiel ist dies die zweite Position. Da die Kapazität des gewählten Schichtmodells (zwei Schichten von jeweils 8 Stunden Länge) zum Einplanen des zusätzlichen Loses nicht ausreicht, wird im zweiten Schritt ein Schichtmodell mit mehr Kapazität ausgewählt. Das ist in diesem Fall ein Schichtmodell mit zwei Schichten und einer Schichtlänge von jeweils 9 Stunden. Zur Vereinfachung wird hier ein identischer Leistungsgrad aller Schichten angenommen und auf die Überprüfung der Pufferkapazitäten verzichtet.

Um die insgesamt über alle Makroperioden und Produktionslinien (der betrachteten Produktionsstufe) günstigste Position zum Einfügen des Loses zu bestimmen, wird dieses Vorgehen für alle erlaubten Makroperioden des Lösungsraums und Produktionslinien wiederholt. Die dann insgesamt günstigste Position wird für das Einfügen des Loses verwendet.

⁶⁸ Für weitere Informationen zur Splittung siehe auch Kapitel 3.2.3.

⁶⁹ Die Laufzeit stünde durch die quadratisch wachsende Anzahl von Aufträgen mit geringem Umfang (bei z.B. vier Splittungen können bis zu 16 neue Aufträge entstehen) nicht in angemessener Relation zur Wahrscheinlichkeit, noch ein gültiges Planungsergebnis zu erzielen.

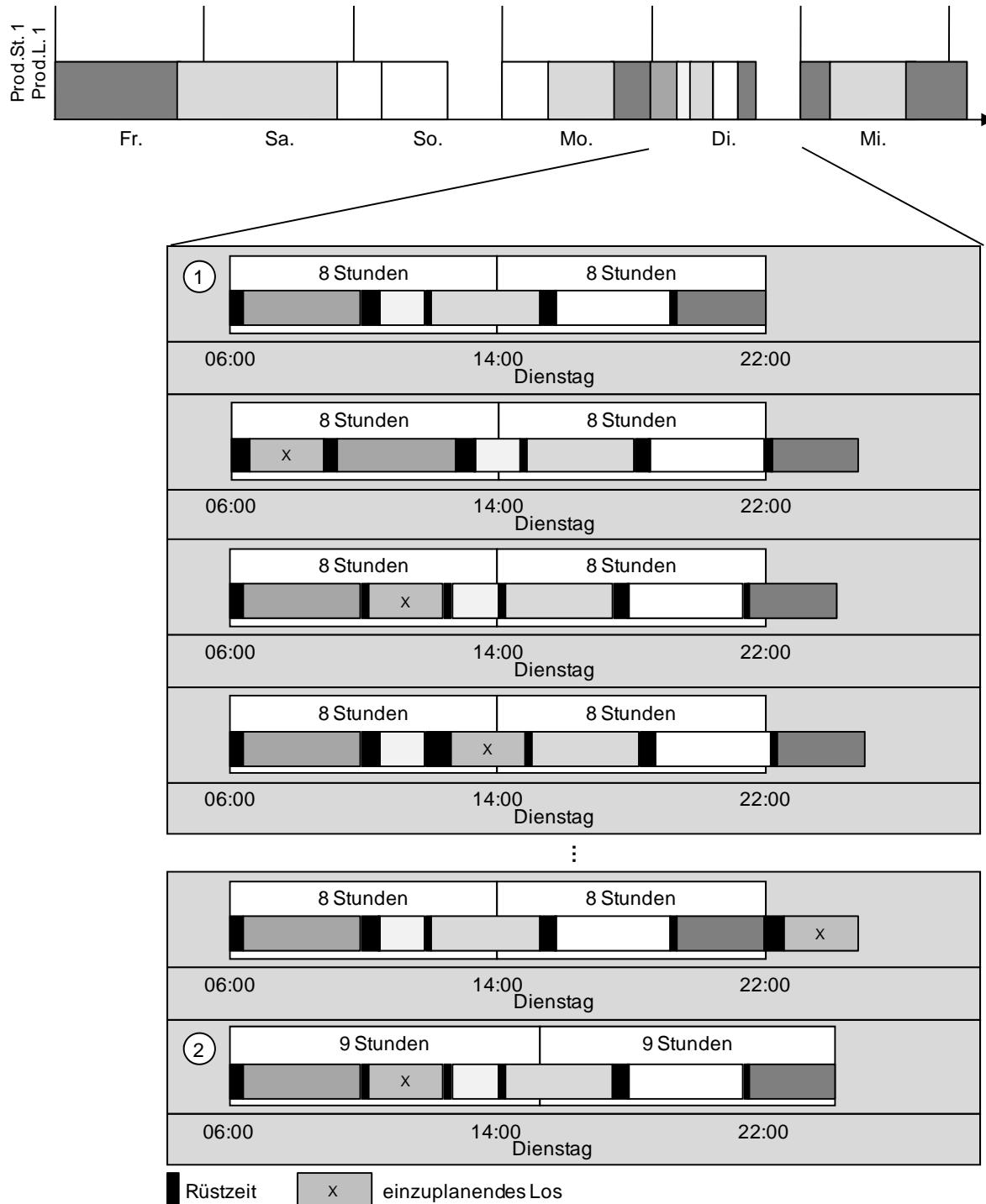


Abbildung 26: Beispiel einer Bedarfserhöhung

5.2.3.1.3 Produktionsstufenabgleich

Selbst bei Ausnutzung der maximalen Kapazität kann der Fall eintreten, dass Termine einiger Bedarfe nicht eingehalten werden können. Handelt es sich hierbei um interne Bedarfe für Erzeugnisse, die zur Weiterverarbeitung benötigt werden, können auf der nachgelagerten Produktionsstufe einige Lose nicht wie geplant beginnen und müssen daher neu eingeplant werden, um die Gültigkeit des Plans wiederherzustellen. Die Abbildung 27 verdeutlicht das Problem. In jeder Iteration des Planungsverfahrens wird jeweils lokal nach der besten Kombina-

tion aus Produktions-, Lager- und Verzugskosten gesucht und nach der Einplanung der gültige Lösungsraum (in dem Beispiel als hellgraue Fläche dargestellt) für die vorgelagerte Produktionsstufe angepasst. Aus diesem Grund wird das Los auf der Produktionsstufe 1 nicht am Samstagmorgen sondern – zur Vermeidung von Nachtzuschlägen – Freitagnachmittags eingeplant. Auf der vorgelagerten Produktionsstufe erfolgt aus gleichen Gründen die Einplanung am Donnerstagnachmittag. Bei der Einplanung des zugehörigen Loses auf der Produktionsstufe 3 ergibt sich nun jedoch ein Problem: Obwohl ursprünglich genügend Kapazität vorhanden war (dargestellt durch den ausgeschlossenen Lösungsraum), reicht die Kapazität auf der Produktionsstufe 3 nicht mehr zur Produktion des Loses aus und der Plan ist ungültig.

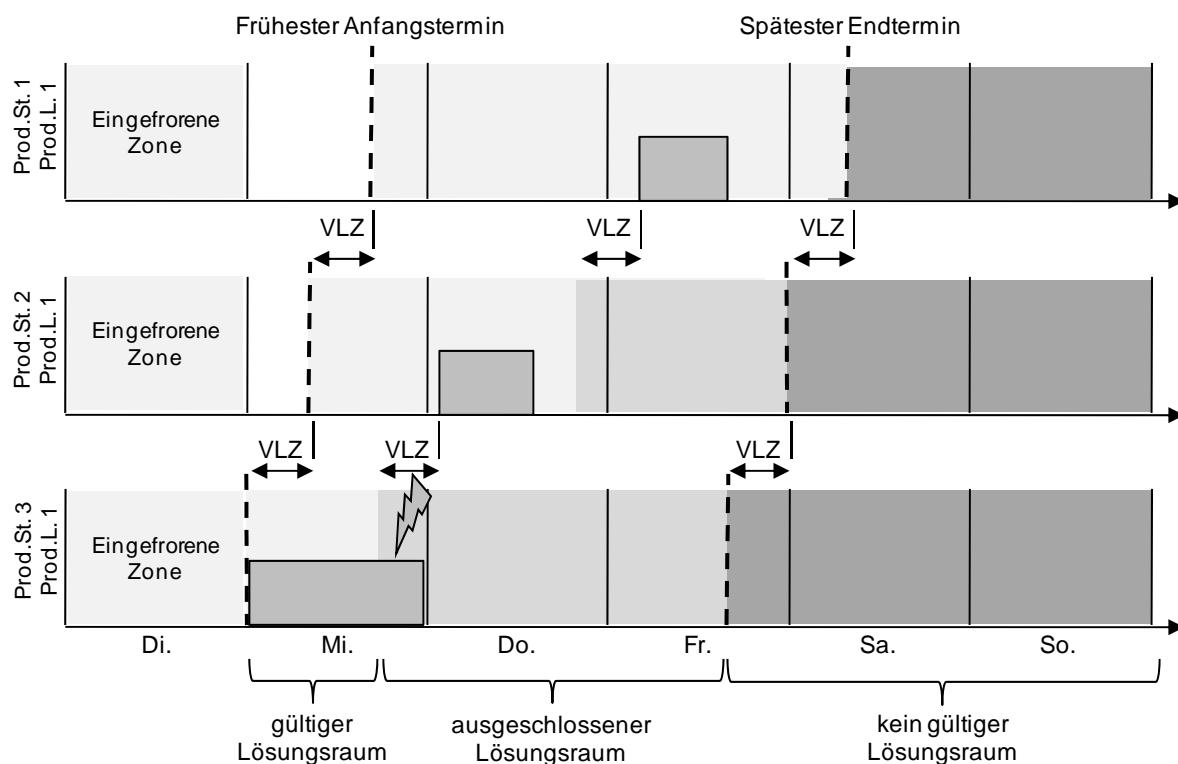


Abbildung 27: Beispiel: Interdependenzen zwischen Produktionsstufen

Zur Lösung dieses Problems dient das in Abbildung 28 dargestellte Vorgehen. Gegenstand dieser Funktion sind sämtliche Lose, die im Rahmen der Rückwärtsplanung nicht korrekt eingeplant werden konnten. Hierzu werden die möglichen Konflikte mit der nachgelagerten Produktionsstufe $ps - 1$ durch den Aufruf der Funktion *KalkLosposition* (siehe Kapitel 5.2.3.1.4.7) mit einer Vorwärtsplanung rekursiv gelöst und der Verzug stufenweise weitergegeben. Sofern die Planung erfolgreich durchgeführt werden konnte und alle Konflikte gelöst sind, wird das Los aus der Liste der Lose mit Verzug entfernt und der konfliktfreie Produktionsplan übernommen. Andernfalls wird der konfliktbehaftete Produktionsplan verworfen, die zugehörige Bedarfserhöhung als ungeplant markiert und die Lose des zugehörigen Auftrags gelöscht. Dieses Vorgehen wird für sämtliche Lose mit Verzug der aktuellen Produktionsstufe durchgeführt.

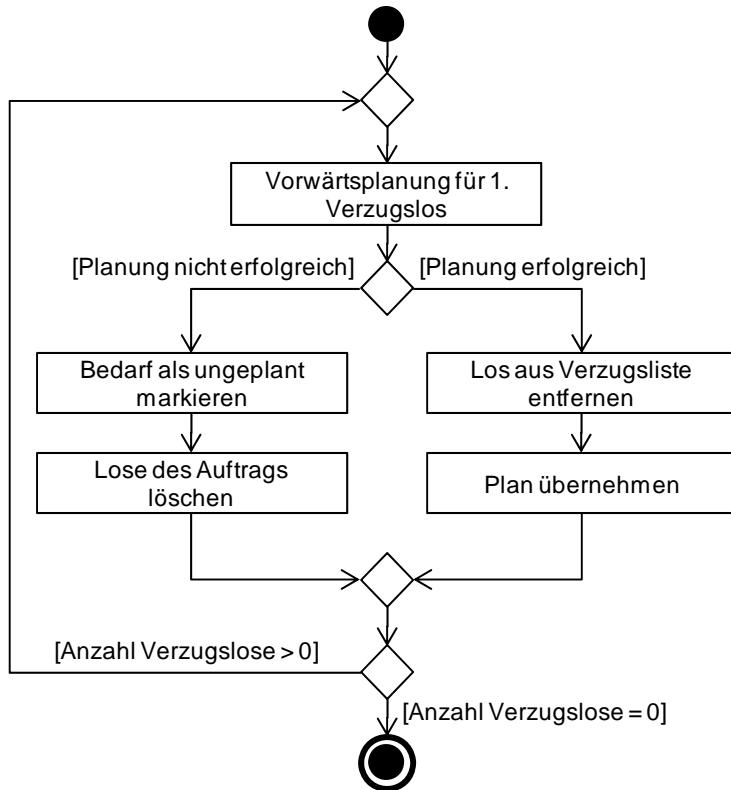


Abbildung 28: Ablauf der Funktion Produktionsstufenabgleich

Zur Verdeutlichung des Produktionsstufenabgleichs wird das bereits oben eingeführte Beispiel fortgeführt (siehe Abbildung 29).

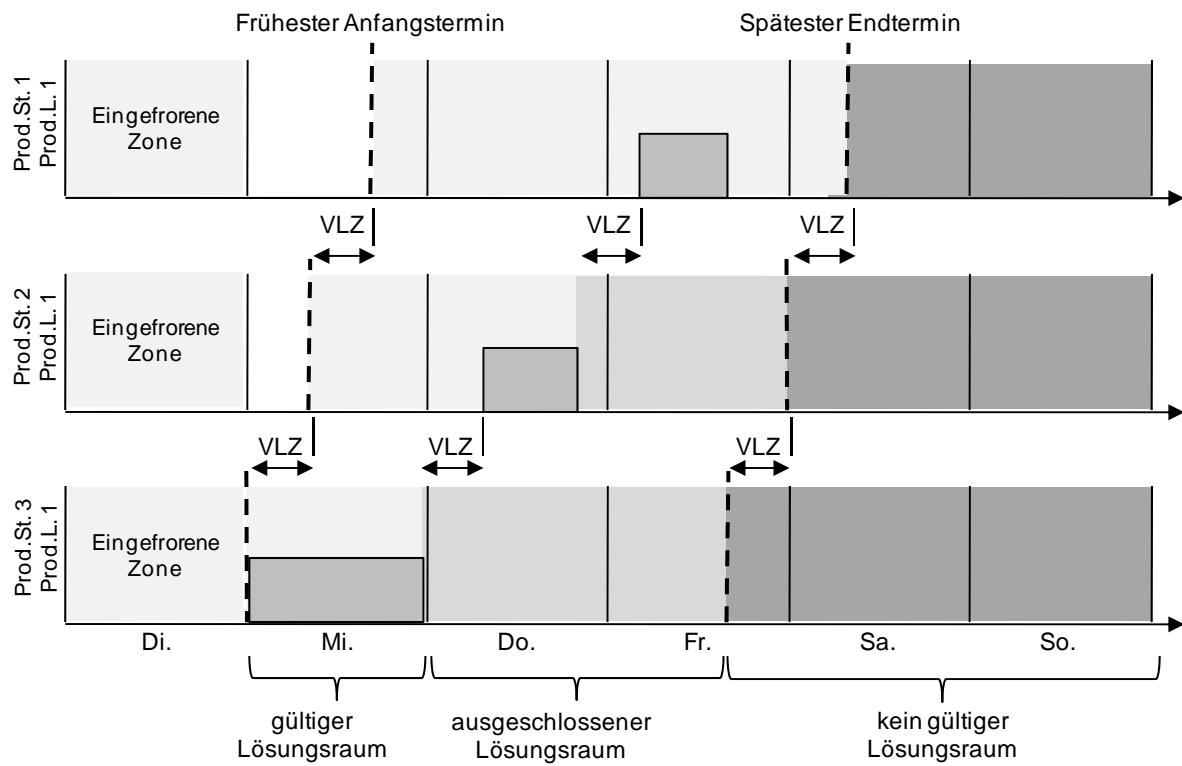


Abbildung 29: Beispiel: aufgelöste Interdependenzen zwischen Produktionsstufen

Aufgrund des frühen Startzeitpunktes des Loses der Produktionsstufe 2 kann das Los auf der Produktionsstufe 3 nicht eingeplant werden. Damit wird das Los der Produktionsstufe 3 zum Konfliktlos. Im Produktionsstufenabgleich wird nun versucht, dieses Konfliktlos ohne Beachtung des Startzeitpunktes der Nachfolgerproduktionsstufe so früh wie möglich einzuplanen. Daraus resultiert eine Inkonsistenz der Startzeit vom Los der Produktionsstufe 2 zu der Endzeit des Loses der Produktionsstufe 3. Also ist nun das Los der Produktionsstufe 2 ein Konfliktlos und wird im Produktionsstufenabgleich behandelt. Auf der Produktionsstufe 2 existiert genügend zeitlicher Spielraum. Daher wird das Los entfernt und ohne neue Inkonsistenzen eingeplant. Damit wäre der Produktionsstufenabgleich erfolgreich beendet. Falls das Los der zweiten Produktionsstufe trotz frühestmöglicher Einplanung zu einer Inkonsistenz zur Produktionsstufe 1 führen würde, ginge der Produktionsstufenabgleich weiter, bis entweder keine Inkonsistenzen mehr auftreten würden oder die Einplanung keinen Erfolg hätte. In diesem Fall würde das Los der Liste nicht einplanbarer Bedarfe zugefügt.

5.2.3.1.4 Hilfsfunktionen

In diesem Kapitel werden die verschiedenen Hilfsfunktionen im Detail vorgestellt.

5.2.3.1.4.1 Funktion: KalkStartzeit

Diese Funktion dient zur Berechnung der neuen Startzeit ($T_{l,pl,ps}^s$) eines Loses und ist erforderlich, wenn ein Los durch eine Bedarfssenkung nicht vollständig gelöscht wird. Aufgrund der möglicherweise unterschiedlichen Leistungsgrade, sind bei der Ermittlung des neuen Startzeitpunktes die ggf. verschiedenen Schichten zu berücksichtigen. Der Ablauf dieser Berechnung ist in Abbildung 30 illustriert. Demnach wird der $T_{l,pl,ps}^s$ dem Endzeitpunkt ($T_{l,pl,ps}^e$) gleichgesetzt. Zur Berechnung wird die in der jeweiligen Schicht vorhandene freie Kapazität ermittelt (dies entspricht der Differenz aus dem Schichtbeginn ($T_{s,pl,ps}^b$) und $T_{l,pl,ps}^s$) und mit der erforderlichen Rüstzeit, unter Beachtung der Leistungsgrade, verrechnet. Reicht dafür die freie Kapazität nicht aus, wird die vorherige Schicht hinzugezogen. Die Funktion terminiert, wenn das Los eingeplant oder das Ende der eingefrorenen Zone ($T_{ps}^{e,eZ}$) erreicht ist.

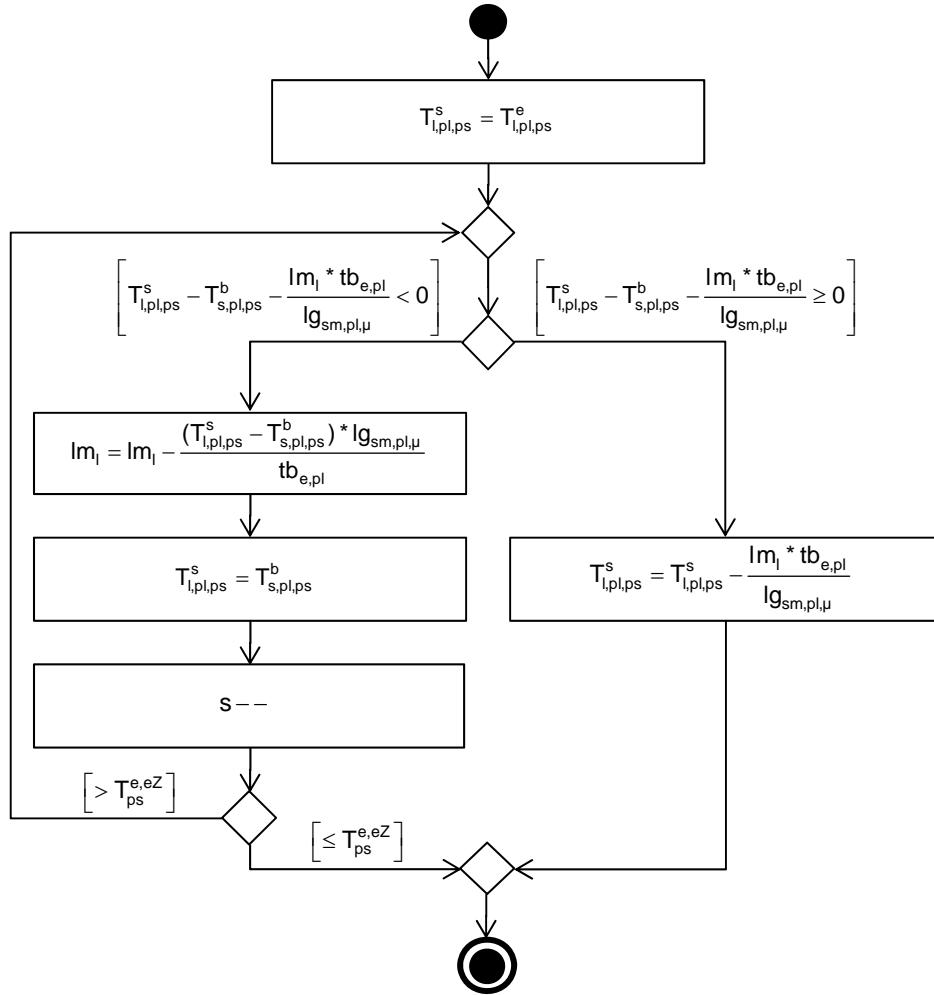


Abbildung 30: Bestimmung des Startzeitpunktes eines Loses

5.2.3.1.4.2 Funktion: DekremSchichtmodell

Das Ziel dieser Funktion besteht in dem Finden eines günstigeren Schichtmodells und wird nach jeder Bedarfssenkung aufgerufen. Wie bereits in Kapitel 2.2.2 beschrieben, sind einer Produktionslinie makroperiodenabhängig mehrere unterschiedliche Schichtmodelle zugeordnet, die entsprechend ihrer Kapazität (abhängig von der Anzahl produktiver Mikroperioden und dem Leistungsgrad) aufsteigend sortiert sind. Zur Wahl eines alternativen Schichtmodells wird nun aus dieser Liste für eine Menge von Makroperioden (sofern diese außerhalb des fixierten Schichthorizontes liegen ($> T_{ps}^{e,fS}$)) iterativ im Rahmen erlaubter Parametrisierung (siehe Kapitel 5.2.2) für die jeweils vielversprechendste Makroperiode ein kleineres Schichtmodell gewählt. Auf dessen Basis erfolgt eine Prüfung (mit der Funktion *KalkLoszeiten*, siehe Kapitel 5.2.3.1.4.3), ob die Lose noch gültig einplanbar sind. Wird ein Fall identifiziert, in der das nicht mehr möglich ist, wird das Schichtmodell des vorherigen Laufs gewählt. Sofern kein alternatives Schichtmodell gewählt werden kann, wird das ursprüngliche beibehalten.

5.2.3.1.4.3 Funktion: KalkLoszeiten

Diese Funktion plant für einen definierten Bereich ($> T_{ps}^{e,eZ}$) und einer Produktionslinie unter Verwendung des zugewiesenen Schichtmodells die zu produzierenden Lose (mit $T_{l,pl,ps}^s \geq T_{s,pl,ps}^b$) frühestmöglich ein. Die Lose liegen zeitlich aufsteigend sortiert vor. Die Abbildung 31 stellt den grundlegenden Ablauf der Methode dar.

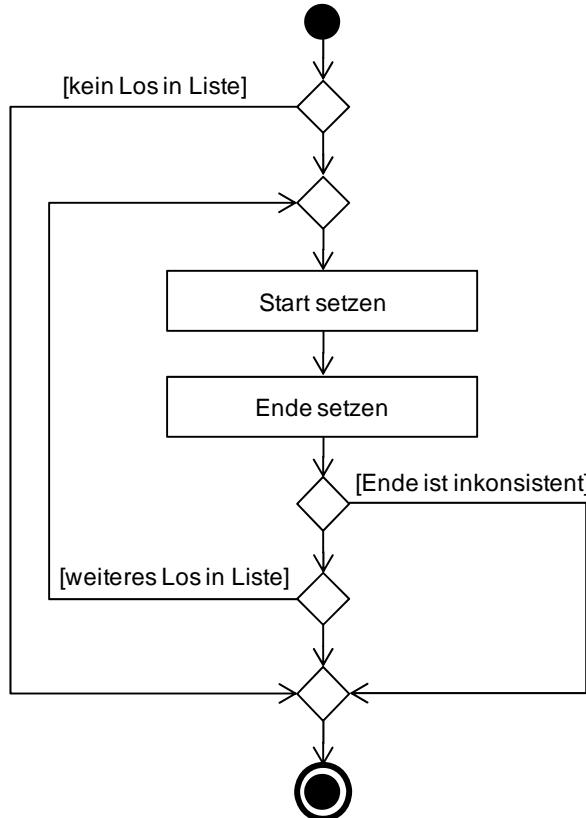


Abbildung 31: Ablauf der Funktion: KalkLoszeiten

Zur Ermittlung dieser Zeiten dienen drei weitere Funktionen. Die Bestimmung der Start- und Endzeitpunkte erfolgt mit den Funktionen *KalkFAZ* (siehe Kapitel 5.2.3.1.4.4) sowie *KalkEndzeit* (siehe Kapitel 5.2.3.1.4.5). Zudem wird analysiert, ob das jeweilige Losende dessen spätesten Endzeitpunkt (*KalkSEZ*, siehe Kapitel 5.2.3.1.4.6) überschreitet (d.h., $T_{l,ps}^{SET} > T_{l,pl,ps}^e$). In diesem Fall ist eine gültige Einplanung nicht möglich.

5.2.3.1.4.4 Funktion: KalkFAZ

Zur Ermittlung des frühesten Anfangszeitpunktes ($T_{l,ps}^{FAT}$) eines Loses dienen die Formeln (5.66) bis (5.69).

$$\alpha := T_{l,pl,ps}^s - p l f_{ps} \quad (5.66)$$

$$\gamma := \max(T_{l,A,ps+1}^e + v l z_e) \quad (5.67)$$

$$\delta := \begin{cases} T_{l-1,pl,ps}^e + rz_{l-1,l,pl}, & \text{wenn } T_{l-1,pl,ps}^e \text{ existiert} \\ T_{ps}^{b,FP}, & \text{sonst} \end{cases} \quad (5.68)$$

$$T_{l,ps}^{FAT} = \begin{cases} \max(\delta, \alpha, T_{ps}^{e,eZ}), & ps = n^{ps} \\ \max(\delta, \alpha, \gamma, T_{ps}^{e,eZ}), & ps < n^{ps} \end{cases} \quad (5.69)$$

Zur Verdeutlichung dient für Los x die Abbildung 32.

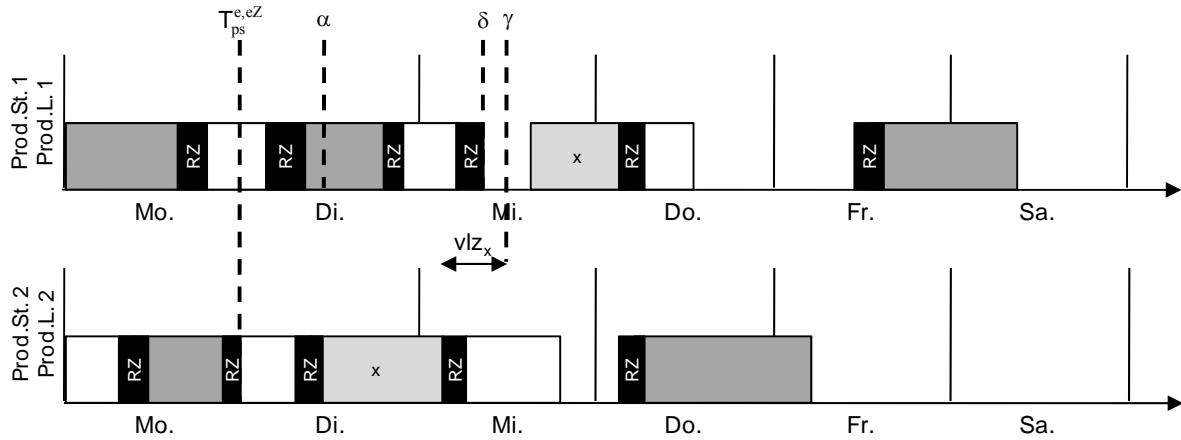
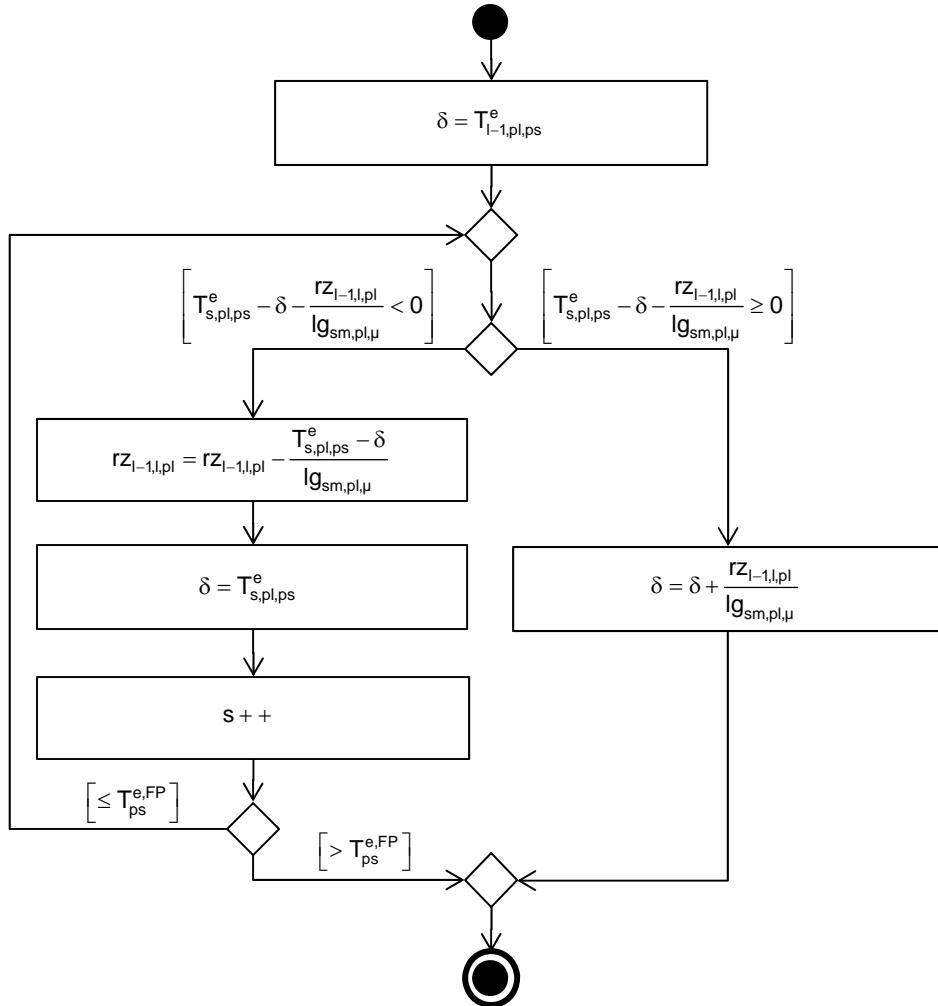


Abbildung 32: Berechnung des frühesten Anfangszeitpunktes

Demnach ist der $T_{l,ps}^{FAT}$ definiert als das Maximum aus der Summe aus der Endzeit des Vorgängerloses ($T_{l-1,pl,ps}^e$, auf der gleichen Produktionslinie) und der Rüstzeit ($rz_{l-1,l,pl}$) sowie des größten Endzeitpunktes der zum Auftrag gehörenden Lose (unter Beachtung der Vorlaufzeit vLz_e) der vorgelagerten Produktionsstufe ($T_{l,A,ps+1}^e$). Zudem müssen der Parameter der maximalen Planungsflexibilität⁷⁰ (plf_{ps}) sowie der Bereich der eingefrorenen Zone ($T_{ps}^{e,eZ}$) berücksichtigt werden. Für die Produktionsstufe n^{ps} existiert offensichtlich keine vorgelagerte Produktionsstufe. Bei der Berechnung von δ müssen auch die unterschiedlichen quantitativen Kapazitätsangebote einer Makroperiode und Produktionslinie berücksichtigt werden. Der Ablauf dieser Berechnung ist in Abbildung 33 illustriert. Demnach wird der δ dem Endzeitpunkt des direkten Vorgängers ($T_{l-1,pl,ps}^e$) gleichgesetzt. Zur Berechnung wird die in der jeweiligen Schicht (s) vorhandene freie Kapazität ermittelt (dies entspricht der Differenz aus dem Schichtende ($T_{s,pl,ps}^e$) und δ) und mit der erforderlichen Rüstzeit, unter Beachtung der Leistungsgrade, verrechnet. Reicht dafür die freie Kapazität nicht aus, wird die nachfolgende Schicht hinzugezogen. Die Funktion terminiert, wenn die Rüstzeit berechnet oder das Ende des Feinplanungshorizontes ($T_{ps}^{e,FP}$) erreicht ist.

⁷⁰ Der Zeitraum, in dem ein Los im Vergleich zu der ursprünglichen Position verschoben werden darf.

Abbildung 33: Ablauf der Berechnung von δ

5.2.3.1.4.5 Funktion: KalkEndzeit

Zur Bestimmung des Losendes wird zu Beginn der Endzeitpunkt ($T_{l,pl,ps}^e$) gleich dem Anfangszeitpunkt des Loses gesetzt. Sofern die in der jeweiligen Schicht für das Los zur Verfügung stehende Kapazität (dies entspricht der Differenz aus dem Endzeitpunkt von dem Los und dem Ende der Schicht ($T_{s,pl,ps}^e$) für die Produktion des Loses ausreicht (die in der entsprechenden Schicht zu belegende Kapazität ergibt sich aus dem Produkt von Losgröße lm_l und Stückbearbeitungszeit $tb_{e,pl}$, dividiert durch den Leistungsgrad der Schicht $lg_{sm,pl,μ}$), wird der Endzeitpunkt aktualisiert und die Funktion endet. Andernfalls wird die Losgröße entsprechend der vorhandenen Kapazität reduziert und die nächste Schicht (s) hinzugezogen. Die Funktion terminiert, wenn das Los vollständig eingeplant oder das Ende des Feinplanungshorizontes ($T_{ps}^{e,FP}$) erreicht ist. Somit erfolgt die Ermittlung des Losendes wie in Abbildung 34 dargestellt.

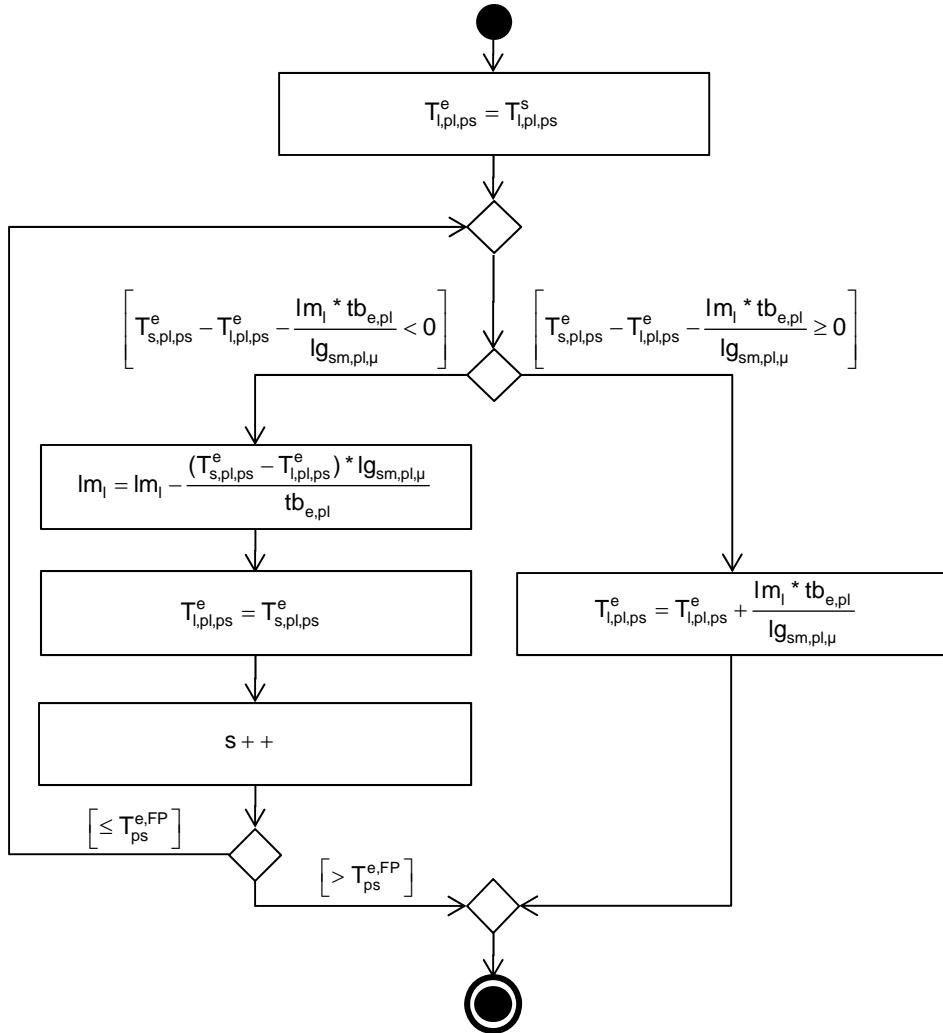


Abbildung 34: Ablauf der Berechnung vom Losende

5.2.3.1.4.6 Funktion: KalkSEZ

Zur Ermittlung des spätesten Endzeitpunktes ($T_{l,ps}^{SET}$) dienen die Formeln (5.70) bis (5.73). Der $T_{l,ps}^{SET}$ ist das Minimum aus der Startzeit des nachfolgenden Loses abzüglich der erforderlichen Rüstzeit sowie des kleinsten Startzeitpunktes der zum Auftrag gehörenden Lose auf der nachgelagerten Produktionsstufe, unter Beachtung der Vorlaufzeit und des Parameters der maximalen Planungsflexibilität. In Abhängigkeit von der Parametrisierung ist zudem – sofern die Einplanung mit Verzug erlaubt ist – das Ende des Feinplanungshorizontes oder der Bedarfsttermin zu berücksichtigen. Für Produktionsstufe 1 ist außerdem zu beachten, dass es keine nachgelagerte Produktionsstufe gibt.⁷¹

⁷¹ Die Berechnung von δ erfolgt analog zu der in Abbildung 33 geschilderten Vorgehensweise. Im Gegensatz zur Ermittlung des $T_{l,ps}^{FAT}$ verläuft die Bestimmung jedoch vom Startzeitpunkt des nachfolgenden Loses $T_{l+1,pl,ps}^s$ rückwärts gerichtet.

$$\alpha := T_{l,pl,ps}^s + plf_{ps} \quad (5.70)$$

$$\gamma := \min(T_{l,A,ps-1}^s - v \cdot l \cdot z_e^v) \quad (5.71)$$

$$\delta := \begin{cases} T_{l+1,pl,ps}^s - rz_{l,l+1,pl}, & \text{wenn } T_{l+1,pl,ps}^s \text{ existiert} \\ T_{ps}^{e,FP}, & \text{sonst} \end{cases} \quad (5.72)$$

$$T_{l,ps}^{SET} := \begin{cases} \min(\delta, \alpha, T_{ps}^{e,FP}), & ps = 1, \text{Verzug} = \text{true} \\ \min(\delta, \alpha, T_{i,ps}), & ps = 1, \text{Verzug} = \text{false} \\ \min(\delta, \alpha, \gamma), & ps > 1 \end{cases} \quad (5.73)$$

5.2.3.1.4.7 Funktion: KalkLosposition

Das Ziel dieser Funktion besteht darin, für das entsprechende Los unter Berücksichtigung produktionslinien- und reihenfolgeabhängiger Rüstzeiten, Puffergrößen und verfügbarer Schichtmodelle die günstigste Einfügeposition und das zum Einfügen benötigte Schichtmodell zu bestimmen. Der Ablauf dieser Funktion hängt von der aufrufenden Funktion ab. Wird diese vom *Produktionsstufenabgleich* (siehe Kapitel 5.2.3.1.3) aufgerufen, wird die Position gemäß dem Prinzip der Vorwärtsplanung bestimmt. Anders verhält sich dies bei einem Aufruf durch die *Bedarfserhöhung* (siehe Kapitel 5.2.3.1.2). Hier folgt die Suche dem Prinzip der Rückwärtsplanung. Dieses Vorgehen liegt darin begründet, dass das Finden einer guten Lösung nahe des Bedarfstermins aufgrund der geringeren Lagerkosten wahrscheinlicher ist als fern des Bedarfstermins. Um diese Wahrscheinlichkeit weiter zu steigern, startet die Suche auf der für das Erzeugnis am besten geeigneten Produktionslinie. Hierzu werden die Produktionslinien nach der Taktzeit sowie darauf aufbauend je gültiger Makroperiode nach belegter Kapazität sortiert.

Im Folgenden wird das Vorgehen dieser Funktion für beide Hauptfunktionen beschrieben. Beim Aufruf der Funktion aus der *Bedarfserhöhung*, d.h., unter Verwendung der Rückwärtsplanung, werden die Bedarfserhöhungen nach absteigendem Bedarfstermin sortiert geplant. Zur Bestimmung der besten Position wird beginnend bei dem jeweiligen Bedarfszeitpunkt, bzw. bei erlaubtem Verzug und der ersten Produktionsstufe vom Ende des Planungshorizontes über die zulässigen Makroperioden in Richtung der eingefrorenen Zone iteriert. In jeder dieser Makroperioden wird unter Verwendung erlaubter Schichtmodelle nach der besten Belegung gesucht. Sofern sich diese Makroperiode außerhalb des fixierten Schichthorizontes ($> T_{ps}^{e,FS}$) befindet, liegen die erlaubten Schichtmodelle nach Kapazität aufsteigend sortiert vor. Da eine Planung der bereits vorhandenen Lose zuzüglich dem neuen Los mit einer kleineren als der aktuell gewählten Kapazität nicht möglich sein wird, kann auf eine Prüfung von Schichtmodellen mit geringerer Kapazität verzichtet werden. Befindet sich die Makroperiode innerhalb des fixierten Schichthorizontes, erfolgt die Suche auf Basis des in einem vorherigen Planungslauf bestimmten Schichtmodells. Zur Bestimmung der Position des neuen Loses wird

der minimale Kapazitätsbedarf für die Einplanung in der jeweilige Makroperiode ermittelt. Diese Position befindet sich an der Stelle, wo durch das Einfügen des Loses die minimalsten Rüstzeiten entstehen. Zur Berechnung dieser dient die Funktion *KalkLosReihenfolgeR*. Nachdem die Reihenfolge der Lose für die aktuelle Produktionslinie bestimmt worden ist, werden die Start- und Endzeitpunkte der Lose in Abhängigkeit des gewählten Schichtmodells mittels Funktion *KalkKapazitätseignungR* berechnet und analysiert. Hierzu werden die Lose in der Rückwärtsplanung ausgehend von dem spätesten Bedarfszeitpunkt durch Left Shifting einplant. Die Position eines Loses ist ungültig, wenn $T_{l,pl,ps}^s \leq T_{ps}^{e,eZ}$ und $T_{l,pl,ps}^e > T_{ps}^{e,FP}$ gilt⁷². Sofern diese Bedingungen nicht verletzt sind sowie $T_{l,pl,ps}^s - vLze > T_{ps}^{e,eZ}$, werden zudem die Bedarfstermine der Vorgänger-Produktionsstufe (ps+1), d.h., $T_{l,pl,ps}^s - vLze$, ermittelt. Da durch die Änderungen von Schichtmodellen die Abhängigkeiten zu den vor- und nachgelagerten Produktionsstufen ungültig werden können, wird zudem die Funktion *KalkKonsistenz* (siehe Kapitel 5.2.3.1.4.8) aufgerufen. Identifiziert diese Funktion eine Inkonsistenz, ist das Einplanen mit dem Schichtmodell nicht möglich. Wurde eine beste Einfügeposition gefunden, wird die Pufferkapazität überprüft. Hierzu dient die Funktion *KalkPufferauslastung* (siehe Kapitel 5.2.3.1.4.9). Bei Gültigkeit werden die Gesamtkosten (Summe aus Produktions-, Lager- und Verzugskosten) der Lösung ermittelt (Funktion *KalkKosten*). Ist die Planung gültig und die bis dahin günstigste Lösung, wird sie als aktuell beste Lösung gespeichert. Falls kein Schichtmodell mit genügend Kapazität zur Einplanung des Loses existiert, gibt es für die Makroperiode und auf der Produktionslinie keine gültige Einfügeposition. Nachdem der beste Plan für die möglichen Schichtmodelle ermittelt worden ist, wird dieses Vorgehen für die nächsten Makroperioden sowie weitere Produktionslinien wiederholt.

Wird die Funktion *KalkLosposition* durch die Funktion *Produktionsstufenabgleich* aufgerufen, werden alle zulässigen Produktionslinien sowie die pro Produktionslinie gültigen Makroperioden vorwärts durchlaufen. Im Gegensatz zur Rückwärtsplanung folgt hierbei die Einplanung der Lose dem Prinzip des Right Shifting. Zur Bestimmung der Lossequenz sowie der Einplanung der sortierten Losliste dienen die Funktionen *KalkLosReihenfolgeV* und *KalkKapazitätseignungV*. Im Anschluss an die Konsistenzprüfung des aktuellen Plans und die Analyse der Pufferauslastung (hierzu dienen wieder die Funktionen *KalkKonsistenz* (siehe Kapitel 5.2.3.1.4.8) und *KalkPufferauslastung* (siehe Kapitel 5.2.3.1.4.9)) wird die insgesamt günstigste Möglichkeit abschließend übernommen sowie sukzessiv über die nachfolgenden Produktionsstufen alle von der Einplanung abhängigen Termine der Planungselemente neu berechnet.

5.2.3.1.4.8 Funktion: KalkKonsistenz

Infolge einer Planänderung kann es zu 4 verschiedenen Konflikten kommen (siehe Abbildung 35).⁷³

⁷² Zur Berechnung dieser Zeiten bedient sich diese Funktion auch wieder den Funktionen *KalkStartzeit* und *KalkEndzeit* (siehe Kapitel 5.2.3.1.4.1 bzw. 5.2.3.1.4.5).

⁷³ Es ist zu beachten, dass dies keine mögliche Situation ist, sondern nur sämtliche möglichen Konflikte kompakt darstellt.

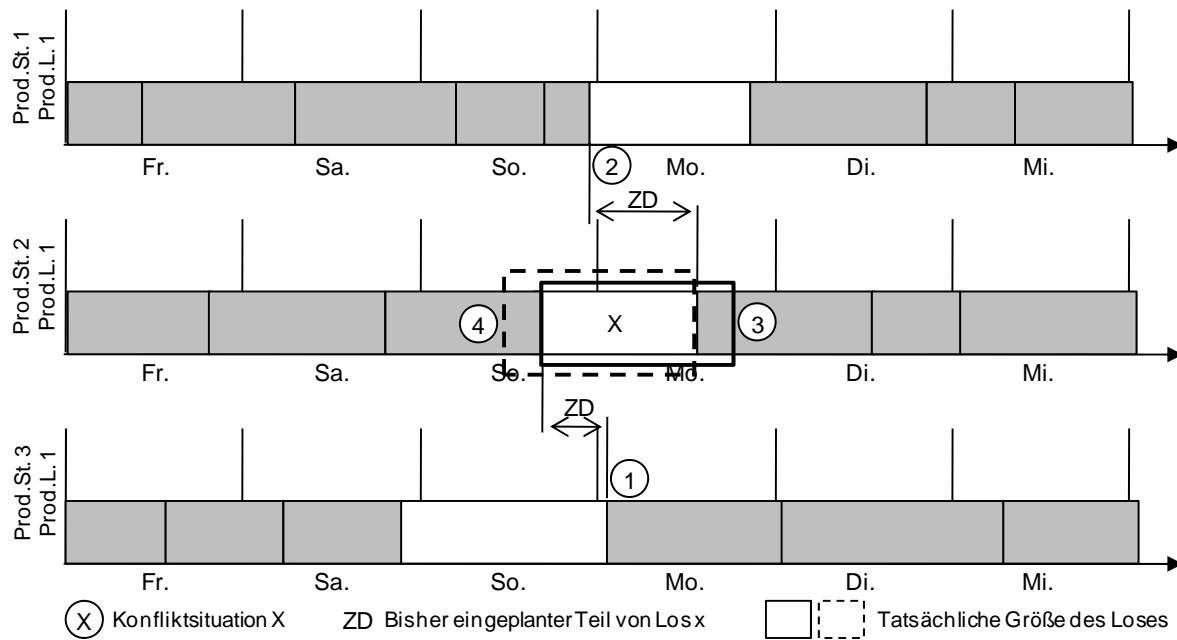


Abbildung 35: Übersicht über Inkonsistenzen der mehrstufigen Planung

Der erste Fall beschreibt die Situation, in der ein Los x (auf der zweiten Produktionsstufe) bzgl. seiner ursprünglichen Position früher eingeplant wurde. Der Startzeitpunkt von x liegt dabei vor dessen frühestem Anfangszeitpunkt, was zu einer negativen Zeitdifferenz (ZD_{VS}) führt. Die in der Abbildung 35 als zweiter Fall dargestellte Situation beschreibt die Einplanung des Loses x zu einem späteren Zeitpunkt. Dieser liegt nach dessen spätestmöglichen Endzeitpunkt, was auch eine negative Zeitdifferenz (ZD_{NS}), verursacht. Der dritte und vierte Fall beschreiben das Vorliegen einer Überschneidung des Loses x mit dem nachfolgenden bzw. vorherigen Los auf der gleichen Produktionslinie. Auch hier ergeben sich negative Zeitdifferenzen (ZD_{NL} bzw. ZD_{VL}) und zwar aus dem Endzeitpunkt des Loses x und dem Startzeitpunkt des nachfolgenden Loses bzw. dem Endzeitpunkt des vorherigen Loses und dem Startzeitpunkt des Loses x (siehe Formeln (5.74)-(5.77)). Wenn ZD (Formel (5.78)) einen negativen Wert annimmt, liegt eine Inkonsistenz vor.

$$ZD_{VS} = T_{x,pl,ps}^s - T_{x,pl,ps}^{FAT} \quad (5.74)$$

$$ZD_{NS} = T_{x,pl,ps}^{SET} - T_{x,pl,ps}^e \quad (5.75)$$

$$ZD_{NL} = T_{x,pl,ps}^e - T_{x+1,pl,ps}^s \quad (5.76)$$

$$ZD_{VL} = T_{x-1,pl,ps}^e - T_{x,pl,ps}^s \quad (5.77)$$

$$ZD = \min (ZD_{VS}, ZD_{NS}, ZD_{NL}, ZD_{VL}) \quad (5.78)$$

5.2.3.1.4.9 Funktion: KalkPufferauslastung

Sind die Pufferkapazitäten knapp bemessen, kann der Fall eintreten, dass durch eine ungünstige Kombination gewählter Schichtmodelle und entsprechender Bedarfstermine zu viel gelagert werden muss. Soll z.B. eine bestimmte für Montag notwendige Menge mit voller Kapazitätsauslastung von Mittwoch bis Freitag produziert werden – um z.B. die teureren Wochenendschichten zu sparen – kann es passieren, dass die kurzfristigen Pufferkapazitäten nicht ausreichen, um die Menge zwischen zu lagern. Hierzu könnte es erst erforderlich sein, dass die geplanten Bedarfe für Donnerstag und Freitag vom Lager abgehen.

Zur Überprüfung (siehe Abbildung 36) wird zunächst die Ausgangsbelegung der betroffenen Puffer zu Beginn des Planungshorizontes bestimmt. Anschließend werden alle für die Pufferbelegung relevanten Vorgänge chronologisch betrachtet. Um die Einhaltung der Pufferkapazitäten zu überprüfen, ist über diese Vorgänge zu iterieren und bei jedem Los ein entsprechender Zugang zum zugehörigen Puffer zu addieren sowie bei einem Bedarf ein entsprechender Abgang vom Puffer zu subtrahieren.

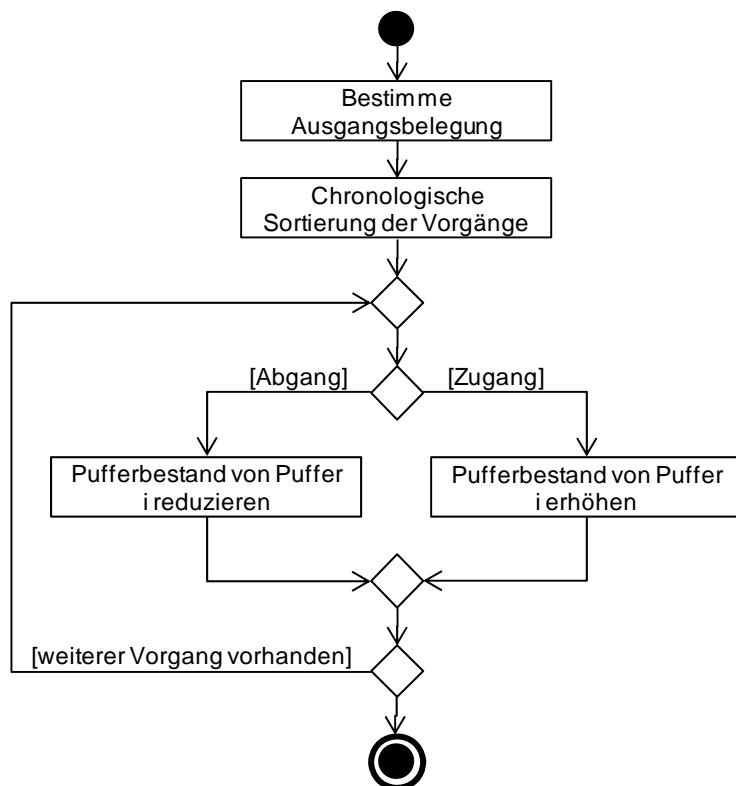


Abbildung 36: Ablauf der Funktion: KalkPufferauslastung

Die Kapazität, die belegt oder freigegeben wird, ergibt sich dabei aus dem Produkt der Losgröße bzw. Bedarfsmenge und einem Kapazitätsfaktor pro Stück des Erzeugnisses (siehe Formel (5.79)).

$$p_{f_{pu}} = \begin{cases} p_{f_{pu}} + lm_{i,ps} * kf_e, & \text{wenn Zugang} \\ p_{f_{pu}} - lm_{i,ps} * kf_e, & \text{wenn Abgang} \end{cases} \quad (5.79)$$

Dieser Faktor ist zwingend erforderlich, um Aspekte wie die unterschiedliche Größe, die Art der Lagerung und weitere Faktoren mit in die Berechnung eingehen zu lassen. Ist nach einem Zugang die maximale Kapazität überschritten, bricht die Überprüfung ab. Werden alle Vorgänge durchlaufen, so war die Validierung erfolgreich.

5.2.3.2 Verbesserungsverfahren

Im vorherigen Kapitel wurde das Eröffnungsverfahren im Detail dargestellt. Aufgrund der großen Komplexität des Planungsproblems ist jedoch das Finden einer bestmöglichen Lösung nicht in einem Schritt möglich. Aus diesem Grund wird dem Eröffnungsverfahren ein Verbesserungsverfahren nachgeschaltet. In diesem Verfahren findet eine weitere der in Kapitel 3.2.3 beschriebenen Maßnahmen zur Änderungsplanung Anwendung. So ist das Ziel, durch eine Änderung der Ausführungsreihenfolge und der Schichtmodellzuweisung den Ursprungsplan (generiert durch das Eröffnungsverfahren) zu verbessern.

In Kapitel 3.2.2 wurden die unterschiedlichen Typen von Verbesserungsverfahren diskutiert. Es wurde deutlich, dass eine generelle ordinale Ordnung bzgl. der Qualität der generierten Lösung, in Abhängigkeit des Verfahrens, nicht möglich ist. Die nachbarschaftsbasierten Varianten nehmen allerdings pro Verbesserungsschritt tendenziell die geringsten Änderungen vor. Da dies der Forderung einer möglichst hohen Planstabilität entspricht sowie die Verfahren einer intuitiven Vorgehensweise folgen⁷⁴, werden in dieser Arbeit das Simulated Annealing und die Tabu Search angewandt. Der Einsatz dieser beiden Verfahren ist von den gewählten Parametern abhängig. Wie bereits in Kapitel 3 beschrieben, eignet sich das Simulated Annealing eher bei guten Ausgangslösungen und einer kurzen Rechenzeit. Dagegen erlaubt die Tabu Search ein schnelleres Verlassen von lokalen Optima und ist daher eher bei einem hohen Änderungsaufkommen geeignet. Werden beide Verfahren parallel zur Lösungsfindung genutzt, wird nach einem Durchlauf jeweils das hinsichtlich des Zielkriteriums beste Ergebnis gewählt.

Nachfolgend wird das Verbesserungsverfahren⁷⁵ schrittweise vorgestellt. Den Ablauf des Verfahrens und damit auch die wesentlichen Schritte verdeutlicht Abbildung 37. Da für die Güte der Planungsergebnisse und der Laufzeit der Verfahren besonders die Art der Nachbarschaft relevant ist, liegt nachfolgend der Fokus auf der Phase der Nachbarschaftserzeugung.

⁷⁴ Nach Kurbel (vgl. [Kurb05]) ist es für die Akzeptanz der Planungsergebnisse durch den Planer relevant, dass die Ergebnisfindung nachvollziehbar ist.

⁷⁵ Aufgrund der großen prinzipiellen Ähnlichkeit der beiden Verfahren – sie differenzieren sich nur durch das Abtasten des Lösungsraums – werden im weiteren Verlauf dieser Arbeit unter dem Begriff des Verbesserungsverfahrens beide Verfahren subsumiert.

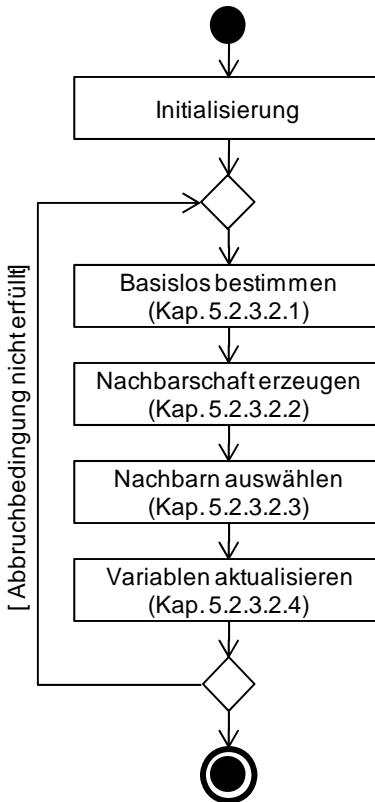


Abbildung 37: Gesamtablauf des Verbesserungsverfahrens

Analog zu dem Eröffnungsverfahren startet das Verbesserungsverfahren⁷⁶ mit der Initialisierung. In dieser Phase erfolgen Berechnungen, die nicht die eigentliche Verbesserung betreffen, sondern eine Grundlage für diese bilden. In Abhängigkeit des konkreten Verbesserungsverfahrens umfasst dies die Initialtemperatur (für das Simulated Annealing), die Länge der Tabu-Liste⁷⁷ (für die Tabu Search), die Initiallösung sowie die Abbruchbedingung. Zur Berechnung der Initiallösung dienen die Funktionen *InitSchichtmodell* und *InitLose*, die die gewählten Schichtmodelle des Planungshorizontes in Abhängigkeit der Produktionslinie sowie die zu jedem Los gehörende Startperiode des Ausgangsplans ermitteln. Nach Abschluss der Initialisierung beginnt die eigentliche Verbesserung. Diese setzt sich aus vier Schritten zusammen. Zu Beginn erfolgt die Bestimmung des Loses mit dem größten Potential (Kapitel 5.2.3.2.1) sowie dessen gültige Nachbarschaft (Kapitel 5.2.3.2.2).⁷⁸ Aus dieser Menge erfolgt im dritten Schritt die Bestimmung des in der jeweiligen Situation besten Nachbarn. Diese Wahl ist abhängig von den entsprechenden Vorgaben (Kapitel 5.2.3.2.3). Im Anschluss daran werden alle Variablen aktualisiert (Kapitel 5.2.3.2.4). Diese Schritte werden solange durchgeführt, bis die Abbruchbedingung erfüllt ist. Diese setzt sich aus der maximalen Rechenzeit und der maximalen Anzahl an aufeinanderfolgenden Iterationen, in denen kein neues Optimum gefunden wird, zusammen. Analog zu dem Eröffnungsverfahren kommen auch im Ver-

⁷⁶ Für eine detaillierte Beschreibung der Abläufe des Simulated Annealing und Tabu Search siehe Abbildung 12 und Abbildung 13.

⁷⁷ In dieser Arbeit wird eine statische Tabu-Liste verwendet, deren Länge im Rahmen der Parametrisierung (siehe Kapitel 5.2.2) zu bestimmen ist.

⁷⁸ Die Berechnung erfolgt in den Funktionen *KalkBasislos* sowie *KalkNachbarn*. (siehe Anhang A)

besserungsverfahren verschiedene Hilfsfunktionen zum Einsatz. Diese werden in Kapitel 5.2.3.2.5 beschrieben.

5.2.3.2.1 Basislos bestimmen

Diese Phase der Nachbarschaftsgenerierung dient der Komplexitätsreduktion. Anstatt alle existierenden Lose zu betrachten, besteht das Ziel in der Bestimmung eines Loses – dem Basislos – mit einem besonders großen Verbesserungspotential. Dieses ergibt sich dabei aus den beiden Faktoren der Kosten und der Flexibilität. Es wird die Idee zugrunde gelegt, dass die Wahrscheinlichkeit für die Generierung eines guten Nachbarn steigt, wenn das Los mit den höchsten Kosten pro Stück an einer anderen Position eingeplant wird. Eine Übersicht über den grundsätzlichen Ablauf gibt Abbildung 38. Zur Identifikation des Loses mit dem maximalen Potential bedarf es der Kenntnis über die detaillierten (potentiellen) Kosten eines Loses. Berechnet werden diese Kosten mit der Funktion *InitKosten* (siehe Kapitel 5.2.3.2.5.1).

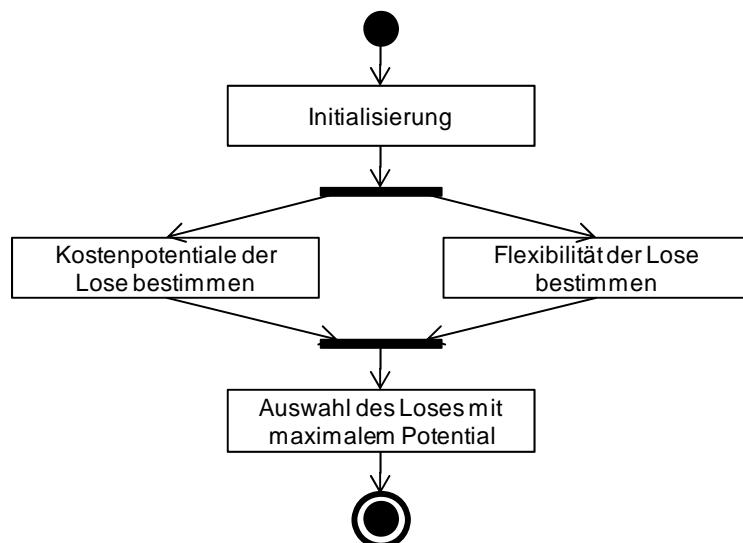


Abbildung 38: Ablauf der Basislosbestimmung

Die ausschließliche Kostenbetrachtung bei der Auswahl des Basisloses ist jedoch nicht ausreichend, da das kostenmaximale Los ggf. an keiner anderen Position eingeplant werden kann. Daher wird im Rahmen der Bestimmung des Loses mit maximalem Potential zusätzlich die Flexibilität bzgl. der zeitlichen Einplanung des jeweiligen Loses berücksichtigt. Da zu diesem Zeitpunkt noch nicht bestimmt wurde, welches Los an welcher Stelle neu eingeplant werden soll, ist die Flexibilität für alle Lose zu berechnen. In dem Verbesserungsverfahren werden diese Berechnungen mit der Funktion *InitFlexibilität* (siehe Kapitel 5.2.3.2.5.2) durchgeführt.

Nachdem die Kosten der Lose und die Flexibilität aller Produktionsstufen ermittelt wurden, ist es durch die gleichzeitige Betrachtung (Summation) der beiden Faktoren möglich, das Los mit maximalem Verbesserungspotential zu bestimmen. Hierzu erfolgt eine Normierung der beiden Werte auf eine einheitliche Skala. Dazu werden die Ausprägungen der Flexibilität mit dem Quotienten q (siehe Formel (5.80)) multipliziert:

$$q = \frac{\maxLosKosten_{plan}}{\maxPSFlexibilität_{plan}} \quad (5.80)$$

Um das vielversprechendste Los zu identifizieren, reicht jedoch eine isolierte Analyse einzelner Lose nicht aus. Dies ist dadurch begründet, dass ggf. die auf einer Produktionsstufe teurere Produktion eines Loses auf einer anderen Produktionsstufe durch eine dort günstigere Lage ausgeglichen werden kann. Um nun das Los mit dem größten Verbesserungspotential zu identifizieren, ist es notwendig, den Auftrag zu ermitteln, der über das gesamte Produktionssystem die höchsten Kosten verursacht. Dazu werden für jeden Auftrag die Kosten der zugehörigen Lose summiert. Auf dieser Basis kann der ungünstigste Auftrag A ermittelt werden. Das Basislos entspricht nun aus dieser Menge dem Los, das den maximalen Wert bzgl. seiner Kombination aus normierten Kosten sowie der Flexibilität der zugehörigen Produktionsstufe aufweist.

In der Abbildung 39 ist die Basislosermittlung dargestellt. Es wird deutlich, dass bei hoher Flexibilität ein Basislos mit relativ geringen Kosten gewählt werden kann. So wird auf der Produktionsstufe 1 das hellgraue Los (mit Kosten (K) von 90 GE) dem zugehörigen Los (mit Kosten von 110 GE) der zweiten Produktionsstufe vorgezogen. Dadurch wird die Wahl eines Loses vermieden, das nur mit geringer Erfolgswahrscheinlichkeit an einer anderen Stelle eingeplant werden kann. Des Weiteren kann sich die Bewertung des Loses mit den höheren Kosten durch die Auswahl des vermeintlich ungünstigeren Basisloses in den nachfolgenden Iterationen erhöhen, so dass es zu einem späteren Zeitpunkt als Basislos gewählt wird und dann eine höhere Erfolgswahrscheinlichkeit aufweist. Bei der Bestimmung des Basisloses ist zu beachten, dass Lose innerhalb der eingefrorenen Zone ($T_{ps}^{e,eZ}$) ausgeschlossen werden.

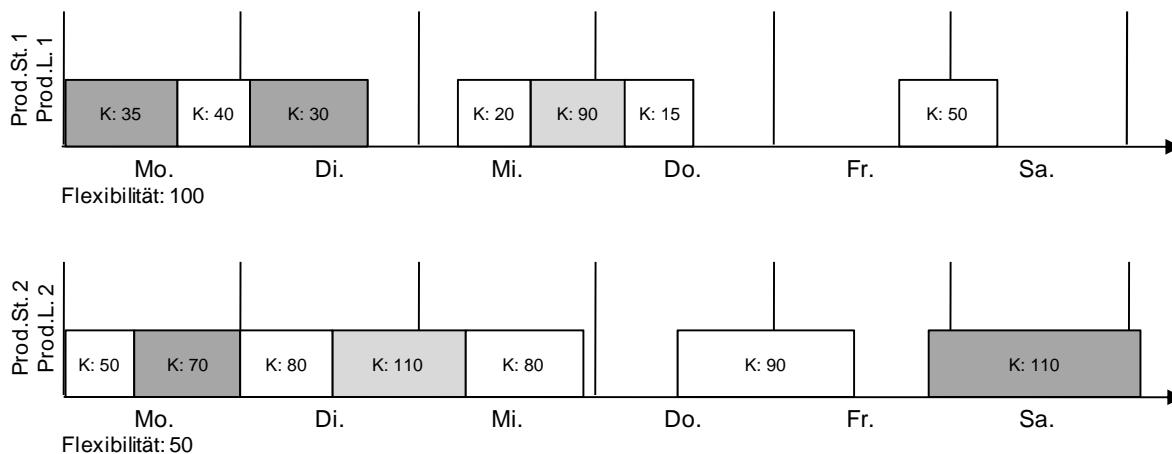


Abbildung 39: Basislos-Flexibilitätsberechnung

5.2.3.2.2 Nachbarschaft erzeugen

Im Stand der Technik wurde aufgezeigt, dass die Bildung der Nachbarschaft, d.h., die Generierung der möglichen Nachbarlösungen aus einer Basislösung, das zentrale Element des Simulated Annealing und der Tabu Search darstellt und für beide Verfahren identisch ist. Die Qualität dieser Nachbarschaftsgenerierung definiert maßgeblich die Güte des gesamten Verbesserungsverfahrens. Die Güte einer Nachbarschaft wird wiederum über deren Effizienz bestimmt, d.h., über die Anzahl der Elemente, die die Nachbarschaftsgenerierung jeder Basislösung

sung zuordnet, sowie über die Auswahl dieser Elemente. Das Ziel besteht somit in der Generierung möglichst weniger Nachbarn, die aber bzgl. der Zielsetzung besonders gut erscheinen. Die folgenden Unterkapitel beschreiben im Detail, wie diesen Anforderungen Rechnung getragen wird. Der Ablauf der Nachbarschaftsgenerierung ist in Abbildung 40 dargestellt.

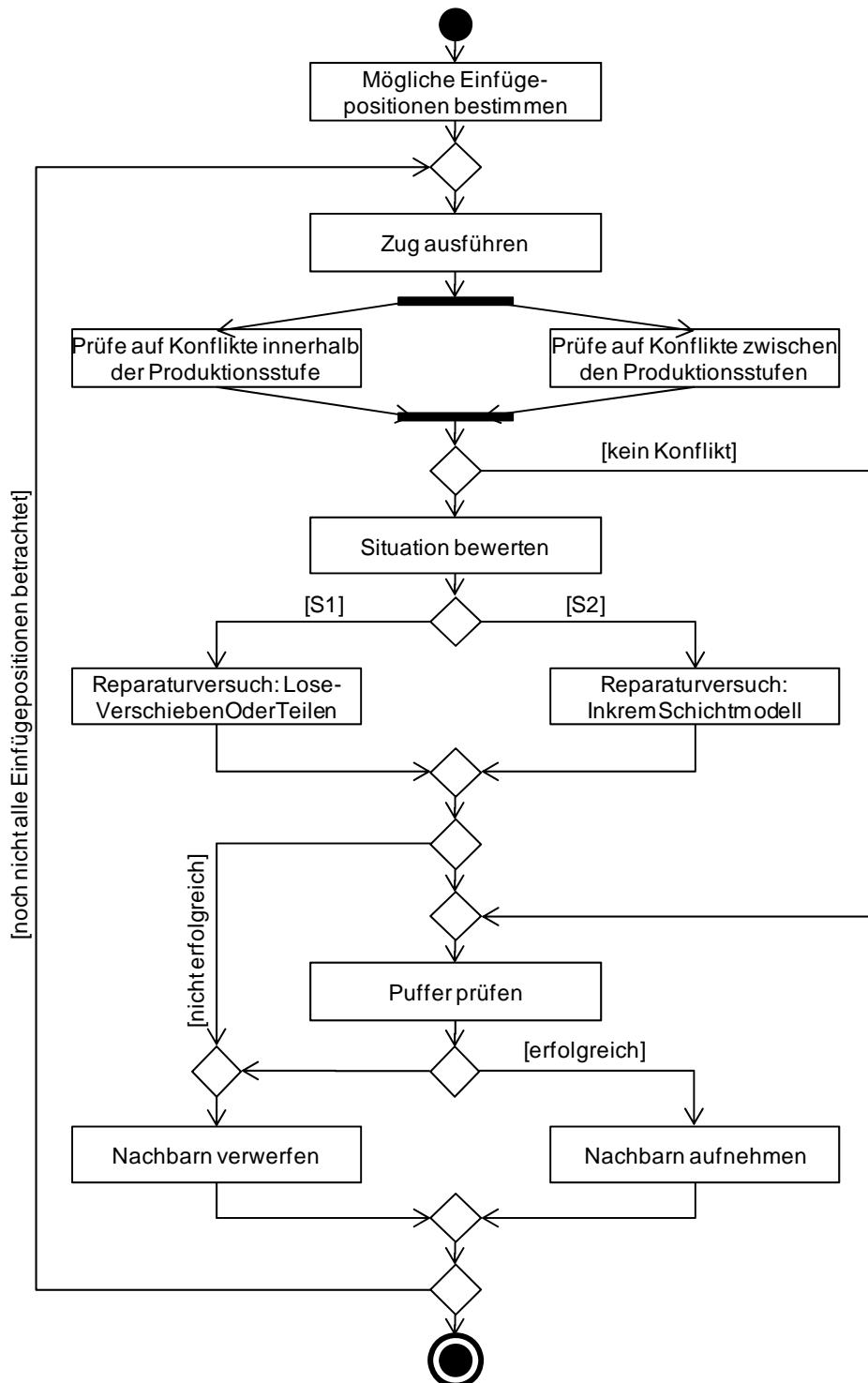


Abbildung 40: Ablauf der Nachbarschaftsgenerierung

Bevor Planänderungen⁷⁹ erfolgen können, sind zunächst sinnvolle Positionen zur Einplanung des Basisloses zu bestimmen. Hierzu dient die Funktion *KalkAlternativpositionen* (siehe Kapitel 5.2.3.2.5.3). Auf Basis dieser sind anschließend die Planänderungen durchzuführen. Hierbei wird jeweils in einer Kopie des Ursprungsplans hinter einer solchen Position das Basislos eingefügt und das resultierende Losende, unter Beachtung der entsprechenden Rüstzeit, berechnet. Hierzu dient die Funktion *KalkEndzeit* (siehe Kapitel 5.2.3.1.4.5). Zuerst werden von einem vorgegebenen Starttermin aus die resultierenden Rüstzeiten (die Berechnung erfolgt ebenfalls mit der Formel (5.65)) ermittelt. Dem schließt sich die Berechnung der Endzeit an. Hierbei ist jedoch zu beachten, dass an dieser Stelle keine Überprüfung hinsichtlich ausreichender Kapazitäten vorgenommen wird, so dass Inkonsistenzen auftreten können. Würden diese temporären Inkonsistenzen verboten, so würde der Lösungsraum unnötig stark eingeschränkt. Die Überprüfung hinsichtlich der Inkonsistenzen sowie ggf. die Auflösung dieser erfolgt in einem späteren Schritt (siehe Kapitel 5.2.3.2.5.4).

Zur Änderung der Ausführungsreihenfolge wird den im Stand der Technik (siehe Kapitel 3.2.2.4) beschriebenen Erkenntnissen gefolgt. Demnach werden anstelle von Swap-Zügen (also dem direkten Tauschen von Elementen) Insert-Züge verwandt. Um einen solchen Zug auszuführen, sind drei Teilschritte notwendig:

- Los entfernen: Das Basislos wird aus dem Plan entfernt. Diese Aufgabe übernimmt die Funktion *EntferneLos*.
- Schichtmodell reduzieren: Durch den ersten Schritt wurde Kapazität freigegeben. Um ungenutzte Kapazität zu vermeiden, ist für die Makroperioden, in denen das Basislos eingeplant war, das Schichtmodell zu reduzieren. Hierzu dient die Funktion *DekremSchichtmodell* (siehe Kapitel 5.2.3.1.4.2).
- Los einfügen: Das zuvor entfernte Basislos ist an einer anderen gültigen Position neu einzuplanen. Dies erfolgt mittels der Funktion *KalkEndzeit* (siehe Kapitel 5.2.3.1.4.5). Des Weiteren ist der durch das Basislos erzeugte Bedarf auf der vorgelagerten Produktionsstufe entsprechend zu aktualisieren.

Im Anschluss daran ist für die aktuelle Lösung die Gültigkeit zu überprüfen, bevor sie der Menge der Nachbarn hinzugefügt werden darf. Hierzu dient wieder die Funktion *KalkKonsistenz* (siehe Kapitel 5.2.3.1.4.8). Nimmt in dieser Funktion *ZD* einen negativen Wert an, so zeigt dies das Vorhandensein einer Inkonsistenz. Diese führt jedoch im Gegensatz zum Eröffnungsverfahren nicht zwangsläufig zum Ausschluss dieser Position. So ist die Berücksichtigung von Einfügepositionen, die vor dem $T_{x,ps}^{FAT}$, nach dem $T_{x,ps}^{SET}$ oder innerhalb eines zu kleinen Leerraums liegen, sinnvoll, da mit Hilfe geeigneter Reparaturmethoden aus diesen Plänen kostenminimale sowie gültige Lösungen entstehen können.

Hat sich eine solche Inkonsistenz ergeben, erfolgt zunächst die Bewertung der Situation sowie anschließend die Bestimmung einer geeigneten Reparaturmethode. Diese Funktionalität übernimmt *SpezifischeReparatur* (siehe Kapitel 5.2.3.2.5.4), wobei Reparaturmaßnahmen grundsätzlich auf die betroffene Produktionsstufe beschränkt werden. Sofern keine Inkonsistenz

⁷⁹ Analog zu dem Eröffnungsverfahren werden Planänderungen auf Basis von Kopien des Ursprungsplans vorgenommen.

aufgetreten ist, wird anschließend die *KalkPufferauslastung* (siehe Kapitel 5.2.3.1.4.9) gestartet. Diese prüft, ob die beschränkten Pufferkapazitäten weiterhin eingehalten werden und entspricht der des Eröffnungsverfahrens. Da eine Veränderung der Produktionszeit jedoch Implikationen auf vor- und nachgelagerte Puffer haben kann, wird diese Funktion zweifach aufgerufen. Ist die Überprüfung erfolgreich, so kann der Plan der Menge den gültigen Nachbarlösungen hinzugefügt werden.

5.2.3.2.3 Wahl eines Nachbarn

Nach Abschluss der Nachbarschaftsberechnung liegt eine Menge von gültigen Nachbarn vor. Das jeweilige Verbesserungsverfahren beschreibt, unter welchen Bedingungen ein Element dieser Menge auszuwählen ist und wie der Auswahlprozess verläuft. Sowohl für die Tabu Search als auch für das Simulated Annealing wurden dazu die in der Literatur bekannten und gut erprobten Standardverfahrensweisen übernommen. Eine detaillierte Beschreibung dieser wurde bereits in Kapitel 3.2.2.2 bzw. Kapitel 3.2.2.3 vorgenommen.

5.2.3.2.4 Aktualisierung der Variablen

Nachdem eine gültige Nachbarlösung ausgewählt wurde, erfolgt die Aktualisierung der Variablen. Dies sind die Variablen der aktuellen Lösung (x), der bisher besten gefundenen Lösung (x^*) sowie die Zählvariablen der jeweiligen Metaheuristik. Auf dieser Basis arbeitet die nächste Iteration des Verbesserungsverfahrens. Wird jedoch festgestellt, dass die Abbruchbedingung erfüllt ist, terminiert das Verfahren und liefert das beste gefundene Ergebnis zurück.

5.2.3.2.5 Hilfsfunktionen

In diesem Kapitel werden die verschiedenen Hilfsfunktionen im Detail vorgestellt.

5.2.3.2.5.1 Funktion: InitKosten

Analog zu dem Eröffnungsverfahren werden Produktionskosten, Lagerkosten und Verzugskosten ermittelt. Die Produktionskosten ergeben sich für ein Los l aus den anteiligen Kosten der Schichtmodelle, die den Makroperioden zugewiesen sind, in denen l liegt. Die für l benötigte Zeit umfasst dessen Bearbeitungsdauer und die ggf. vorhandene Rüstzeit. Die Lagerkosten entsprechen dem Produkt der Lagerzeit, der Losgröße und dem Lagerkostenfaktor des Erzeugnistyps. Lagerkosten entstehen erst dann, wenn ein Los vollständig produziert wurde. Analog dazu entstehen die Verzugskosten als Produkt der Verzugszeit, der Losgröße und den Strafkosten je Zeit des Erzeugnistyps. Der Ablauf der Kostenermittlung ist in Abbildung 41 dargestellt.

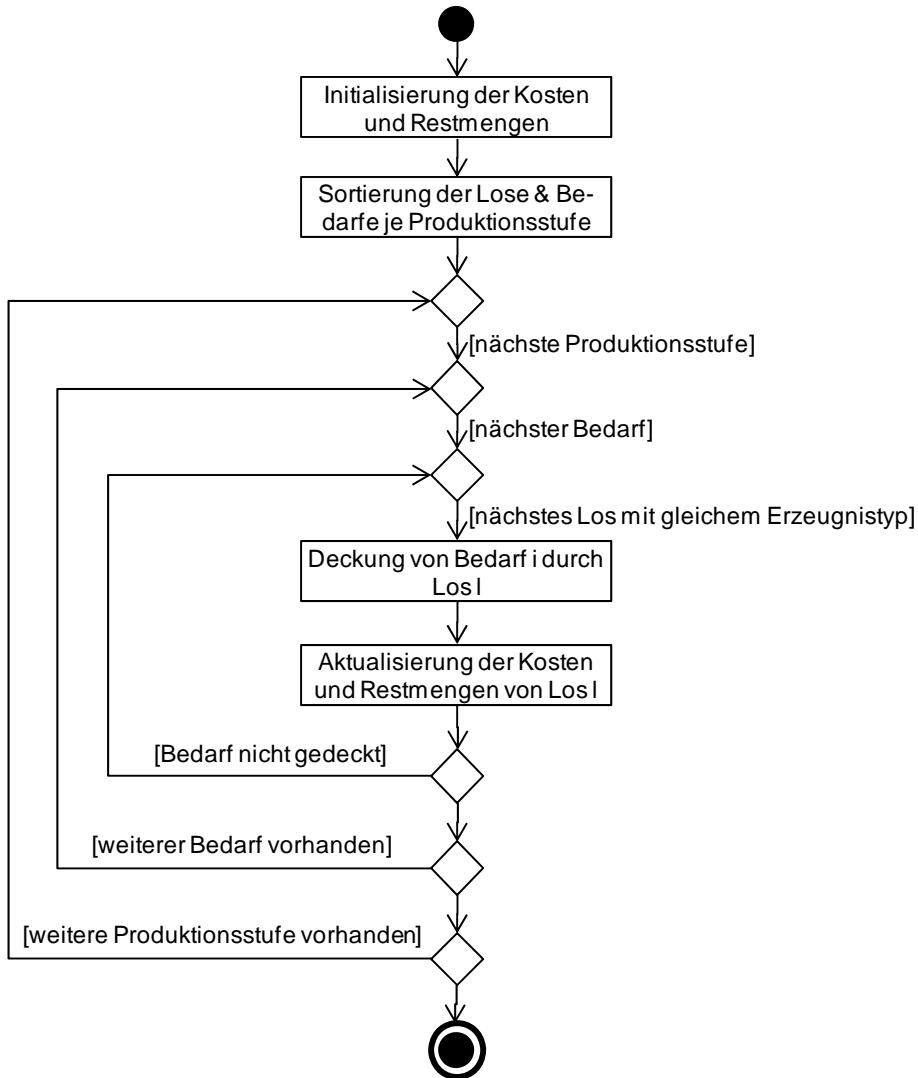


Abbildung 41: Berechnung losspezifischer Kosten

Die Kosten aller Lose sind initial auf 0 gesetzt. Des Weiteren wird zu jedem Los eine Restmenge ($rest_l$) ermittelt, wobei diese zu Beginn der Losgröße entspricht. Unter der Annahme, dass sowohl die Bedarfe als auch die Lose in Abhängigkeit der Produktionsstufe aufsteigend sortiert vorliegen, wird die Vorwärtsrechnung durch sukzessive Deckung der Bedarfe von Produktionsstufe n^{ps} bis Produktionsstufe 1 durchgeführt. Dazu werden für jeden Bedarf so lange Lose desselben Erzeugnistyps durchlaufen und deren Restmengen aktualisiert, bis der Bedarf vollständig gedeckt wurde. Dabei werden die Kosten für jedes Los l , das an der Deckung des aktuellen Bedarfs beteiligt ist, entsprechend der Zeitdifferenz ($zeitDiff$) zwischen Bedarfsdatum und Enddatum sowie der Mengendifferenz ($mengDiff$) aus Restbedarf und Losrest nach Formeln (5.81) bis (5.84) aktualisiert.

	$mengDiff \geq 0$	$mengDiff < 0$
$zeitDiff \geq 0$ (Lagerkosten)	$lk_l = lk_l + zeitDiff * lk_e * rest_l$ (5.81)	$lk_l = lk_l + zeitDiff * lk_e * (rest_l + mengDiff)$ (5.82)
$zeitDiff < 0$ (Verzugskosten)	$vk_l = vk_l - zeitDiff * vk_e * rest_l$ (5.83)	$vk_l = vk_l - zeitDiff * vk_e * (rest_l + mengDiff)$ (5.84)

Tabelle 2: Kostensituationen bei der Vorwärtsrechnung

5.2.3.2.5.2 Funktion: InitFlexibilität

Die Funktion hat die Bestimmung der Flexibilität des Produktionsplans zum Ziel. Da zu diesem Zeitpunkt jedoch noch nicht bekannt ist, welches Los an welcher Stelle neu eingeplant werden soll, wird die Flexibilität für alle Lose ermittelt. Zur Verringerung des Rechenaufwands wird die Flexibilität grob durch drei Indikatoren für jede Produktionsstufe beschrieben. Das reicht aus, um die meisten Situationen zu vermeiden, in denen ein kostenmaximales Los als Basislos bestimmt wird, obwohl es nicht umgeplant werden kann. Zur Berechnung dieser Indikatoren werden analog zu dem Eröffnungsverfahren die Informationen über den frühestmöglichen Anfangszeitpunkt ($T_{l,ps}^{FAT}$) und den spätest möglichen Endzeitpunkt ($T_{l,ps}^{SET}$) der Lose genutzt (siehe Kapitel 5.2.3.1.4.4 und 5.2.3.1.4.6). Im Gegensatz zum Eröffnungsverfahren werden die Formeln (5.68) und (5.72) ignoriert. Dies ist dadurch begründet, dass an dieser Stelle Kenntnis über die produktionsstufenbezogene grobe Flexibilität ausreicht.

Auf Basis des frühesten Anfangszeitpunktes und spätesten Endzeitpunktes können die folgenden Indikatoren für jede Produktionsstufe für jedes Los berechnet werden:

- Flexibilität in der Einplanung: Die Formel (5.85) gibt die Summe der Mikroperioden an, um die die Lose früher oder später eingeplant werden könnten. Die Berechnung erfolgt durch die Iteration über die Lose aller Produktionslinien der Produktionsstufe (PL_{ps}) und der gleichzeitigen Summierung der Abweichung vom frühestmöglichen Anfangszeitpunkt sowie vom spätestmöglichen Endzeitpunkt des aktuellen Loses.

$$flex_{ps}^{einp} = \sum_{l=1}^{|L|} \sum_{pl=1}^{|PL_{ps}|} T_{l,pl,ps}^s - T_{l,ps}^{FAT} + T_{l,ps}^{SET} - T_{l,pl,ps}^e \quad (5.85)$$

- Kapazitätserweiterung: Die Formel (5.86) gibt die maximal mögliche Erweiterung der Kapazität einer Produktionsstufe innerhalb des Planungshorizontes an (vom Ende des fixierten Schichthorizontes ($T_{ps}^{e,fS}$) bis zum Ende der Feinplanung ($T_{ps}^{e,FP}$)). Die Berechnung erfolgt durch eine Iteration über alle Produktionslinien innerhalb des Planungshorizontes und der gleichzeitigen Summierung der Differenz aus aktueller Kapazität und maximaler Kapazität.

$$k_{ps}^{diff} = \sum_{pl=1}^{|PL_{ps}|} \sum_{t=T_{ps}^{e,FS}+1}^{T_{ps}^{e,FP}} k_{pl,t}^{max} - k_{pl,t} \quad (5.86)$$

- Ungenutzte Kapazität: Dieser Indikator gibt die freie Kapazität der Produktionsstufe bei gegebenen Schichtmodellen an. Zur Berechnung wird über die Lose aller Produktionslinien der Produktionsstufe iteriert und jeweils die Differenz aus Endzeit des nachfolgenden Loses und Startzeit des aktuellen Loses summiert (Formel (5.87)). Für das erste und letzte Los je Produktionslinie existiert kein Vorgänger resp. Nachfolger und daher gilt der Beginn bzw. das Ende des Planungshorizontes als Referenz.

$$LO_{ps}^{sum} = \sum_{pl=1}^{|PL_{ps}|} \left((T_{1,pl,ps}^s - T_{ps}^{s,FS}) + (T_{ps}^{e,FP} - T_{L,pl,ps}^s) \right) + \sum_{l=PL_{ps}+1}^{|L-PL_{ps}|} \sum_{pl=1}^{|PL_{ps}|} T_{l,pl,ps}^e - T_{l+1,pl,ps}^s \quad (5.87)$$

Die Summe dieser drei einzelnen Faktoren (Formel (5.88)) ergibt den Indikator der Flexibilität ($flex_{ps}$) der jeweiligen Produktionsstufe.

$$flex_{ps} = flex_{ps}^{einp} + k_{ps}^{diff} + LO_{ps}^{sum} \quad (5.88)$$

5.2.3.2.5.3 Funktion: KalkAlternativpositionen

Aufgabe dieser Funktion ist die Bestimmung gültiger Einfügepositionen auf Basis des ermittelten Basisloses. Als gültige Einfügepositionen werden die Endzeitpunkte der Lose definiert. Dadurch ist es möglich, die Größe der Nachbarschaft gering zu halten, ohne dabei die erfolgversprechendsten Kandidaten auszuschließen. Dies sind Einfügepositionen, bei denen für einen zeitlich begrenzten nachfolgenden Bereich ungenutzte Kapazität (Leerraum) vorliegt. Dort ist die Wahrscheinlichkeit am größten, dass das Basislos unter geringem Änderungsaufwand eingeplant werden kann. Dabei gilt es allerdings, die Länge der eingefrorenen Zone ($> T_{ps}^{e,eZ}$) sowie die maximale Planungsflexibilität (plf_{ps}) zu beachten. Mit der beschriebenen Basislosbestimmung sowie dieser Nachbarschaftsgenerierung wird der Rechenaufwand somit stark reduziert. Zur Veranschaulichung dient Abbildung 42.

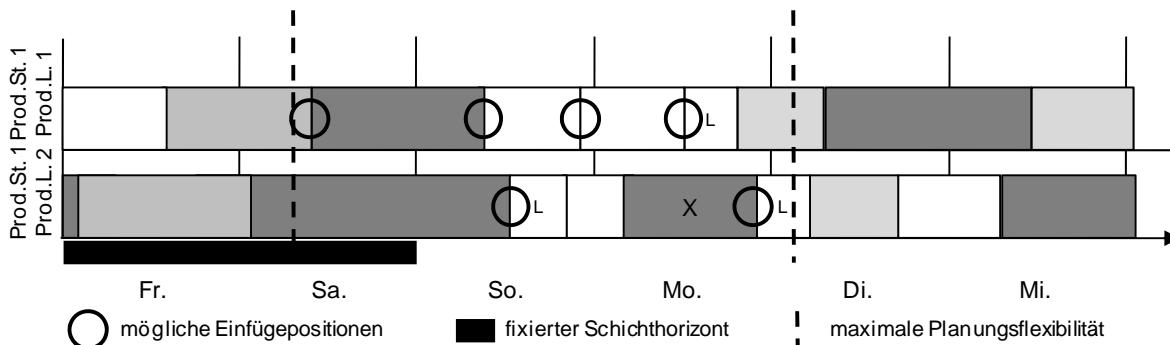


Abbildung 42: Alternative Startzeitpunkte eines Loses

5.2.3.2.5.4 Funktion: SpezifischeReparatur

Beim Aufruf dieser Funktion wurde das Basislos eingeplant, so dass die Flexibilität auf dieser Informationsbasis präziser berechnet werden kann. Dazu werden die bereits vorgestellten Indikatoren der Flexibilität in der Einplanung ($flex_{ps}^{einp}$), der maximalen Kapazitätsausweitung (k_{ps}^{diff}) und der ungenutzten Kapazität (LO_{ps}^{sum}) neu bestimmt. Bei der Berechnung von $flex_{ps}^{einp}$ und LO_{ps}^{sum} sind nur die Lose bzw. Leerräume der entsprechenden Produktionslinie relevant (also nicht der ganzen Produktionsstufe), auf der das Basislos eingeplant wurde. Bei der vorherigen Berechnung wurden zudem die auf der jeweiligen Produktionslinie vorhandenen Lose ignoriert. Relevant war ausschließlich die Flexibilität der ganzen Produktionsstufe. An dieser Stelle ist jedoch die Flexibilität auf einer Produktionslinie relevant, so dass die existierenden Vorgänger- bzw. Nachfolgerlose berücksichtigt werden müssen. Hierzu werden bei der Berechnung der frühesten Anfangs- und spätesten Endtermine auch die Formeln (5.68) und (5.72) genutzt.

Die Berechnung von k_{ps}^{diff} wird auf die Makroperioden beschränkt, in denen das Basislos eingeplant wurde. Ist die Anzahl an Makroperioden geringer als drei, so werden zur Erhöhung der Flexibilität die vorherige und die nachfolgende Makroperiode ebenfalls berücksichtigt. Zur Herstellung eines gültigen Plans stehen zwei Reparaturmethoden *LoseVerschiebenOderTeilen* (siehe Kapitel 5.2.3.2.5.5) und *InkremSchichtmodell* (siehe Kapitel 5.2.3.2.5.6) zur Verfügung. Die Funktion *LoseVerschiebenOderTeilen* wird ausgelöst, wenn $flex_{ps}^{einp} + LO_{ps}^{sum} > k_{ps}^{diff}$ ([S1] in Abbildung 40). Überwiegt dagegen der Indikator der maximalen Kapazitätsausweitung ([S2] in Abbildung 40), wird die Reparaturmethode *InkremSchichtmodell* aufgerufen.

5.2.3.2.5.5 Funktion: LoseVerschiebenOderTeilen

Diese Reparaturfunktion greift, wenn die durch die Schichtmodelle zugeordnete Kapazität nicht voll genutzt wird (es bestehen also Lücken im Plan) und gleichzeitig die Lose über eine große Flexibilität verfügen. In diesem Fall versucht diese Funktion in einem maximal dreistufigen Vorgehen die Konsistenz herzustellen. Zur weiteren Illustration dieses Vorgehens dient die Abbildung 43.

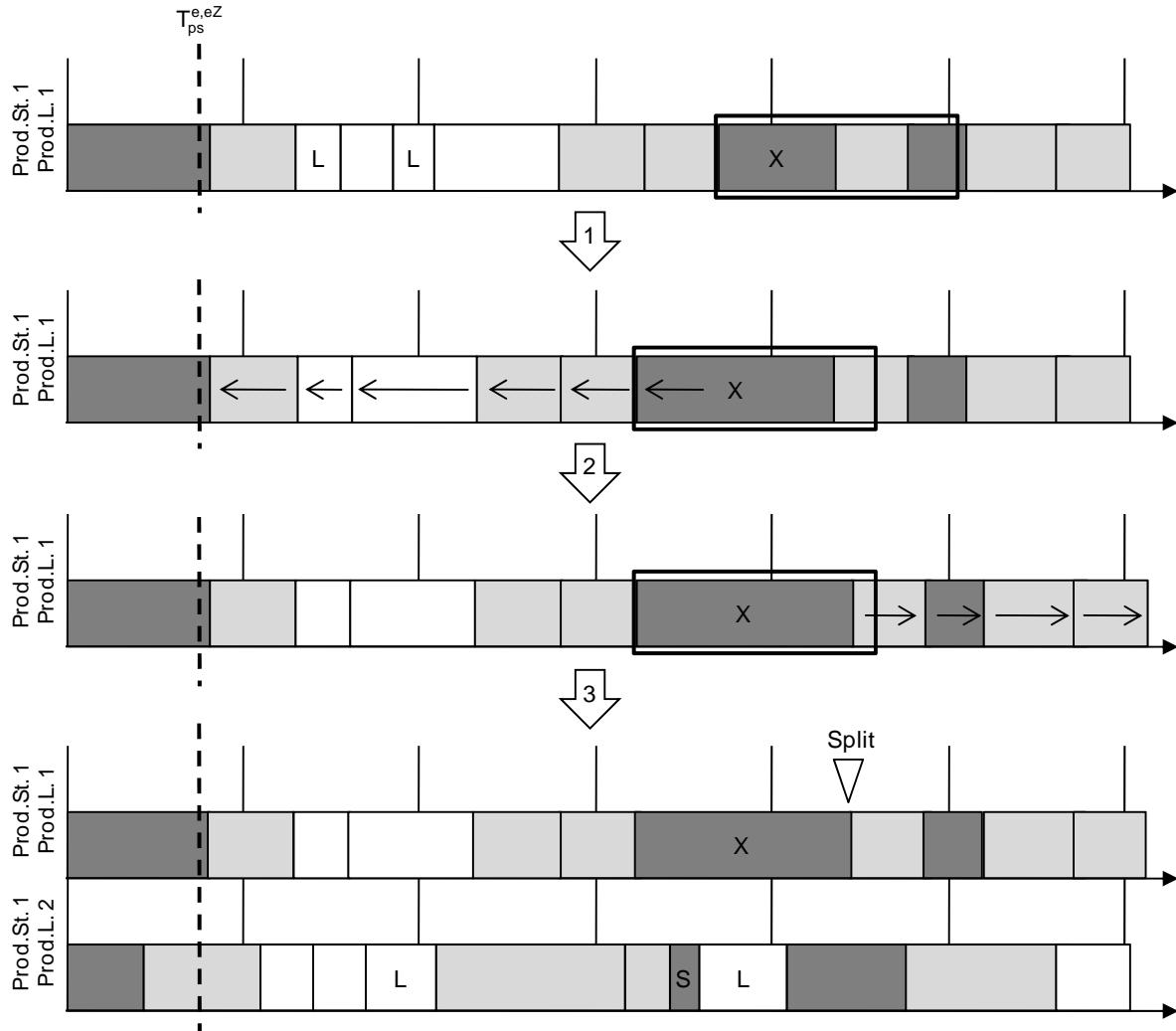


Abbildung 43: Ablauf der Funktion: LoseVerschiebenOderTeilen

Die erste Zeile stellt die Ausgangssituation des Beispiels dar. In dieser soll ein Los x in einen Leerraum eingeplant werden, der jedoch nicht über ausreichend Kapazität verfügt. Aus diesem Grund resultiert eine Überschneidung zu dessen Nachfolger (die eigentliche Länge von Los x wird durch das schwarzes Rechteck dargestellt). Im ersten Schritt sind nach der eingefrorenen Zone ($T_{ps}^{e,eZ}$) alle Lose der inkonsistenten Produktionslinie bis einschließlich Los x durch Left Shifting frühestmöglich einzuplanen sowie die Bedarfszeitpunkte entsprechend zu aktualisieren. Zur Bestimmung der Anfangs- und Endzeiten der Lose dienen wieder die Funktionen *KalkFAZ* (siehe Kapitel 5.2.3.1.4.4) sowie *KalkEndzeit* (siehe Kapitel 5.2.3.1.4.5). Anschließend wird geprüft, ob der frühestmögliche Anfangszeitpunkt von Los x soweit vorgezogen werden konnte, dass die vollständige Bearbeitung vor dem spätestmöglichen Endzeitpunkt abgeschlossen wird ($T_{x,pl,ps}^e \leq T_{x,ps}^{SEZ}$). Ist dies nicht der Fall, erfolgt im zweiten Schritt die spätestmögliche Einplanung (Right Shifting) sämtlicher Nachfolger von Los x sowie die Aktualisierung der entsprechenden Bedarfszeitpunkte. Begonnen wird mit dem letzten Los im Planungshorizont, um dem Los x die maximale Menge an vorhandener freier Kapazität zur Verfügung zu stellen. Danach erfolgt erneut die Prüfung, ob der spätestmögliche Endzeitpunkt von Los x eingehalten wird. Wurde abermals keine Konsistenz erreicht, startet der dritt-

te Schritt. In diesem wird der Problemteil (dieser sollte durch die beiden vorherigen Schritte bereits kleiner geworden sein) von dem Los x abgesplittet. Im Anschluss daran wird versucht, für dieses abgesplittete Los S eine parallele Produktionslinie zu finden, dessen qualitative und quantitative Kapazität geeignet ist, um das Los innerhalb seines frühestmöglichen Anfangszeitpunktes sowie spätestmöglichen Endzeitpunktes ohne zusätzliche Planänderungen zu bearbeiten. Zur Berechnung dieser Zeitpunkte dienen wieder die Funktionen *KalkFAZ* (Kapitel 5.2.3.1.4.4) sowie *KalkSEZ* (Kapitel 5.2.3.1.4.6). Analog zu dem vorherigen Vorgehen werden die Produktionslinien, die qualitativ zur Produktion des Splitloses S geeignet sind, aufsteigend nach der quantitativen Eignung sortiert. Entsprechend dieser Reihenfolge werden anschließend iterativ diese Produktionslinien hinsichtlich einer erfolgreichen Einplanung des Splitloses durch Ausnutzung freier Kapazitäten untersucht. Bei erfolgreicher Einplanung ist der durch das Originallos erzeugte Bedarf der vorgelagerten Produktionsstufe zu reduzieren und ein weiterer Bedarf in Höhe der Splitlosgröße zu erstellen. Sofern keine konsistente Einplanung möglich ist, wird der bisher potentielle Nachbar verworfen, da ohne tiefgreifende Planänderungen keine Gültigkeit erreicht werden kann. Umfangreiche Änderungen werden ausgeschlossen, da sie dem Prinzip der Nachbarschaft widersprechen. Diese Vorgehensweise ergibt sich analog dazu bei einer Inkonsistenz mit der vor- oder nachgelagerten Produktionsstufe.

5.2.3.2.5.6 Funktion: *InkremSchichtmodell*

Im Eröffnungsverfahren erfolgte bereits eine detaillierte Erklärung des Ablaufs der Funktion *DekremSchichtmodell* (siehe Kapitel 5.2.3.1.4.2). Die Funktion *InkremSchichtmodell* arbeitet generell nach dem gleichen Vorgehen. Zur Veranschaulichung dient das nachfolgende Beispiel (Abbildung 44).

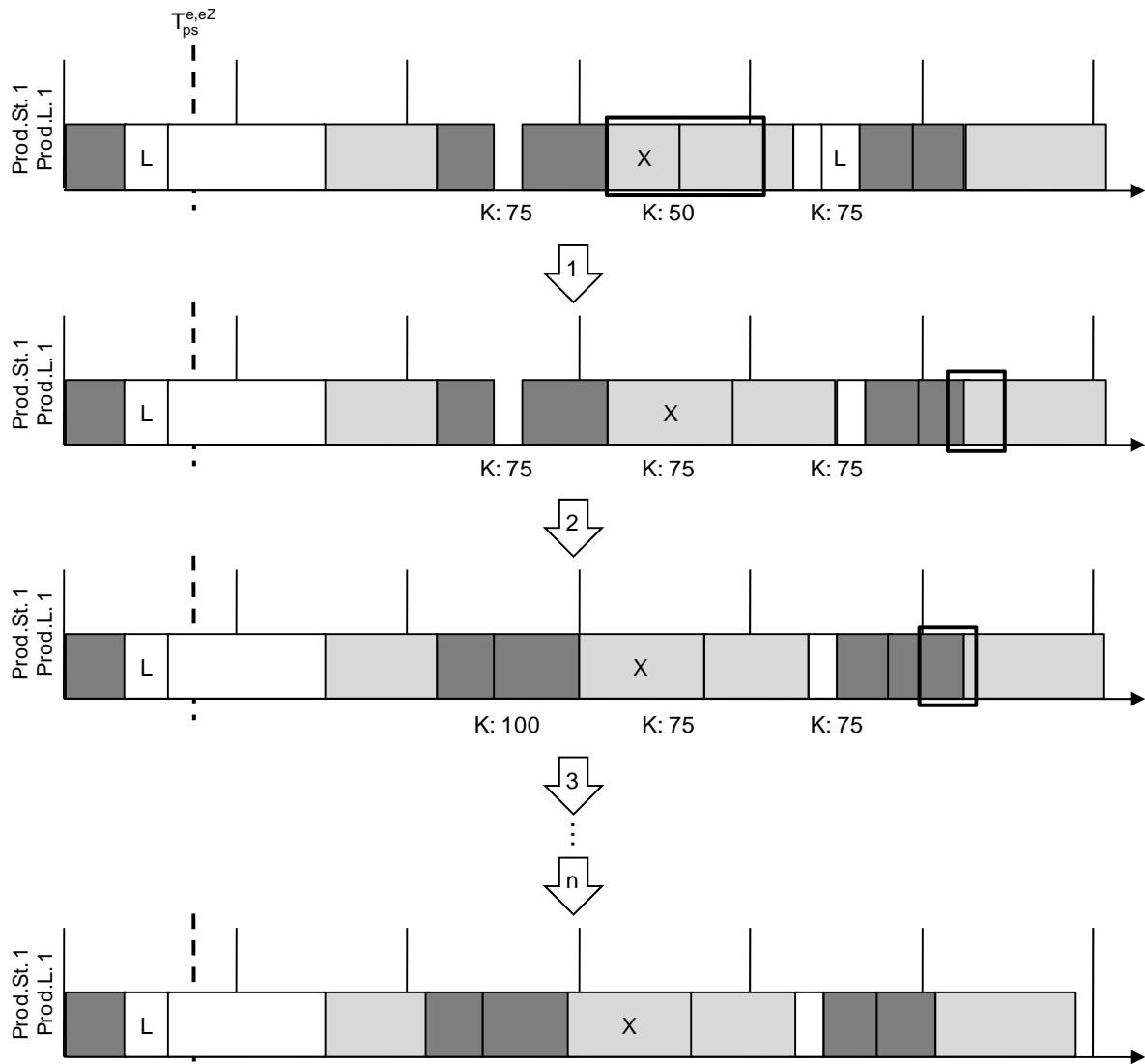


Abbildung 44: Ablauf der Funktion: InkremSchichtmodell

Analog dem Beispiel der *LoseVerschiebenOderTeilen*-Funktion existiert eine Überschneidung (schwarzes Rechteck) zwischen dem Los x sowie dessen Nachfolger. Vor der Konsistenzherstellung erfolgt zuerst die Ermittlung der relevanten Makroperioden. Um tiefgreifende Änderungen über den gesamten Planungshorizont zu vermeiden (dies widerspricht dem Nachbarschaftskonzept), werden nur die Makroperioden berücksichtigt, über die sich das Los x erstreckt. Dabei resultiert für Lose mit einer kurzen Bearbeitungsdauer eine geringere Wahrscheinlichkeit der Konsistenzherstellung, da sich in diesem Fall die Planänderungen auf bspw. nur eine Makroperiode auswirken könnten. Um diesen Nachteil entgegenzuwirken, werden mindestens drei Makroperioden betrachtet. Liegt also ein Los x in nur einer Makroperiode, so werden die vorhergehende sowie die nachfolgende Makroperiode der Menge der relevanten Makroperioden für potentielle Schichtmodellanpassungen hinzugefügt. Nachdem alle relevanten Makroperioden identifiziert wurden, werden solange Schichtmodellanpassungen vorgenommen, bis keine Inkonsistenz mehr vorliegt bzw. keine weitere Kapazitätserweiterung mehr möglich ist. Eine Schichtmodellanpassung besteht dabei aus zwei Schritten. Im ersten Schritt wird von den relevanten Makroperioden diejenige ermittelt, deren Kapazitätserhöhung

die geringsten zusätzlichen Kosten resultiert, die außerhalb des fixierten Schichthorizontes ($> T_{ps}^{e,fs}$) liegt sowie nicht den Parameter der maximalen Schichtveränderung verletzt (siehe Kapitel 5.2.2). Eine Erhöhung der Kapazität entspricht der Zuweisung des nächstgrößeren Schichtmodells. Dabei können zwei unterschiedliche Situationen auftreten: die Kapazitätserweiterung kann durch die Erhöhung des Leistungsgrades (in Abbildung 44 die erste Schichtmodellanpassung) oder durch die Erhöhung der Arbeitszeit (die zweite Anpassung des Schichtmodells in Abbildung 44) erreicht worden sein. Nach Zuweisung des neuen Schichtmodells werden die Lose der relevanten Makroperioden im zweiten Schritt der Schichtmodellanpassung frühestmöglich neu eingeplant. Hierzu dient wieder die Funktion *KalkLoszeiten* (siehe Kapitel 5.2.3.1.4.3). Resultiert aus der Schichtmodellanpassung eine konsistente Einplanung, wurde eine gültige Nachbarlösung erzeugt. Ist die Inkonsistenz weiterhin vorhanden, wurde sie durch die *KalkEndzeit*-Funktion (siehe Kapitel 5.2.3.1.4.5) auf das letzte Los der letzten betrachteten Makroperiode verschoben. Auf dieser Basis sowie der relevanten Makroperioden wird mit einer weiteren Schichtmodellanpassung fortgefahrene. Ist dies in keiner der betrachteten Makroperioden möglich, so wird der potentielle Nachbar verworfen, da ohne tiefgreifende Änderungen kein gültiger Plan erzeugt werden kann.

5.2.4 Entwicklung eines Verfahrens für den Horizont der Grobplanung

Der Planungsbereich der Grobplanung ist mit einer höheren Unsicherheit behaftet. Mit zunehmender Anzahl der in einen bestehenden Plan einzuarbeitenden Änderungen sinkt jedoch die Wahrscheinlichkeit, eine im Vergleich zum Optimum gute Plangüte zu erhalten. Da zudem die Flexibilität anderer Unternehmensbereiche (z.B. Werkslogistik, Einkauf, Personaleinsatzplanung) mit zunehmender Entfernung vom Beginn des Planungshorizontes steigt, erfolgt die Grobplanung in dieser Arbeit mit einem Neuplanungsverfahren. Im Gegensatz zur Feinplanung werden in diesem Verfahren einige Annahmen nicht betrachtet. So erfolgt die Planung ausschließlich auf Basis von Makroperioden, wodurch eine detaillierte Reihenfolgebildung entfällt. Um dennoch Rüstzeiten zu berücksichtigen, fließen diese als Durchschnittswerte⁸⁰ in die Planung ein. Außerdem wird von der simultanen Planung der Kapazitäten abgesehen. Stattdessen erfolgt die Planung auf Basis fester durch den Planer vorgegebener Kapazitäten (z.B. einer Standardkapazität).⁸¹

⁸⁰ Die Rüstzeit ($rz_{e,pl}$) eines Erzeugnisses e für eine Produktionslinie pl resultiert aus der Summation aller Rüstzeiten eines Erzeugnisses e zu Erzeugnis j , dividiert durch die Anzahl dieser Erzeugnisse j . Sei $E_{e,pl}$ die Menge der Erzeugnisse auf die von e auf der Produktionslinie pl gerüstet werden kann, dann gilt: $rz_{e,pl} = \left(\sum_{j \in E_{e,pl}} rz_{e,j,pl} \right) / |E_{e,pl}|$.

⁸¹ Es existieren zur Kopplung der Fein- und Grobplanung innerhalb der mehrstufigen Planung verschiedene Möglichkeiten, die durch die Reichweiten der verschiedenen Planungshorizonte begründet sind. Werden für alle Produktionsstufen identische Feinplanungshorizonte unterstellt ($\forall ps: T_{ps}^{e,GP} = T_{ps+1}^{e,GP}$), kann auf der ersten Produktionsstufe ($ps = 1$) aufgrund der mit dem Flexible Flow Shop verbundenen geschlossenen Produktion von $T_1^{b,GP}$ mindestens $|PS| - 1$ Makroperioden kein neuer – d.h., im vorherigen Planungslauf nicht vorhandener – Bedarf eingeplant werden. So besteht für dieses Enderzeugnis keine Möglichkeit, die erforderlichen Vorerzeugnisse vor $T_1^{b,GP} + |PS| - 1$ zu beziehen (dieser Bereich vergrößert sich natürlich bei höheren Vorlaufzeiten). Dies hätte zur Konsequenz, dass dieser Bedarf in der Grobplanung mit Verzug eingeplant werden müsste, obwohl freie Kapazitäten in der Feinplanung vorhanden sein können, die eine fristgerechte Produktion ermöglichen. Werden indes unterschiedlich lange Feinplanungshorizonte unterstellt, besteht das Problem

Aufgrund der jüngsten Entwicklungen moderner Computer und effizienter Solver können immer mehr Probleme mit Hilfe von Optimierungsmethoden gelöst werden (vgl. [Kurb05]). Aus diesem Grund findet für den Bereich der Grobplanung die Optimierung auch in dieser Arbeit Anwendung. Hierzu wird das in Kapitel 3.1.1 beschriebene MLCLSP modifiziert. Wenn den vorhandenen Bedarfen nicht ausreichend Kapazität gegenübersteht, können die Bedarfe nur mit Verzug oder ggf. gar nicht eingeplant werden. Zur Berücksichtigung etwaiger freier Kapazitäten in der Feinplanung werden in diesem Fall Bedarfe in diesen Planungsbereich verschoben. Zur Identifikation entsprechender Bedarfe dient ein Verfahren zur Bedarfsglättung.

5.2.4.1 Modell der Grobplanung

Als Grundlage dient das in Kapitel 3.1.1 beschriebene MLCLSP. Dieses wird nachfolgend so erweitert, dass es parallele Produktionslinien, eine Rüstzustandsübertragung zwischen Folgeperioden (vgl. [SBT+09]), unterschiedliche Puffer zwischen einzelnen Produktionsstufen sowie Verzugskosten umfasst.

Zur Modellierung des MLCLSP_{grob} werden folgende Mengen, Parameter und Variablen verwendet:

Definition der Mengen

$E = \{1..e\}$	Menge aller Erzeugnisse
$E_{pl} = \{1..e_{pl}\}$	Menge aller Erzeugnisse, die auf Produktionslinie pl gefertigt werden
$E_{pu} = \{1..e_{pu}\}$	Menge der Erzeugnisse, die über den Puffer pu laufen
$N_e = \{1..pl\}$	Menge der Nachfolger des Erzeugnisses e
$PL = \{1..pl\}$	Menge aller Produktionslinien
$PS = \{1..ps\}$	Menge aller Produktionsstufen
$PU = \{1..pu\}$	Menge aller Puffer

in dem Bestimmen dieser Größen. Da u.U. zwischen den Bearbeitungszeiten der Vorerzeugnisse Unterschiede existieren können, müssten die Abstände zwischen den Feinplanungshorizonten sehr groß gewählt werden. Es können bspw. von zwei Erzeugnissen (E1 und E2) in einer Makroperiode 10 Stück produziert werden. Benötigt die Produktion des Vorerzeugnisses E1 unter sonst gleichen Bedingungen doppelt so lange wie das von E2, müssten entsprechend zwei Zeiteinheiten zwischen den beiden Makroperiodenlängen liegen (d.h., $T_1^{e,GP} + 2 = T_2^{e,GP}$). Es ist offensichtlich, dass dies bei mehreren Produktionsstufen zu größeren Verzerrungen führen kann (dies führt vor allem dann zu Lücken im Plan, wenn das Erzeugnis mit der längeren Bearbeitungszeit selten auftritt). Aus diesem Grund werden in dieser Arbeit gleich lange Feinplanungshorizonte unterstellt. Zudem wird eine Kapazitätsflexibilität (dies entspricht der Differenz aus der Kapazität der Standardschicht und der Kapazität des gewählten Schichtmodells) der Produktionsstufen ($ps \neq 1$) im Feinplanungshorizont bei der Grobplanung mit berücksichtigt und eine Einplanung von Losen erlaubt. Ist diese erfolgreich, können die Bedarfe der ersten Produktionsstufe, die auch zu Beginn des Grobplanungshorizontes eingeplant wurden, über das Produktionssystem produziert werden. D.h., dass die Kopplung der Fein- und Grobplanung immer gleichzeitig für ein Los über alle Produktionsstufen in Abhängigkeit von der ersten Produktionsstufe erfolgt. Dies bedeutet, dass die von der Grobplanung auf der ersten Produktionsstufe für den $PZ(g)$ eingeplanten Bedarfe an die Feinplanung übergeben und mit ihren entsprechend notwendigen Vorerzeugnissen auf den anderen Produktionsstufen eingeplant werden.

$$T = \{T_{ps}^{b,GP} \dots T_{ps}^{e,GP}\} \quad \text{Länge des Planungshorizontes}$$

Definition der Parameter

$a_{e,j} \in \mathbb{N}$	Direktbedarfskoeffizient, der angibt, wie viele Einheiten des Erzeugnisses e zur Herstellung einer Einheit des Erzeugnisses j benötigt werden
$b_{e,t} \in \mathbb{N}$	Bedarf nach Erzeugnis e in Makroperiode t
$B \in \mathbb{N}$	Hinreichend große Zahl
$I_e^{init} \in \mathbb{N}$	Initialer Lagerbestand von Erzeugnis e
$kf_e \in \mathbb{R}^0$	Kapazitätsfaktor für die Lagerung einer Einheit von Erzeugnis e
$k_{pl,t} \in \mathbb{N}$	Kapazitätsangebot einer Produktionslinie pl in Makroperiode t
$lk_e \in \mathbb{R}^0$	Lagerkostenfaktor für eine Einheit des Erzeugnisses e
$p_{e,pl,t} \in \mathbb{R}^0$	Produktionskostensatz für ein Erzeugnis e in Makroperiode t auf Produktionslinie pl
$pf_{pu}^{max} \in \mathbb{N}$	Maximale Pufferkapazität des Puffers pu
$rz_{e,pl} \in \mathbb{N}$	Rüstzeit für Erzeugnis e auf Produktionslinie pl
$tb_{e,pl} \in \mathbb{N}$	Stückbearbeitungszeit für Erzeugnis e auf Produktionslinie pl
$vk_e \in \mathbb{R}^0$	Verzugskostenfaktor für eine Einheit des Erzeugnisses e
$vlz_e \in \mathbb{N}$	Vorlaufzeit von Erzeugnis e

Definition der Variablen

$I_{e,t} \in \mathbb{N}$	Lagerbestand für Erzeugnis e am Ende der Makroperiode t
$q_{e,pl,t} \in \mathbb{N}$	Losgröße für Erzeugnis e in Makroperiode t auf Produktionslinie pl
$RS_{e,t} \in \mathbb{N}$	Verzugsmenge eines Erzeugnisses e in Makroperiode t
$v_{pl,t} \in \mathbb{N}$	Anzahl der Erzeugnisse, die auf Produktionsline pl in Makroperiode t produziert wurden
$w_{e,pl,t} \in \{0,1\}$	Binärvariable, die angibt, ob zu Beginn der Makroperiode t auf Produktionslinie pl der Rüstzustand der Vorperiode für Erzeugnis e übernommen werden kann ($w_{e,pl,t} = 1$) oder nicht ($w_{e,pl,t} = 0$) (Variable ist 1, wenn zu Beginn der Makroperiode t die Produktionslinie l für Erzeugnis e gerüstet ist)
$x_{e,pl,t} \in \{0,1\}$	Binärvariable, die angibt, ob in der Makroperiode t auf Produktionslinie l eine Umrüstung für Erzeugnis e stattfindet ($x_{e,pl,t} = 1$) oder nicht ($x_{e,pl,t} = 0$)

5.2.4.1.1 Zielfunktion

Durch die Zielfunktion (5.89) werden die über alle Erzeugnisse, Produktionslinien und Makroperioden aufsummierten Kosten minimiert. Maßgeblich für deren Bestimmung ist der Planungsbereich der Grobplanung ($T_{ps}^{b,GP}$ bis $T_{ps}^{e,GP}$).

$$\min z = \text{Verzugskosten} + \text{Lagerkosten} + \text{Produktionskosten} \quad (5.89)$$

Verzugskosten (5.90) entstehen für nach dem Bedarfstermin gefertigte Erzeugnisse und sind das Produkt aus der Verzugsmenge und dem Verzugskostensatz des Erzeugnisses.

$$\text{Verzugskosten} = \sum_{e \in E} \sum_{t \in T} v k_e * RS_{e,t} \quad (5.90)$$

Lagerkosten (5.91) fallen für vor dem Bedarfstermin gefertigte Erzeugnisse an und sind das Produkt aus der Lagermenge und dem Lagerkostensatz des Erzeugnisses.

$$\text{Lagerkosten} = \sum_{e \in E} \sum_{t \in T} l k_e * I_{e,t} \quad (5.91)$$

Analog zu den Lagerkosten ergeben sich die Produktionskosten (5.92) aus dem Produkt der Produktionsmenge eines Erzeugnisses und dem produktionslinien- und makroperiodenabhängigen Produktionskostensatz.

$$\text{Produktionskosten} = \sum_{e \in E} \sum_{pl \in PL} \sum_{t \in T} p_{e,pl,t} * q_{e,pl,t} \quad (5.92)$$

5.2.4.1.2 Restriktionen

Die Lager- und Verzugsmengenbilanzgleichung (5.93) stellt sicher, dass der Bedarf gedeckt wird. Die gesamte Nachfrage eines Erzeugnisses setzt sich aus dem für dieses Erzeugnis bekannten Primärbedarf $b_{e,t}$ und aus dem durch die direkten Nachfolger verursachten Sekundärbedarfe zusammen. Die Berechnung dieses Sekundärbedarfs erfolgt mit Hilfe des Direktbedarfskoeffizienten $a_{e,j}$, der mit der Primärbedarfsmenge des übergeordneten Erzeugnisses j multipliziert wird. Um zudem die rechtzeitige Verfügbarkeit zu garantieren, wird für die Erzeugnisse eine Vorlaufzeit vlz_e berücksichtigt. Die in einer Periode zu produzierende Menge ergibt sich aus dem Lagerbestand der Vorperiode sowie dem angestrebten Lagerbestand der entsprechenden Periode. Falls das Kapazitätsangebot zur Deckung dieser Bedarfsumfänge nicht ausreicht, wird der Bedarf einer Makroperiode t reduziert und in die Nachfolgerperiode verschoben. Realisiert wird dies durch die Rückstandvariable $RS_{e,t}$.

$$I_{e,t-1} - RS_{e,t-1} + \sum_{pl \in PL} q_{e,pl,t-vlz_e} - \sum_{j \in N_e} \left(a_{e,j} * \sum_{pl \in PL} q_{j,pl,t} \right) - I_{e,t} + RS_{e,t} = b_{e,t} \quad \forall t \in T \quad (5.93)$$

Wenn die Differenz zwischen der Einplanung eines Loses und dem Beginn des Grobplanungshorizontes geringer ist als die Vorlaufzeit zur vorgelagerten Produktionsstufe, muss sichergestellt sein, dass der Lagerbestand des Vorerzeugnisses mindestens dem resultierenden Sekundärbedarf entspricht. Dazu dient die Restriktion (5.94).

$$\sum_{j \in N_e} \left(a_{e,j} * \sum_{pl \in PL} q_{j,pl,t} \right) \leq I_{e,t} \quad \begin{array}{l} \forall e \in E \\ \forall t: T_{ps}^{b,GP} \leq t \\ \leq T_{ps}^{b,GP} + v_{lz_j} \end{array} \quad (5.94)$$

Alle zur Herstellung eines Erzeugnisses benötigten Vorerzeugnisse müssen rechtzeitig zur Verfügung stehen. Die Einplanung dieser muss daher rechtzeitig unter Beachtung der Vorlaufzeit erfolgen. Daher muss in jeder Makroperiode t die Summe aus zur Produktion eingeplanten Menge und initialem Lagerbestand eines als Vorerzeugnis benötigten Erzeugnisses i mindestens so groß sein wie die zur Weiterverarbeitung benötigte Menge dieses Erzeugnisses. Diese ergibt sich aus der eingeplanten Menge aller konsumierenden Erzeugnisse j bis zur Makroperiode $t + v_{lz_j}$. Hierdurch wird die produktionsstufenübergreifende Konsistenz des Plans hergestellt (siehe (5.95)).

$$\sum_{pl \in PL} \sum_{\bar{t}=1}^t q_{i,pl,\bar{t}} + I_i^{init} \geq \sum_{j \in E} \sum_{pl \in PL} \sum_{t=1}^{t+v_{lz_j}} q_{j,pl,t} * a_{i,j} \quad \begin{array}{l} \forall e \in E \\ \forall t \in T \end{array} \quad (5.95)$$

In dem Produktionssystem existieren Puffer mit jeweils unterschiedlichen Pufferkapazitäten. Damit diese eingehalten werden, wird Restriktion (5.96) eingesetzt. Demnach darf der Lagerbestand aller in einem Puffer E_{pu} eingehenden Erzeugnisse mit entsprechendem Kapazitätsfaktor k_{fe} nicht größer als die maximale Kapazität des Puffers sein.

$$\sum_{e \in E_{pu}} I_{e,t} * k_{fe} \leq p_{f_{pu}}^{max} \quad \begin{array}{l} \forall p_{f_{pu}} \in PU \\ \forall t \in T \end{array} \quad (5.96)$$

Analog zu den unterschiedlichen Pufferkapazitäten müssen auch die Produktionskapazitäten beachtet werden. Diese sind von der jeweiligen Produktionslinie und Makroperiode abhängig. Hierzu beschränkt die Restriktion (5.97) die Summe der einer Makroperiode und Produktionslinie allokierten Produktionsmenge mit entsprechenden Rüstzeiten auf die maximal zur Verfügung stehende Kapazität $k_{pl,t}$. Die erzeugnisabhängige Rüstzeit ist dabei nur dann relevant, wenn der Rüstzustand des zu produzierenden Erzeugnisses nicht aus der Vorperiode übernommen werden kann.

$$\sum_{e \in E_{pl}} [tb_{e,pl} * q_{e,pl,t} + rz_{e,pl} * (x_{e,pl,t} - w_{e,pl,t})] \leq k_{pl,t} \quad \begin{array}{l} \forall pl \in PL \\ \forall t \in T \end{array} \quad (5.97)$$

Ein Erzeugnis kann in einer Makroperiode nur produziert werden, wenn die entsprechende Produktionslinie im richtigen Rüstzustand ist. Die Restriktion (5.98) stellt dies sicher.

$$q_{e,pl,t} - B * x_{e,pl,t} \leq 0 \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall t \in T \end{array} \quad (5.98)$$

Eine Produktionslinie kann zum Ende einer Makroperiode nur maximal einen Rüstzustand (ungerüstet/Rüstzustand für Erzeugnis e) haben. Folglich ist die Übertragung nur eines Rüstzustandes in die Nachfolgerperiode möglich (siehe (5.99)).

$$\sum_{e \in E_{pl}} w_{e,pl,t} \leq 1 \quad \begin{array}{l} \forall pl \in PL \\ \forall t \in T \end{array} \quad (5.99)$$

Eine Rüstzustandsübertragung in die Makroperiode t für ein Erzeugnis ist nur möglich, wenn sich in der Vorperiode $t - 1$ die Produktionslinie im gleichen Rüstzustand befand, d.h., in beiden Makroperioden das gleiche Erzeugnis produziert wird. Dieser Sachverhalt wird mit den Restriktionen (5.100) und (5.101) gewährleistet.

$$w_{e,pl,t} \leq x_{e,pl,t-1} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall t: T_{ps}^{b,GP} + 1 \leq t \\ \leq T_{ps}^{e,GP} \end{array} \quad (5.100)$$

$$w_{e,pl,t} \leq x_{e,pl,t} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall t \in T \end{array} \quad (5.101)$$

Ein Rüstzustand muss auch über n Makroperioden $t - 1, t, \dots, t + n$ übertragen werden können. Dies impliziert, dass in den Zwischenperioden $t, \dots, t + n - 1$ keine Veränderung des Rüstzustandes vorgenommen wird, also maximal ein Erzeugnis auf der Produktionslinie pl produziert wird. Hierfür wird die Variable $v_{pl,t}$ eingeführt ((5.102) und (5.103)).

$$w_{e,pl,t} + w_{e,pl,t+1} \leq 1 + v_{pl,t} \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall t: T_{ps}^{b,GP} \leq t \\ \leq T_{ps}^{e,GP} - 1 \end{array} \quad (5.102)$$

$$x_{e,pl,t} - w_{e,pl,t} + v_{pl,t} \leq 1 \quad \begin{array}{l} \forall e \in E \\ \forall pl \in PL \\ \forall t \in T \end{array} \quad (5.103)$$

Die Restriktion (5.104) stellt sicher, dass in der ersten Planungsperiode kein Rüstzustand existiert (die Produktionslinie also ungerüstet ist).

$$w_{e,pl,1} = 0 \quad \forall e \in E \quad \forall pl \in PL \quad (5.104)$$

Der Lagerbestand zum Planungszeitpunkt $I_{e,0}$ wird für jedes Erzeugnis e in der Restriktion (5.105) auf einen Eingangswert I_e^{init} gesetzt.

$$I_{e,0} = I_e^{init} \quad \forall e \in E \quad (5.105)$$

5.2.4.2 Bedarfsglättungsverfahren

Stehen den Bedarfen im Grobplanungshorizont nicht in ausreichender Menge und zum passenden Zeitpunkt entsprechende Kapazitäten gegenüber, ist ein Lösen des Planungsproblems ohne Verzug nicht möglich. Da angenommen wird, dass das Produktionssystem generell den Bedarfsumfängen entspricht, ist diese Unterdeckung nur temporär und z.B. das Resultat von Werbeaktionen. Um nun in einem solchen Fall etwaige Verzugskosten so gering wie möglich zu halten, ist es notwendig, den gesamten Planungshorizont zu betrachten. So können im Bereich der Feinplanung noch freie Kapazitäten bestehen, deren Ausnutzung eine fristgerechte Produktion der Bedarfsumfänge ermöglichen. Realisiert wird dies durch das Vorziehen entsprechender Bedarfsumfänge in die Feinplanung, in der – sofern ausreichend Kapazitäten vorhanden sind – die Bedarfe eingeplant werden. Stehen auch dort keine ausreichenden Kapazitäten zur Verfügung, werden diese Bedarfsumfänge in einer Liste gespeichert und beim nächsten Planungslauf der Grobplanung erneut versucht einzuplanen.

Basis dieses Verfahrens sind die im Rahmen der Optimierung ermittelten Lose mit Verzug (vm_j). Aufgrund der Zielsetzung minimaler Kosten sollen jedoch nicht diese Verzugsumfänge in die Feinplanung verschoben werden, sondern entsprechende Bedarfe zu Beginn des Horizontes der Grobplanung. Dadurch entsteht eine kürzere Lagerdauer der produzierten Erzeugnisse und folglich geringere Lagerkosten. Der Gesamtablauf ist in Abbildung 45 dargestellt.

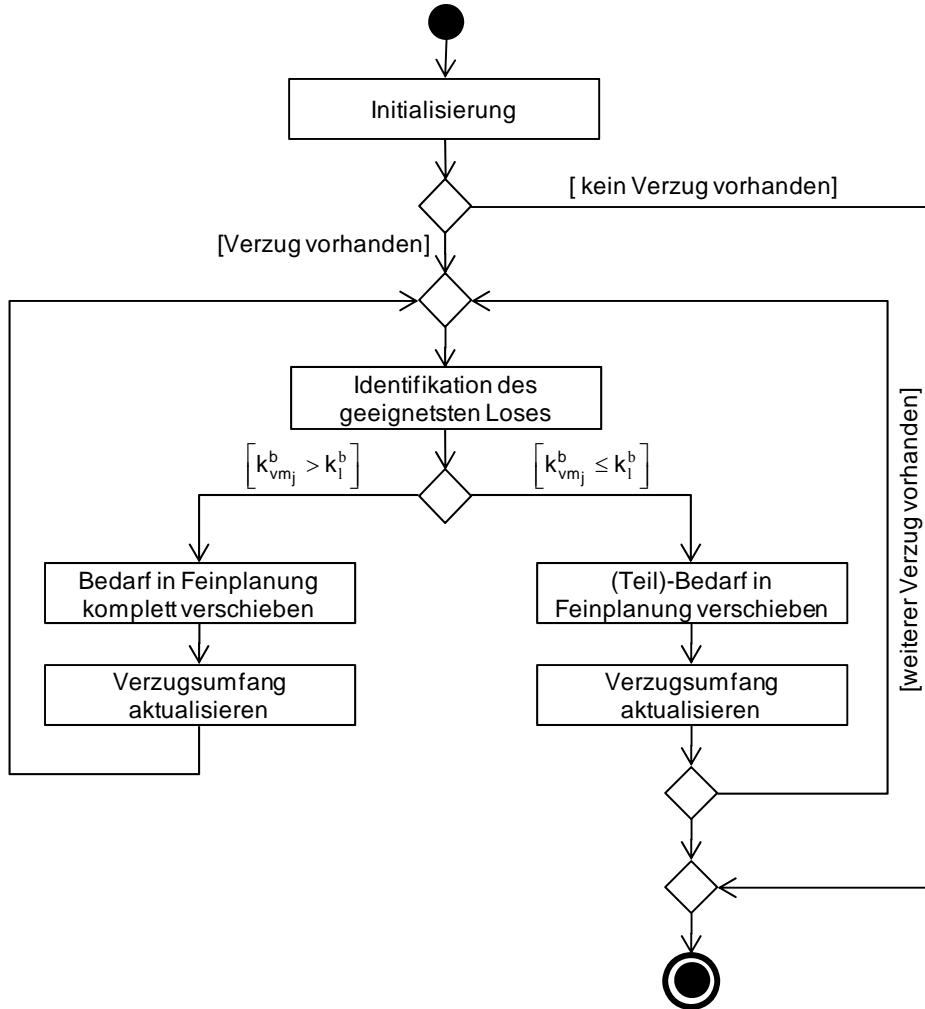


Abbildung 45: Ablauf des Bedarfsglättungsverfahrens

Nach der Initialisierung werden zur Bestimmung der in die Feinplanung zu verschiebenden Umfänge die einzelnen Verzugslose – diese liegen zeitlich aufsteigend sortiert vor – sukzessiv abgearbeitet. Für ein einzelnes Los werden vom Beginn des Grobplanungshorizontes aus auf Basis des durch den Optimierer ermittelten Plans geeignete Lose gesucht. Ein Los ist genau dann geeignet, wenn die Produktionslinie, auf der das Los liegt, auch zur Produktion der Verzugsmenge genutzt werden kann. Aus allen der in einer Makroperiode geeigneten Lose wird das Los gewählt, das über die größte Anzahl an Alternativlinien verfügt. Daraus resultiert eine höhere Wahrscheinlichkeit bei der Einplanung in der Feinplanung. Von diesem Los l wird anschließend die korrespondierende Bedarfsmenge $b_{l,t}$ angepasst. Ist der Kapazitätsbedarf des Verzugsumfangs (k^b_{vmj}) größer als die durch das geeignete Los gebundene Kapazität (k^b_l), wird die Bedarfsmenge des Loses komplett in die Feinplanung verschoben, der Verzugsumfang reduziert (siehe Formel (5.106)) und nach einem weiteren Los gesucht. Andernfalls wird mit der Formel (5.107) der in der Makroperiode t verbleibende Bedarf ermittelt. Die Differenz wird in die Feinplanung verschoben.

$$vm_j = vm_j - \left\lfloor \frac{b_{l,t} * tb_{e,pl}}{tb_{j,pl}} \right\rfloor \quad (5.106)$$

$$b_{l,t} = b_{l,t} - \left\lceil \frac{vm_j * tb_{j,pl}}{tb_{l,pl}} \right\rceil \quad (5.107)$$

Nachdem der gesamte Verzugsumfang reduziert wurde, wird die Optimierung erneut gestartet. Treten weitere Verzugsumfänge auf, wird mit ihnen analog verfahren.

6 Realisierung

„Ich weiss nicht, ob es besser wird,
wenn es anders wird.

Aber es muss anders werden,
wenn es besser werden soll.“

(Georg Christoph Lichtenberg)

In diesem Kapitel wird ein Verfahren zur Beurteilung der vorgestellten Planungsmethoden vorgestellt. Grundlage dieser Evaluation bildet die in Kapitel 3.3 beschriebene simulativ-empirische Analyse⁸². Um mit dieser Methode ein aussagekräftiges, objektives und zielgerichtetes Benchmarking zu erstellen, müssen der Untersuchungsbereich (hier die Produktionsplanung), die zu untersuchenden Planungsmethoden, die Datenbasis, die Benchmarkingkriterien und die Art der Analyse der Ergebnisse spezifiziert werden. Diese Punkte werden in den nachfolgenden Kapiteln konkretisiert. Hierzu wird im Kapitel 6.1 die zur Untersuchung verwendete Benchmarkingplattform vorgestellt. Dem schließt sich in Kapitel 6.2 eine konkrete experimentelle Untersuchung an. Basis dieser sind unterschiedlich komplexe Anwendungsszenarien.

6.1 Benchmarkingplattform

Die Benchmarkingplattform⁸³ besteht aus vier Komponenten, deren Funktionalität und Zusammenwirken nachfolgend vorgestellt werden. Zum Benchmarking der Verfahren sind grundsätzlich zwei unterschiedliche Arten an Daten – statische und dynamische – notwendig. Statische Daten sind über die Zeit unveränderlich und eine Modifikation dieser Daten kann nur mittels entsprechender Eingabemasken über die Stammdatenkomponente erfolgen. Dynamische Daten werden hingegen in Abhängigkeit der Zeit durch spezielle Verfahren ermittelt und laufend geändert. Die Komponente Benchmarkingdesign dient zur Spezifikation des konkreten Versuchsaufbaus und steuert den gesamten Versuchsablauf. Damit bildet sie den Kern der Benchmarkingplattform. Hierzu wird die Komponente in eine Parametrisierung- und Workflowkomponente gegliedert. Erste ermöglicht die Bestimmung des Untersuchungsgegenstands (z.B. Anzahl an Produktionslinien, Erzeugnissen oder Schichtmodellen), des Benchmarkingablaufs (Anzahl der Versuchsläufe und Umfang der Daten) sowie die Bestimmung der Benchmarkingkriterien. Auf Basis der gewählten Parametrisierung erzeugt die

⁸² Aufgrund der beschriebenen Probleme mit den anderen Methoden wird diese Analysemethode auch in der Literatur als der „vorziehenswürdige Ansatz“ bezeichnet (vgl. [PfHe82]).

⁸³ Die Anwendung wurde als Client/Server-Applikation realisiert. Die Bedienung erfolgt über ein Web-Frontend, die Ausführung der Applikation auf einem Server. Aufgrund der meist besseren Hardwareausstattung können somit bessere Ergebnisse hinsichtlich der Laufzeit und eine bessere Vergleichbarkeit erzielt werden. Die Implementierung der Benchmarkingplattform erfolgte mit PHP. Neben den Vorteilen einer dynamischen Typisierung und Plattformunabhängigkeit zeichnet sich PHP insbesondere durch die native Unterstützung von XML-Dateien aus, die für den vorliegenden Anwendungsfall gerade im Bereich der Workflowkomponente (siehe Kapitel 6.1.2.2) von entscheidender Bedeutung ist.

Workflowkomponente eine dynamische Planungsumgebung und steuert die rollierende Ausführung der einzelnen Verfahren. Deren Ergebnisse werden anhand zu spezifizierender Kriterien in der Reportingkomponente berechnet und graphisch und tabellarisch aufbereitet. Ergänzend existiert noch eine Konfigurationskomponente, die zur Administration der Anwendung dient.

6.1.1 Stammdatenkomponente

In der Stammdatenkomponente werden sämtliche Daten verwaltet, die für einen Benchmarklauf erforderlich sind. Diese Informationen sind statisch und spiegeln die Produktionsumgebung innerhalb eines Unternehmens wider. In Kapitel 6.1.1.1 wird zur Repräsentation dieser Daten eine Klassenstruktur eingeführt. In dem nachfolgenden Kapitel (6.1.1.2) wird beispielhaft die GUI beschrieben.

6.1.1.1 Klassenstruktur der Stammdaten

Die Produktionsumgebung besteht grundsätzlich aus zwei wesentlichen Bereichen. Dies sind die eigentliche Produktionsstruktur sowie der Schichtkontext. Die Produktionsstruktur bildet dabei den Aufbau der Fertigung ab. Der Schichtkontext betrachtet die Objekte und Parameter, die direkt oder indirekt mit der Schichtplanung im Zusammenhang stehen. Abbildung 46 zeigt die logische Strukturierung der Stammdaten.

Neben einer eindeutigen ID, die zur systemweiten Identifikation eines Datensatzes verwendet wird, verfügen nahezu alle Entitäten über einen Namen. Des Weiteren wurden die Parameter als Attribute der zugehörigen Entität modelliert. Sie können somit im Rahmen der Applikation als Eigenschaften der jeweiligen Objekte adressiert werden. Neben den zahlreichen skalaren Attributen, verfügen die Entitäten über Beziehungen zu anderen Entitäten. Diese sollen dabei ebenfalls als virtuelle Attribute adressiert werden können. Um bspw. alle Erzeugnisse (Product) zu erhalten, die in einem bestimmten Puffer gelagert werden können, kann auf das Attribut „possibleProducts“ des jeweiligen Puffers zurück gegriffen werden. Im Rahmen des Klassendiagramms wurden daher an den entsprechenden Assoziationsenden die Namen der jeweiligen Eigenschaften inkl. deren Kardinalität festgehalten. Eine besondere Stellung nehmen jene Attribute der Entität „Product“ ein, die mit dem Namen „create“ oder „change“ beginnen. Diese Werte beschreiben erzeugnisspezifische Wahrscheinlichkeiten und Verteilungsparameter zur späteren dynamischen Generierung der Bedarfe.

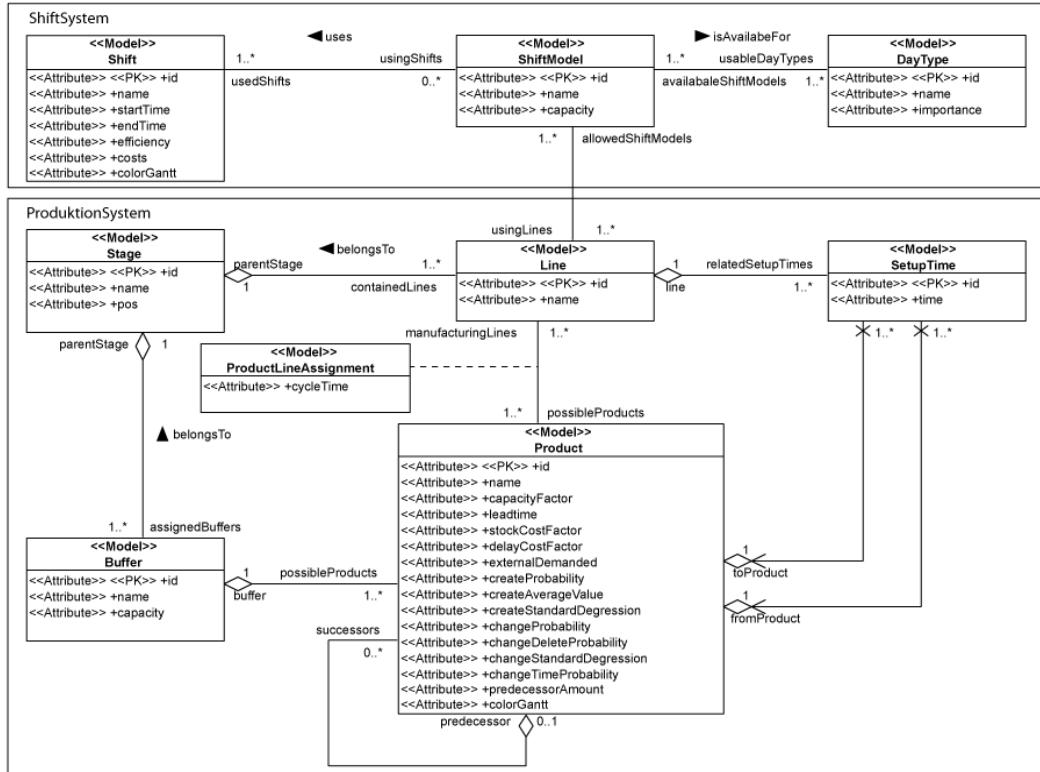


Abbildung 46: Klassendiagramm der Stammdatenkomponente

6.1.1.2 Frontend der Stammdatenverwaltung

Entsprechend der im vorherigen Kapitel eingeführten Unterteilung (Produktionsstruktur und Schichtkontext) gestaltet sich auch die Benutzerschnittstelle zur Stammdatenverwaltung. Aufgrund einer generischen Eingabestruktur wird nachfolgend der Aufbau der entsprechenden Eingabemasken exemplarisch für die Entität „Erzeugnis“ diskutiert.

Die einzelnen Entitäten (hier Erzeugnisse) werden in einer Listenansicht visualisiert (siehe Abbildung 47). Um auch bei großen Produktionskontexten ein schnelles Auffinden des gewünschten Objekts zu ermöglichen, befindet sich im oberen Bereich eine Suchfunktion. Neben den grundlegenden Eigenschaften einer Entität (ID, Name, etc.), werden Links zum Bearbeiten und Löschen des jeweiligen Datensatzes angezeigt (in Abhängigkeit der jeweiligen Entität können auch noch weitere Links vorhanden sein, z.B. Zuordnung von Schichten zu einem Schichtmodell). Zusätzlich zu dieser Listenansicht sind abhängig von der Entität weitere Informationen vorhanden. So können z.B. bei einer komplexen Erzeugnisstruktur die einzelnen Vorgänger-/Nachfolgebeziehungen zwischen den Erzeugnissen komplex und unübersichtlich werden. Zur transparenten Ansicht steht ein zusätzlicher Abhängigkeitsgraph zur Verfügung, der eine schematische Darstellung der Erzeugnisstruktur widerspiegelt.

The screenshot shows the FlexFlowBench software interface. The top navigation bar includes a logo, a dropdown for the current production system (set to 'Produktionssystem A'), and a link to 'Neues anlegen'. The left sidebar contains four sections: 'Stammdaten' (with sub-sections for Produktionssystem and Schichtsystem), 'Benchmark.Design' (with sub-sections for Neuer Durchlauf and Vorhandene Durchläufe), 'Reporting' (with sub-sections for Übersicht, Daten-Browser, Statistik-Browser, Gantt-Chart (Ist-Werte), and Gantt-Chart (Plan-Werte)), and 'Konfiguration' (with sub-sections for System-Check, Einstellungen, DB-Export/Import, Verfahren, Benchmark-Kriterien, and Diagrammtypen). The main content area features a table titled '[Neues Element]' with columns for ID, Name, and Einträge/Seite (set to 25). Below the table is a message 'Einträge (insgesamt): 5' and a count '[1]'. A detailed table follows with columns for ID, Name, Kapazitätsfaktor (kapF), Vorlaufzeit (vlz), Externe Bedarfsanforderung, and Aktion. The data is as follows:

ID	Name	Kapazitätsfaktor (kapF)	Vorlaufzeit (vlz)	Externe Bedarfsanforderung	Aktion
76	E1	10	00:20:00	ja	[Bearbeiten] [Löschen]
77	E2	15	00:40:00	ja	[Bearbeiten] [Löschen]
78	E3	10	00:30:00	nein	[Bearbeiten] [Löschen]
79	E4	15	00:15:00	nein	[Bearbeiten] [Löschen]
80	E5	10	00:00:00	nein	[Bearbeiten] [Löschen]

Below the table is a class diagram with entities F5, E5, E3, E4, E1, and E2. F5 is at the top, connected to E3 and E4 with multiplicity 1. E3 and E4 are connected to E1 and E2 respectively with multiplicity 2.

Abbildung 47: Abbildung der Erzeugnisstruktur

Über den Link „Neues Element“ bzw. „Bearbeiten“ für ein bereits vorhandenes Objekt wird eine weitere Maske aufgerufen, mittels derer alle Attribute eines Datensatzes angelegt bzw. geändert werden können (siehe Abbildung 48). Die einzelnen Masken unterstützen bestmöglich die Eingabe der Daten. So dienen bspw. zur Angabe von prozentualen Werten Schieberegler. Ist die jeweils editierte Entität mit einer weiteren Entität verknüpft (erkennbar durch die Raute im obigen Klassendiagramm), kann mittels einer Auswahlbox die jeweils aggregierte Entität gewählt werden. Bei einem Erzeugnis trifft dies neben dem Puffer auch auf die selbstreferenzierende „predecessor“-Assoziation zu. Damit kann definiert werden, ob und welches Vorerzeugnis erforderlich ist, um das aktuelle Erzeugnis produzieren zu können. Bei Assoziationen die eine n:m-Kardinalität aufweisen, wird hingegen auf eine separate Bearbeitungsseite zurückgegriffen, die eine Zuordnung sowie etwaige Anpassung von AssoziationsAttributen (z.B. ist die Taktzeit ein Relationsattribut der Verbindung Produktionslinie und Erzeugnis) ermöglicht.

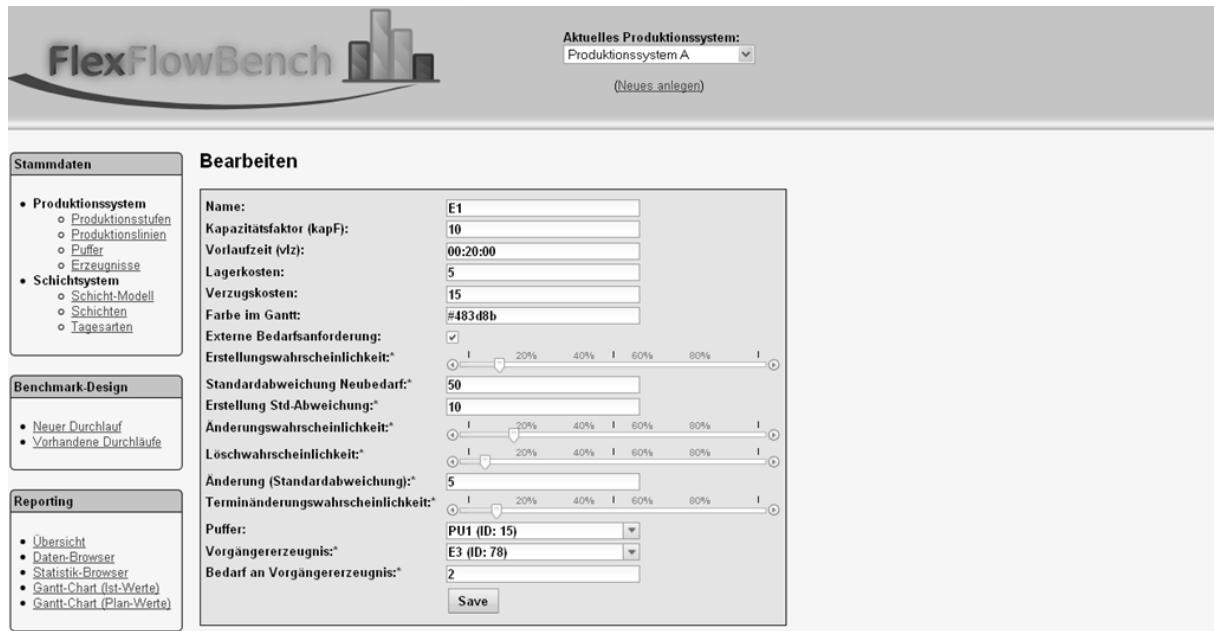


Abbildung 48: Stammdatenmasken der Benchmarkingplattform

6.1.2 Benchmarkingdesign

Das Benchmarkingdesign besteht aus zwei Komponenten. Dies sind die Parametrisierungskomponente zur Spezifikation des Untersuchungsgegenstands sowie die Workflowkomponente zur Steuerung des Benchmarkinglaufs.

6.1.2.1 Parametrisierungskomponente

Die Parametrisierungskomponente stellt die gesamten Konfigurationsoptionen für die Durchführung und Anpassung eines Benchmarkingprozesses dar. Sie setzt vollständig gepflegte Stammdaten für das aktuelle Produktionssystem voraus, die innerhalb dieser Komponente angepasst bzw. parametrisiert werden können. Grundsätzlich gilt die angepasste Konfiguration stets nur für den direkt folgenden Durchlauf (auch Workflow genannt) der Verfahren. Zum Design einer Benchmarkingstudie dient ein spezieller Wizard, der bei der Einstellung der folgenden Parameter Unterstützung leistet.

Auswahl der zu benchmarkenden Verfahren

In einer Benchmarkingstudie erfolgt jeweils ein paarweiser Vergleich von Methoden. Aus dem Vorliegen verschiedener Methoden ergibt sich somit eine Vielzahl unterschiedlicher Paarungen. So können die unterschiedlichen Methoden zur Feinplanung (mathematisches Modell, Eröffnungs- und Verbesserungsverfahren) untereinander bewertet werden. Um zudem das Zusammenspiel der einzelnen Methoden zu untersuchen, können diese verkettet werden. So mit ist es möglich, bspw. die Ergebnisse des mathematischen Modells einer Verknüpfung aus Eröffnungsverfahren und Verbesserungsverfahren gegenüberzustellen. Bei dem Verbesserungsverfahren kann zudem gewählt werden, ob das Simulated Annealing oder die Tabu Search Anwendung finden sollen. Durch Hinzunahme der Grobplanung kann ferner deren Einfluss auf die Planungsergebnisse bestimmt werden.

Aufbau des Produktionskontextes

Der Aufbau des Produktions- und Schichtsystems kann großen Einfluss auf die Ergebnisermittlung haben, weshalb dieser weiter zu spezifizieren ist. Neben der Auswahl der zur Verfügung stehenden Schichtmodelle und Produktionslinien umfasst dies auch eine Reduzierung des Erzeugnisspektrums. Zur Vermeidung von Inkonsistenzen bezieht sich die Selektion der Erzeugnisse jedoch nur auf Enderzeugnisse. Des Weiteren ist sicherzustellen, dass für jede Produktionslinie mindestens ein Schichtmodell mit einem Kapazitätsangebot größer 0 existiert.

Definition des Zeitmodells

Der Aufbau des Zeitmodells kann gravierende Auswirkungen auf die Laufzeit und Ergebnisse besitzen und ist abhängig von dem jeweiligen Unternehmen. Zur Definition unterschiedlicher Szenarien sowie dem Umfang an Versuchsläufen dient daher ein weiterer Wizardschritt. In diesem Schritt erfolgt insbesondere die Bestimmung der Anzahl und die Detaillierung der zu betrachtenden Makroperioden. Diese entsprechen dem Untersuchungszeitraum, welcher simuliert werden soll. Für jede Makroperiode ist dabei festzulegen, welcher Tagesart sie entspricht (Werktag, Wochenende, Feiertag, etc.). Dies bildet die Grundlage der Zuordnung unterschiedlicher periodenabhängiger Schichtmodelle. Des Weiteren sind die für die rollierende Planung erforderlichen Parameter zu definieren. Hierzu gehören die einzelnen Planungsfenster (für die Feinplanung und ggf. die Grobplanung) sowie die Länge der eingefforenen Zone und des fixierten Schichthorizontes. Es wurde bereits beschrieben, dass auch die Länge einer Mikroperiode (bzw. die Anzahl der Mikroperioden pro Makroperiode) ein entscheidender Faktor für die Laufzeit optimaler Modelle ist, welcher daher zu bestimmen ist.

Definition der Zielsetzung

In diesem Wizardschritt erfolgt die Definition unterschiedlicher Zielsetzungen. So besteht die Möglichkeit, einzelne Kostenarten (z.B. Verzugskosten) zu negieren. Zudem können die Bestandteile der Zielfunktion – deren Kostensätze auf realen, d.h., in den Stammdaten gepflegten, Werten basieren – über unterschiedliche Gewichtungen in Steuerparameter transformiert werden. Eine Gewichtung der Lagerkosten um den Faktor zwei bedeutet folglich, dass sämtliche Werte aus dem Produktionssystem, die sich auf diese Kostenart beziehen, mit dem Faktor zwei multipliziert als Input an die Planungsverfahren übergeben werden.

Generierung und Anpassung der dynamischen Bedarfe

Ein wesentlicher Teil der Parametrisierungskomponente ist die Definition der Bedarfe⁸⁴ sowie ihrer dynamischen Veränderung innerhalb des Planungshorizontes. Um im Rahmen eines simulativ-empirischen Benchmarkings aussagekräftige Werte zu erzielen, ist die Ausgestaltung und Verteilung der entsprechenden Bedarfstermine und Bedarfsmengen ein entscheidender Faktor. Grundlage hierfür ist ein zufallsbasiertes Verfahren. Um diese Bedarfe in adäquater Form bestimmen zu können, ist zwischen der Art einer Bedarfsänderung, der Wahrscheinlichkeit für das Auftreten sowie dem Umfang der Bedarfe zu differenzieren. Ein Bedarf ist durch einen Termin, einem Erzeugnis und einer Menge eindeutig beschreibbar. Dementspre-

⁸⁴ Auf eine explizite Ausgestaltung und Integration stochastischer prozessbedingter Störungen wird im Weiteren verzichtet.

chend ist zwischen Mengenänderungen (Neubedarfen, Stornierungen sowie Bedarfserhöhungen bzw. -senkungen), Terminänderungen sowie Erzeugnisänderungen zu differenzieren. Da sich eine Erzeugnisänderung als Löschung des alten Bedarfs von Erzeugnis A und Erstellung eines neuen Bedarfs von Erzeugnis B darstellen lässt, wird diese Änderung im Weiteren nicht gesondert beachtet. Bei der Erstellung von Bedarfen können die verschiedenen Arten von Bedarfsänderungen jeweils über eine unterschiedliche Wahrscheinlichkeit verfügen. So kann bspw. in Zeiten hoher Nachfrage die Wahrscheinlichkeit für das Auftreten von Neubedarfen höher als die von Stornierungen sein. Um verschiedene Szenarien nachstellen zu können, werden diese äußeren Einflüsse dem Verfahren vom Planer vorgegeben. Im Falle einer aufgetretenen Änderung, ist deren Umfang zu bestimmen. Bei einem Neubedarf ist zunächst die Art des Erzeugnisses zu ermitteln. Hierzu dient eine für jedes Erzeugnis zu spezifizierende Eintrittswahrscheinlichkeit. Im Anschluss wird der Bedarfstermin festgelegt. Um diesen zu bestimmen, wird ein gleichverteilter Zufallswert aus dem Intervall [Startzeitpunkt der Periode, Endzeitpunkt der Periode] auf der Basis von Mikroperioden generiert. Abschließend wird die Menge des Neubedarfs erzeugt. Hierzu wird auf der Grundlage eines vorher definierten Mittelwertes und einer Standardabweichung eine standardnormalverteilte Zufallszahl gebildet und mit Hilfe einer linearen Transformation an die vorgegebenen Parameter angepasst. Derart wird sichergestellt, dass der generierte Mittelwert für den Neubedarf nicht allzu häufig vom Mittelwert abweicht. Analog zu dem Neubedarf wird ein alternativer Termin bei einer Terminänderung bestimmt. So liegt dieser gleichverteilt an einem beliebigen Zeitpunkt innerhalb des betrachteten Zeitraums. Bei einer Mengenänderung wird angenommen, dass kleine Änderungen wahrscheinlicher als Veränderungen großen Ausmaßes sind. Aus diesem Grund wird die Normalverteilung zur Bestimmung der neuen Bedarfsmenge herangezogen. In dieser wird der Mittelwert dem bisherigen Umfang gleichgesetzt. Die Standardabweichung wird von der vorgegebenen Standardabweichung des betrachteten Erzeugnisses übernommen. Für jede Makroperiode wird das in Abbildung 49 dargestellte Verfahren zur Änderung der Bedarfe genutzt. Beim Starten des Verfahrens werden sukzessiv die einzelnen Makroperioden betrachtet. Tritt in einer Makroperiode eine Änderung auf (zu $x\%$), ist als nächstes hinsichtlich der Art der Änderung zu unterscheiden. Hierbei gibt es generell zwei Fälle. Entweder erfolgt eine Änderung an einem bereits bekannten Bedarf oder die Generierung eines neuen Bedarfs. Im Falle der Änderung eines bestehenden Bedarfs wird der zu ändernde Bedarf zufallsbasiert aus der Menge der in der Makroperiode enthaltenen Bedarfe bestimmt. Anschließend ist zu unterscheiden, ob eine Änderung des Bedarfstermins oder des Bedarfsumfangs erfolgen soll. Die Bestimmung eines neuen Bedarfstermins erfolgt mit Hilfe der Gleichverteilung. Hierzu ist zu Beginn des Verfahrens ein genereller Bereich erlaubter Verschiebungen (z.B. $+/- 3$ Makroperioden) festzulegen. Die Änderung einer Bedarfsmenge erfolgt analog zu der Änderung des Bedarfstermins auf Basis gewählter Einstellungen (z.B. $+/- 100\%$). Zur Ermittlung eines gänzlich neuen Bedarfs werden zuerst zufallsbasiert ein Erzeugnis und ein Bedarfstermin bestimmt. Anschließend wird der Bedarfsumfang bestimmt.

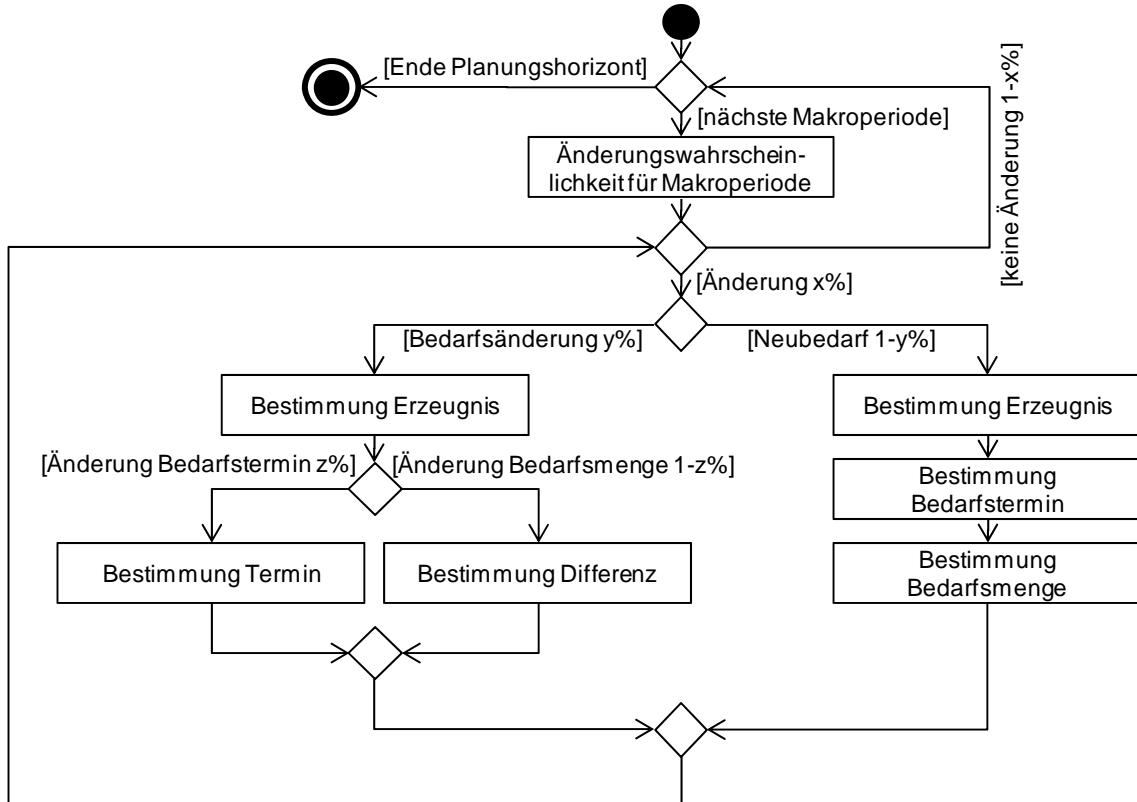


Abbildung 49: Ablauf des Bedarfsgenerators

Neben der dynamischen Aktualisierung und Generierung von Bedarfen beim Fortschreiten des Planungshorizontes dient das Verfahren zudem zur Erstellung einer Initialreihe. In diesem Fall sind die Länge des Planungshorizontes, die Gesamtzahl an zu erstellenden Bedarfen über den Planungshorizont sowie ein Minimum an Bedarfen pro Periode zu spezifizieren. Zur Bewertung der Planungsverfahren unter extremen Bedingungen können zudem Peaks (Ballungszentren, in denen eine viel größere Menge an Bedarfen besteht – z.B. infolge von Werbeaktionen) definiert werden.

Mit dieser Methode können auch für sehr große Zeiträume sowie einem großen Erzeugnisspektrum schnell und effizient Bedarfe angelegt werden, die den Kriterien der in den Stammdaten hinterlegten Verteilungsparameter genügen. Die Integration dieser Methode in die Benchmarkingkomponente ist in Abbildung 50 dargestellt.

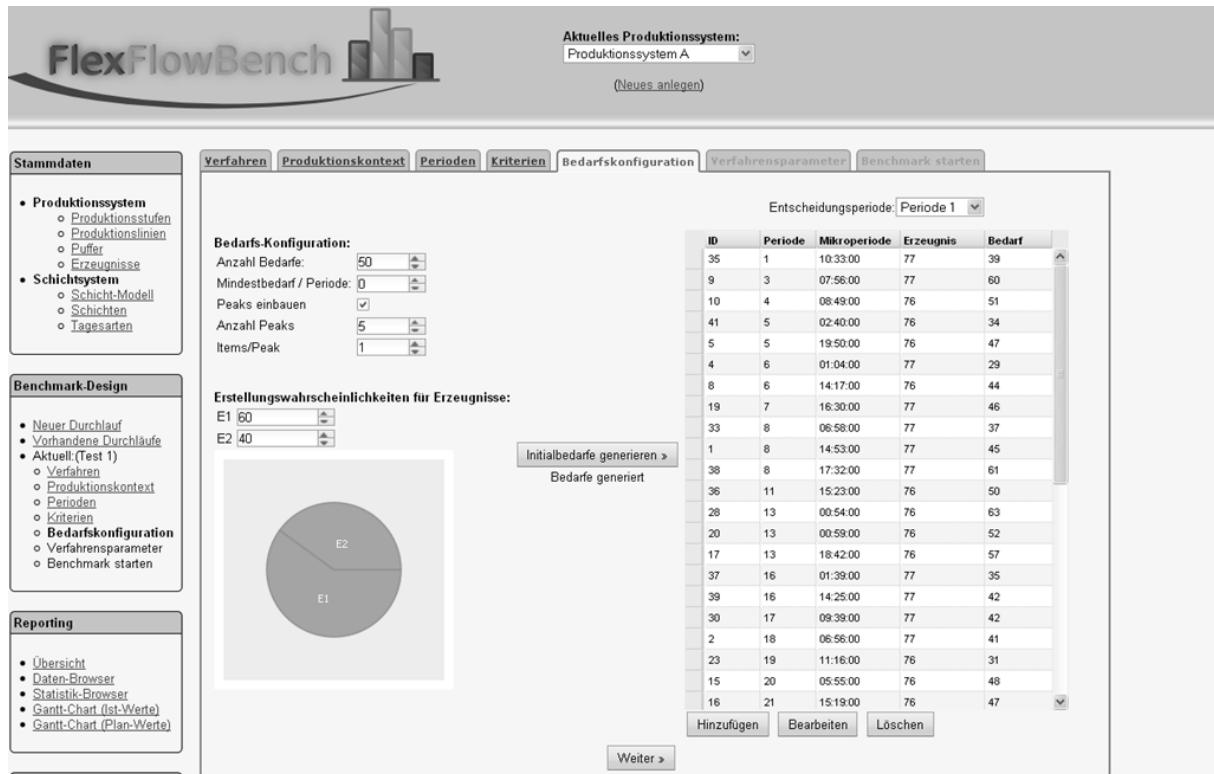


Abbildung 50: Bedarfsgenerator in der Benchmarkingkomponente

Konfiguration der einzelnen Planungsverfahren

Dieser Bereich ermöglicht die Bestimmung verfahrensspezifischer Parameter (siehe Abbildung 51). Dies umfasst z.B. die Festlegung der Temperaturen für das Simulated Annealing Verfahren, die Länge der Tabu-Liste für das Tabu Search-Verfahren oder die Angabe maximal möglicher Laufzeiten zur Lösungssuche.

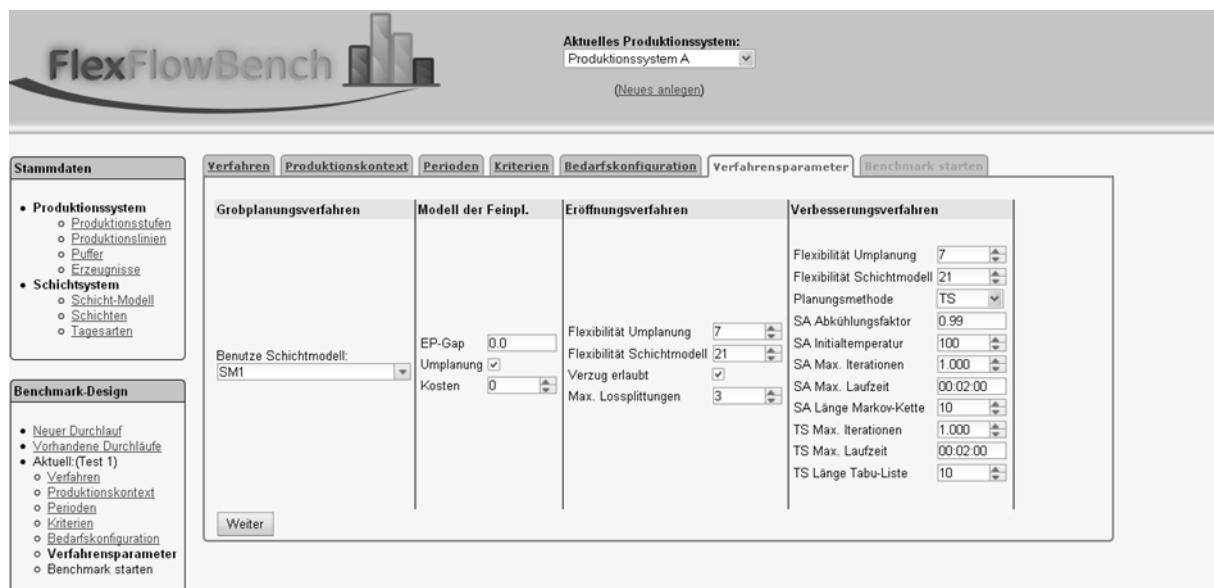


Abbildung 51: Verfahrensoptionen

6.1.2.2 Workflowkomponente

Nach der Spezifikation des Untersuchungsgegenstandes schließt sich die Durchführung des eigentlichen Benchmarkingprozesses an. Die dazu notwendige Ablaufsteuerung ist ein zentraler Bestandteil der Benchmarkingplattform. Die Abbildung 52 illustriert die Funktionsweise dieser Komponente.

Initialisierung

Nach dem Start eines Benchmarkingprozesses ist die komplette Workflowumgebung zu initialisieren. Dies umfasst auch die Berücksichtigung der innerhalb der Parametrisierung konfigurierten Optionen und Einstellungen, wie z.B. die Speicherung der Verfahrenskonfiguration in entsprechenden Konfigurationsdateien.

Periodenauswahl und Bedarfsveränderung

Zur Simulation der rollierenden Planung sind sukzessiv (um jeweils eine Makroperiode) der Planungshorizont fortzuschreiben und die Planungsumgebung zu aktualisieren. Zur Generierung aussagekräftiger Ergebnisse sollen die unterstellten Annahmen bestmöglich der Realität entsprechen. In dieser stellt die rollierende Planung einen fortwährenden Prozess dar. In einem zeitlich begrenzten System, wie die Durchführung eines Benchmarkingprozesses mit einer fest definierten Anzahl an Perioden, kann dies nicht komplett abgebildet werden. Um dennoch sinnvolle Benchmarkingergebnisse zu erhalten, dürfen nur jene Resultate Berücksichtigung finden, die in einem „eingeschwungenen System“ generiert werden. Ein System gilt als eingeschwungen, wenn jede Makroperiode (im Weiteren auch als benchmarkingrelevante Makroperiode bezeichnet) entsprechend der Anzahl des gesamten Planungshorizontes geplant wurde. Sei t_a (aktive Periode) die erste Makroperiode eines Planungshorizontes, dann gilt:

$$\left((T_{ps}^{e,GP} - T_{ps}^{b,GP}) + (T_{ps}^{e,FP} - T_{ps}^{b,FP}) \right) \leq t_a \leq |T| - \left((T_{ps}^{e,GP} - T_{ps}^{b,GP}) + (T_{ps}^{e,FP} - T_{ps}^{b,FP}) \right) \quad (6.1)$$

Gemäß der Formel (6.1) terminiert der gesamte Benchmarkinglauf, wenn die aktive Periode (t_a) den benchmarkingrelevanten Bereich überschreitet.

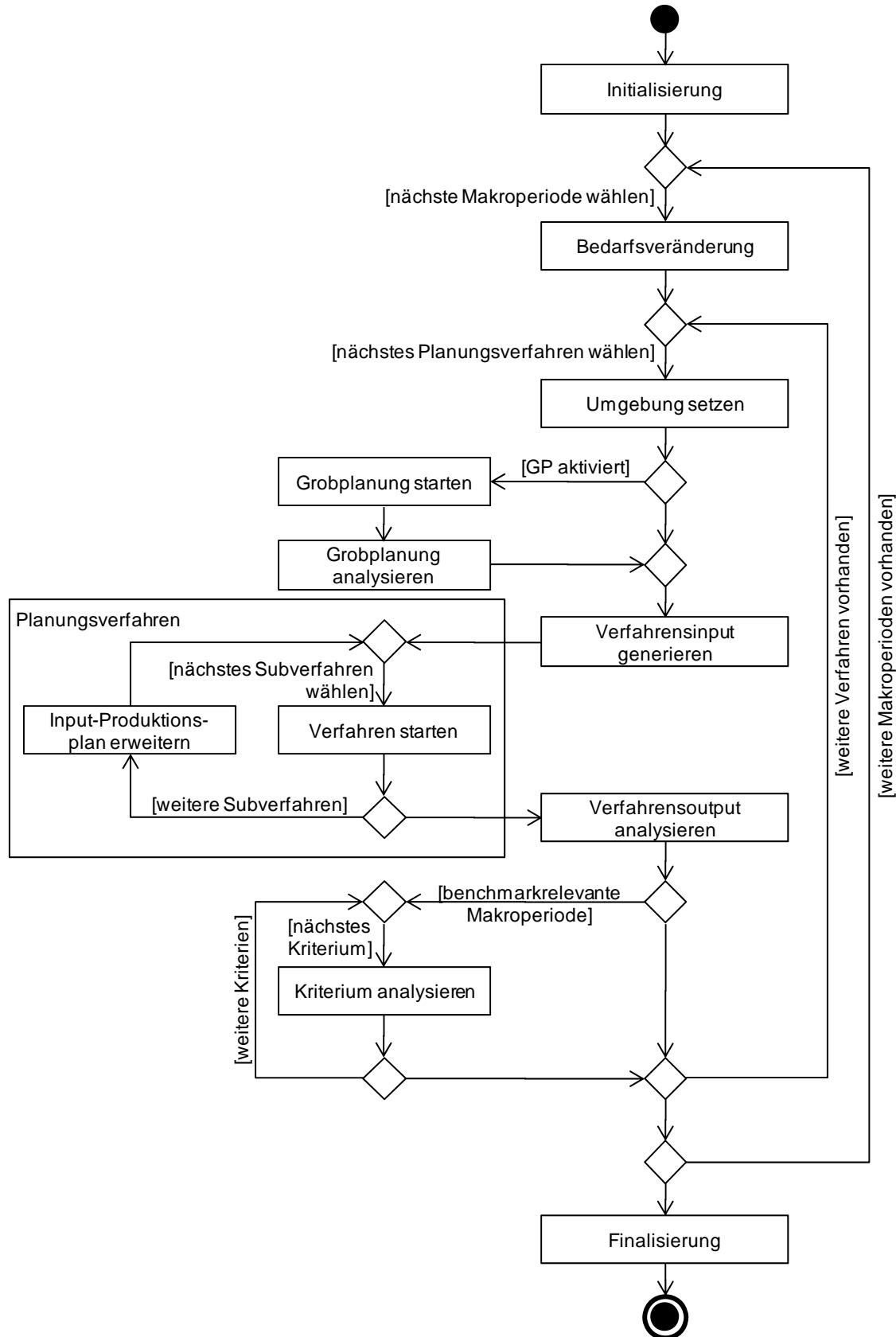


Abbildung 52: Ablaufdiagramm Workflowkomponente

Verfahrensdurchlauf

Das Kernelement eines jeden Benchmarkinglaufs stellt die Ausführung der eigentlichen Planungsverfahren mit den korrekten Eingabeparametern dar. Die Integration dieser Verfahren⁸⁵ in die Benchmarkingkomponente erfolgt u.a. mittels verschiedener XML-Dateien. Da diese auch für die Analyse der Planungsergebnisse relevant sind, sollen sie kurz skizziert werden. Die XML-Datei „Produktionskontext“ enthält die statischen Elemente des Produktionssystems. Dies umfasst neben dem Aufbau des Produktionssystems auch die Erzeugnisstruktur sowie die Schichtmodelle und deren mögliche Produktionslinienzuordnungen. Diese Datei beinhaltet somit eine ggf. durch die Parametrisierungskomponente eingeschränkte Sicht auf die Stammdaten. Die XML-Datei „Input-Produktionsplan“ untergliedert sich in zwei Teile. So sind in dieser Informationen über den aktuellen Stand der rollierenden Planung sowie deren Parameter (aktueller Planungshorizont, Angabe der aktuell eingefrorenen Zonen etc.) und die innerhalb des Planungshorizontes fälligen Bedarfe notiert. Ist der zu planenden Makroperiode bereits eine Produktionsplanung vorausgegangen, so sind deren Ergebnisse in Form von Schichtmodellzuordnungen und Losen ebenfalls in dieser Datei enthalten. Die XML-Datei „Output-Produktionsplan“ wird vom entsprechenden Planungsverfahren nach der Durchführung geschrieben und enthält die gewählten Schichtmodellzuordnungen sowie die Anordnung und Größe der Lose. Die ersten beiden XML-Dateien werden in jedem Durchlauf von der Workflowkomponente geschrieben. Der Output-Produktionsplan ist hingegen vom Planungsverfahren anzulegen.

Der Ablauf zur Erzeugung der Planungsergebnisse ist in Abbildung 52 illustriert. Zu Beginn (Umgebung setzen) werden die für das Planungsverfahren korrekten Daten aus dem vorherigen Durchlauf (sofern nicht erste Iteration) gesammelt. Anschließend wird bei entsprechender Parametrisierung die Grobplanung gestartet. Sofern in dem Grobplanungshorizont zur Produktion eines Erzeugnisses eines Auftrags mehr als eine Makroperiode erforderlich ist, werden diese Teilmengen in Folge der rollierenden Planung gestückelt in den Feinplanungshorizont übergeben. Liegt die von der Grobplanung bestimmte Makroperiode vor dem realen Bedarfstermin, wird diesem Los als Bedarfstermin die letzte Mikroperiode der entsprechenden Makroperiode zugewiesen und auf dieser Basis in der Feinplanung geplant. Dies ist notwendig, um den jeweils gültigen Bereich der Feinplanung nicht zu verlassen, aber dennoch rechtzeitig die Produktion zu starten und damit Lagerbestände zur Bedarfsbefriedigung aufzubauen. Da jedoch in der Grobplanung eine feste Kapazität (Standardkapazität) unterstellt wird, wird mit diesem Vorgehen die Flexibilität in der Feinplanung eingeschränkt. Aus diesem Grund erfolgt beim Fortschreiten des Planungshorizontes für Teilmengen, die zu einem Auftrag gehören (diese werden über ein spezielles Attribut identifiziert), ein Merging, in dem der Bedarfstermin auf das Maximum dieser Teilmengen gesetzt wird. Aufgaben der Inputgenerierung sind zudem die Schichtmodellzuordnungen sowie Lose vorheriger Planungsläufe (sofern nicht erste Iteration) in die entsprechende XML-Datei (Input-Produktionsplan) zu überführen. Im Anschluss daran erfolgt die eigentliche Planungsverfahrensausführung. In Abhängigkeit gewählter Parametrisierung begrenzt sich dies auf ein Verfahren (z.B. dem mathematischen

⁸⁵ Die Lösung der mathematischen Modelle erfolgt mit ILOG CPLEX 12. Hierzu werden die Modelle inkl. der Daten mit Java erzeugt und an die Optimierungssoftware übergeben. Das Eröffnungs- und Verbesserungsverfahren wurden mit C#/.Net-Framework realisiert. Dies ist in dem besseren Laufzeitverhalten gegenüber Java begründet.

Modell der Feinplanung) oder einer verketteten Ausführung mehrerer Verfahren (z.B. dem Eröffnungs- und Tabu Search-Verfahren). Im letzten Fall wird dem zweiten Subverfahren zudem die Ausgabe des Eröffnungsverfahrens übergeben. Nach einem Planungslauf werden die Ergebnisse für die weitere Kriterienanalyse gespeichert.

Kriterienanalyse

Um einen Benchmark durchführen zu können, bedarf es geeigneter Kriterien zur Evaluierung der Produktionsplanungsverfahren. Analog zu der in Kapitel 3.3 vorgestellten Differenzierung (siehe auch Berens (vgl. [Bere92])) werden ressourcenorientierte und ergebnisorientierte Beurteilungskriterien betrachtet.

Ressourcenorientierte Kriterien:

- Rechenzeit (RZ): Wie in Kapitel 3.3 beschrieben, ist ein Vergleich spezifischer Werte wie z.B. der Speicherverbrauch oder die Auslastung des Prozessors während der Ausführung wenig zielführend. So stellen aufgrund der Entwicklung heutiger Computer diese Faktoren keine Begrenzung mehr dar bzw. sind relativ günstig erweiterbar. Relevant ist hingegen die reale Rechenzeit zum Auffinden einer Lösung.

Ergebnisorientierte Kriterien:

Die Bewertung von Planungsverfahren erfolgt in der Literatur häufig auf Basis von Planwerten. Es wird folglich ein Produktionsplan für einen definierten Horizont erstellt und für diesen die jeweiligen Ausprägungen der Kriterien ermittelt. Dieses Vorgehen kann jedoch nur in einem statischen Planungsumfeld verfolgt werden. In einem dynamischen Umfeld muss hingegen die Ermittlung der unterschiedlichen Größen auf Ist-Werten basieren – also rückwirkend über den realisierten Produktionsplan. Hieraus ergeben sich jedoch weitere Anforderungen an die Beurteilungskriterien. So müssen neben den kosten- und zeitorientierten Kriterien auch Stabilitätsorientierte Aspekte berücksichtigt werden. Letztere betrachten die Veränderung eines Plans über die Zeit. So hat die Stabilität eines Plans Auswirkungen auf die Arbeit anderer Unternehmensbereiche (z.B. der Werkslogistik oder dem Einkauf), deren Arbeit teils nicht resp. nur schwer quantifizierbar ist (so kann eine hohe Plannervosität zu einer großen Unzufriedenheit der Mitarbeiter führen). Daher werden die nachfolgenden Kriterien berücksichtigt:

- Produktionskosten (KP): Die Produktionskosten resultieren aus der zur Produktion erforderlichen Schichtmodellvariation.
- Lagerkosten (KL): Die Lagerkosten sind das Resultat aus der Menge an Erzeugnissen, die vor dem Bedarfstermin gefertigt wurden.
- Verzugskosten (KV): Verzugskosten werden für die Menge nicht fristgerecht gefertigter Erzeugnisse ermittelt.
- Gesamtkosten (KG): Dieses Kriterium stellt eine Summation der Produktions-, Lager- und Verzugskosten dar.
- Stabilität Schichtmodelle (SSM): Diese Kennzahl ermittelt, wie stabil die Planung der Schichtmodellzuordnungen im Rahmen des Planungshorizontes war. Dazu werden für jede Produktionslinie ($\forall pl \in PL$) die Anzahl an geänderten Schichtmodellzuordnungen ($\delta_{sm,pl}$) zu den maximal möglichen Schichtmodellwechseln (ergibt sich aus der Län-

ge des Feinplanungshorizontes ($T_{ps}^{e,FP}$), subtrahiert um den fixierten Schichthorizont ($T_{ps}^{e,fS}$)) ins Verhältnis gesetzt und anschließend für das gesamte Produktionssystem der entsprechende Mittelwert gebildet (siehe Formel (6.2)). Die initiale Schichtmodellzuordnung einer Makroperiode, gilt dabei nicht als Schichtmodellwechsel. Ein Wert von $SSM = 1$ entspricht dabei einer hohen Stabilität (keine Änderung), während ein Wert von $SSM = 0$ die geringste Stabilität kennzeichnet.

$$SSM = \frac{1}{|PL|} \sum_{pl \in PL} \left(1 - \frac{\delta_{sm,pl}}{T_{ps}^{e,FP} - T_{ps}^{e,fS}} \right) \quad (6.2)$$

- Stabilität Loszuordnung (SLZ): Diese Kennzahl ermittelt die Abweichung der Produktionszeiten (Start- und Endzeit) und Produktionslinienzuordnung von Losen des alten und des neuen Produktionsplans. Eine Normierung erfolgt über die gesamte Anzahl aller Lose.

Nach einem Verfahrensdurchlauf wird geprüft, ob die aktive Periode benchmarkrelevant ist (s.o.). In diesem Fall erfolgt in Abhängigkeit gewählter Parametrisierung die Bestimmung der Kriterienwerte für diese aktive Periode. Nach dem letzten Benchmarkinglauf werden zudem die Ausprägungen der Kriterien auf Basis der realisierten Abrufe ermittelt. Dadurch ist es möglich, die Auswirkungen der dynamischen Bedarfsentwicklung zu analysieren (es basiert z.B. die Berechnung der Schichtmodellstabilität auf dieser nachgeordneten Analyse).

6.1.3 Reportingkomponente

Die Datenaufbereitung der im Benchmarkingdesign ermittelten Werte und die Darstellung in graphischer und tabellarischer Form erfolgt in dieser Komponente. Diese besteht aus einem Übersichts-, Daten- und Statistikbrowser sowie zwei unterschiedlichen Gantt-Charts.

Übersichtsbrowser

Dieser Browser dient zur schnellen Navigation vorhandener Auswertungen. Es besteht die Möglichkeit, dynamisch eine kurze Zusammenfassung der wesentlichen Kriterien zu einzelnen Workflows zu erstellen. Hierzu dient eine Matrix, in der oben die verschiedenen Planungsverfahren und im linken Bereich die Kriterien abgetragen werden. Innerhalb der Matrix werden für jedes Kriterium neben den kumulierten Werten (insbesondere für die Kostenarten) auch das Minimum, der Mittelwert und das Maximum über die Perioden dargestellt. Durch diese Funktionalität können bei mehreren Workflows schnell entsprechende Unterschiede erkannt werden, die z.B. aus verschiedenen Parametrisierungen resultieren.

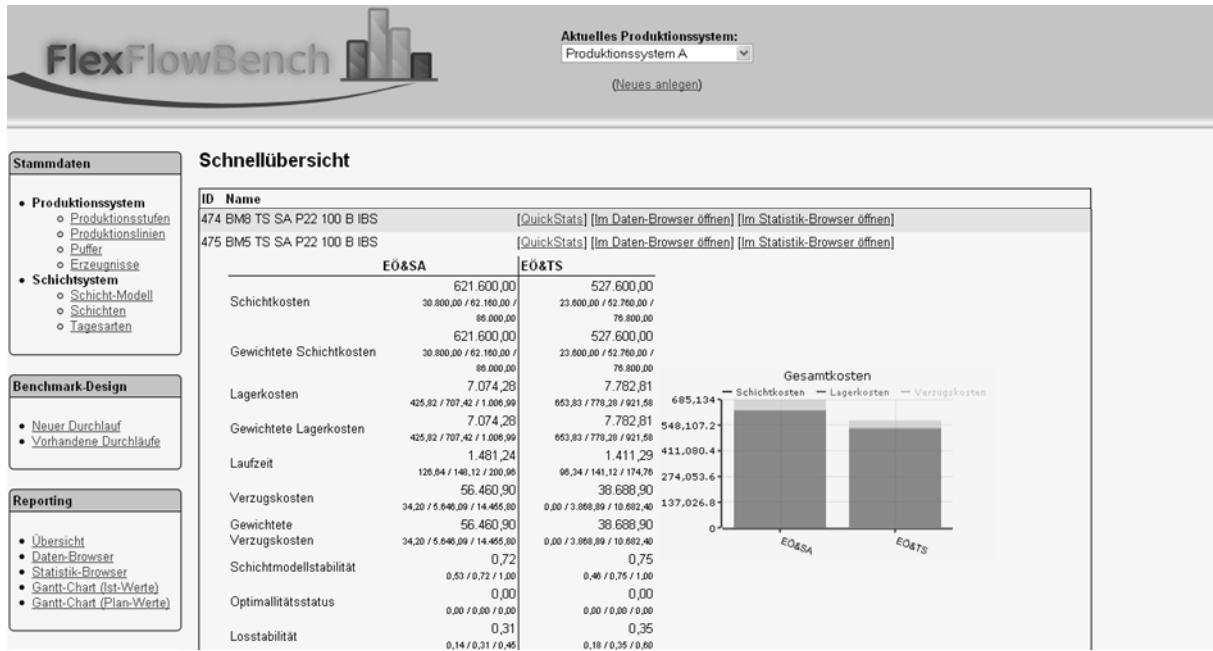


Abbildung 53: Übersichtsbrowser

Datenbrowser

Die komplette Kommunikation zwischen der Benchmarkingplattform und den Planungsverfahren erfolgt direkt über den Austausch von XML-Dateien (siehe Kapitel 6.1.2.2). Diese stellen daher umfang- und detailreiche Daten dar, die auch zur Nachvollziehbarkeit der Ergebnisse relevant sind. Hierzu kann eine periodenabhängige Auflistung der XML-Dateien der einzelnen Planungsverfahren angezeigt werden. In dieser sind die benchmarkingrelevanten Perioden gesondert markiert.

Statistikbrowser

Bei dem Vergleich der Planungsverfahren ist es möglich, dass nicht eines der Verfahren das andere vollständig dominiert, also für jedes Kriterium die optimale Wahl darstellt, sondern eventuell jedes Kriterium durch ein anderes Verfahren am besten erfüllt wird. Um diese nun detailliert vergleichen zu können, dient – als Kernelement der Reportingkomponente – dieser Browser. Er bietet eine transparente und konfigurierbare Zusammenfassung sämtlicher berechneter Kriterien eines Workflows an. Die Visualisierung der verschiedenen Werte wird dabei durch verschiedene Diagrammarten unterstützt. Mit diesem ist eine flexible Anordnung der Kennwerte und der entsprechenden Bezugspunkte möglich. So kann bspw. für ein Linendiagramm (siehe Abbildung 54) bestimmt werden, ob die Perioden oder die Planungsverfahren auf der X-Achse abgetragen werden sollen. Zudem existiert die Option „DrillDown“. So lässt sich eine dritte Informationsdimension anordnen, mit der ein „Eintauchen“ in die Daten ermöglicht wird. Grundsätzlich erfolgt die Werteübergabe an das Diagramm durch Kumulation aller Werte, die nicht Gegenstand einer sichtbaren Zuordnung sind. Wurde als X-Achse bspw. die Dimension „Verfahren“ gewählt, so erfolgt für jedes Planungsverfahren eine Summation der jeweiligen Zielgrößen über alle Perioden. Der damit verbundene Informationsverlust kann durch Angabe der DrillDown-Dimension „Perioden“ vermieden werden, wodurch neben der kumulierten Anzeige auch die Anzeige der entsprechenden Zielgrößen innerhalb

der einzelnen Perioden ermöglicht wird. Neben der optischen Darstellung werden die jeweiligen Daten auch tabellarisch mit demselben Informationsgehalt visualisiert.

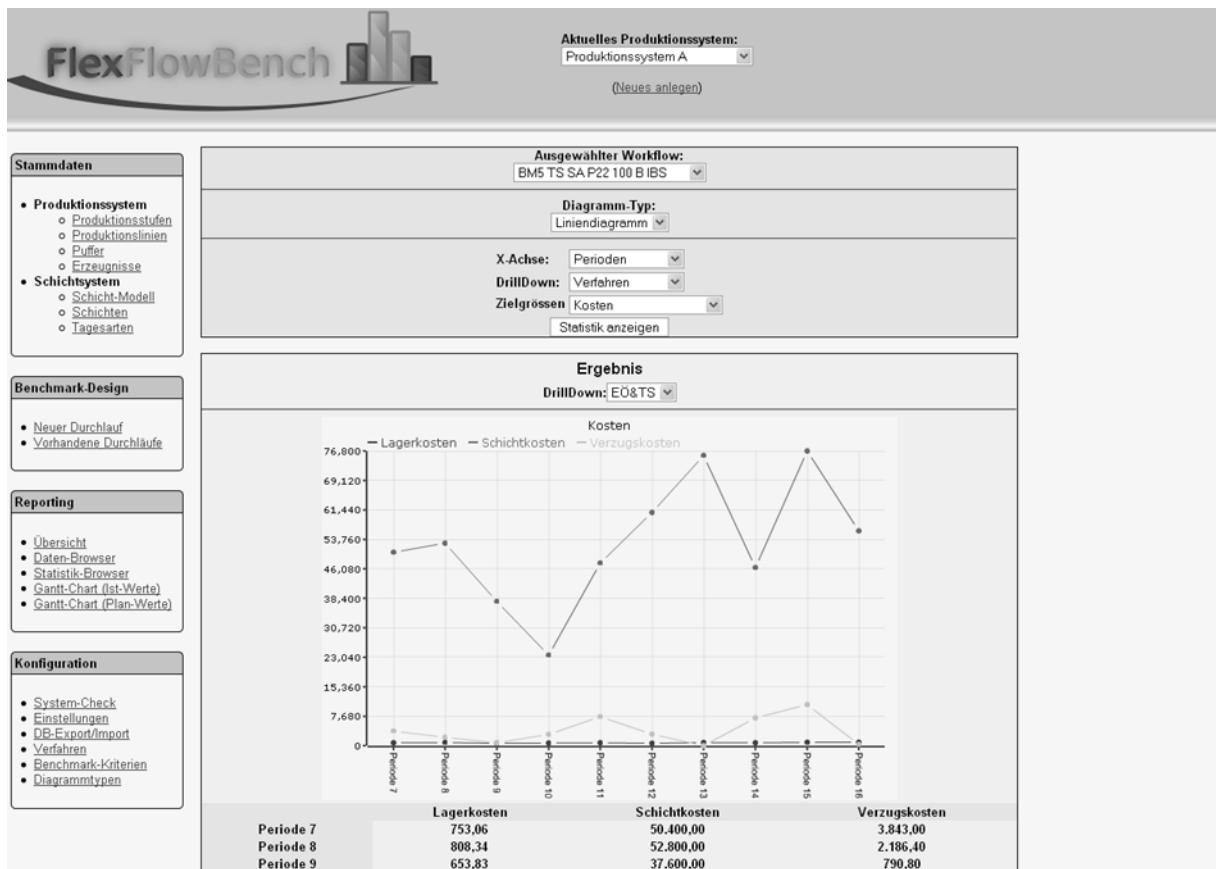


Abbildung 54: Statistikbrowser mit Liniendiagramm

Gantt-Chart

Zur Visualisierung der durch die Planungsverfahren generierten Werte dienen zwei unterschiedliche Arten von Gantt-Charts. Diese unterscheiden sich generell in der zeitlichen Be trachtung des Planungshorizontes. So werden mit dem „Gantt-Chart (Plan-Werte)“ (siehe Abbildung 55 für ein Beispiel mit einer Bucketgröße von 8 Stunden) jeweils die Planungsergebnisse eines Planungslaufs zu einem bestimmten Zeitpunkt – d.h., auf Basis der zu diesem Zeitpunkt vorliegenden Bedarfe – dargestellt. Zudem existiert das „Gantt-Chart (Ist-Werte)“. Dieses zeigt nach der Beendigung eines kompletten Benchmarkingdurchlaufs für den gesamten Planungshorizont die wirklich eingetretene Bedarfssituation. Um eine bessere Lesbarkeit zu gewährleisten, werden die einzelnen Lose und Schichten in unterschiedlichen Farben visualisiert. Diese können über die Stammdatenkomponente verwaltet werden (siehe auch Abbildung 46).

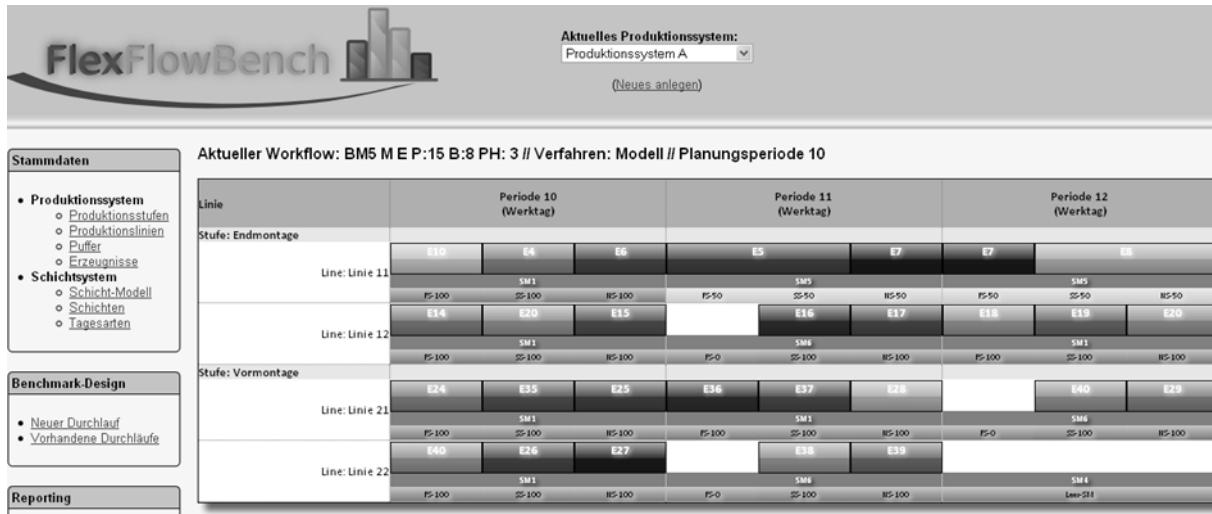


Abbildung 55: Gantt-Chart (Plan-Werte)

6.1.4 Konfigurationskomponente

Die Konfigurationskomponente ist nicht integraler Bestandteil der Benchmarkingplattform, sondern steht komplementär zu den bereits vorgestellten Komponenten und dient der Administration der eigentlichen Applikation. Die Beschreibung dieser erfolgt kurz aus Gründen der Vollständigkeit. Für den erfolgreichen Betrieb der Applikation sind einige Einstellungen erforderlich, die bei der Installation und ggf. auch im Betrieb zu konfigurieren sind. Hierzu gibt es spezielle Masken, um die Verbindungsparameter an die unterschiedlichen Verfahren sowie Pfadangaben anpassen zu können. Zur Validierung dieser existiert eine Funktion, die die getätigten Einstellungen überprüft (ggf. Fehler benennt) und somit ein funktionierendes System sicherstellt. Der gesamte Datenbestand der Applikation befindet sich in einer Datenbank. Zur Sicherung des gesamten Datenbestands existiert eine entsprechende Backup-Funktion.

6.2 Experimentelle Untersuchung

In diesem Kapitel werden unter Verwendung der in 6.1 beschriebenen Benchmarkingplattform die dargestellten Planungsverfahren untersucht. Hierzu werden generell zwei unterschiedliche Arten von Simulationsstudien durchgeführt.⁸⁶ Die erste Studie dient zum direkten Vergleich des mathematischen Modells (siehe Kapitel 5.1) und des heuristischen Verfahrens⁸⁷ (siehe Kapitel 5.2). Um in angemessener Zeit mit dem mathematischen Modell eine Lösung zu erhalten, wird in dieser Studie ein Anwendungsszenario von geringer Komplexität unterstellt. Ziel der Untersuchung ist generell die Funktionsfähigkeit der Verfahren zu zeigen so-

⁸⁶ Die Test-Umgebung stellt ein 64bit-System mit einem Vierkernprozessor (je 2,67 GHz) und 8 GB RAM dar. Zur Lösung der Optimierungsmodelle fand ILOG CPLEX in der Version 12 Anwendung. Die genutzte Konfiguration von ILOG CPLEX entspricht den Standardeinstellungen, abgesehen von dem Parameter „probe“ mit dem Wert 2. Zudem erfolgt die Threadbelegung opportunistisch. Dies reduziert die Laufzeiten signifikant.

⁸⁷ Innerhalb des heuristischen Verfahrens wird das Eröffnungsverfahren sowohl in Kombination mit dem Verbesserungsverfahren Tabu Search als auch dem Simulated Annealing angewendet. Wie in Kapitel 3.2.2 diskutiert, wird die Performance der Verfahren stark durch die bestimmten Parameter beeinflusst. In dieser Arbeit werden die Standardparameter unterstellt. Für eine Analyse der jeweiligen Daten sei auf Anhang C verwiesen.

wie die Abweichung der heuristischen von der optimalen Lösung zu bestimmen. Zur Analyse von komplexeren Planungsproblemen und der Einflüsse der einzelnen Subverfahren auf die gesamte Lösung dient hingegen die zweite Studie.

6.2.1 Anwendungsszenario 1

In diesem Szenario (AZ 1) wird ein einfaches Planungsproblem betrachtet. Dieses besteht aus einem Produktionssystem mit 2 Produktionsstufen (ps_1, ps_2). Die erste Produktionsstufe (z.B. Endmontage) verfügt über 2 verschiedenartige parallele Produktionslinien (pl_{11}, pl_{12}), die zweite Produktionsstufe (z.B. Vormontage) über 2 identische parallele Produktionslinien (pl_{21}, pl_{22}). Auf diesem Produktionssystem sind 20 verschiedene Enderzeugnisse ($E_1 \dots E_{20}$) zu produzieren. Die Erzeugnisse $E_1 \dots E_{10}$ können auf der Produktionslinie pl_{11} und die Erzeugnisse $E_{11} \dots E_{20}$ auf Produktionslinie pl_{12} hergestellt werden. Infolge der identischen Produktionslinien auf der zweiten Produktionsstufe ist keine Zuordnung erforderlich. Um das Beispiel einfach zu halten, werden für die Erzeugnisse gleiche Lager- und Verzugskosten sowie Taktzeiten unterstellt. Der Feinplanungshorizont beträgt 4 Makroperioden, wobei jede dieser Makroperioden aus 3 Mikroperioden besteht⁸⁸. Zur Analyse der rollierenden Planung werden 10 benchmarkingrelevante Perioden (siehe Formel (6.1)) betrachtet. Zu Beginn der Untersuchung existieren keine Verzugsmengen sowie Initiallagerbestände.

Dieses Planungssystem ist Basis primär zweier verschiedener Untersuchungen. In der ersten Untersuchung werden die Auswirkungen der flexiblen Zuordnung von Schichtmodellen analysiert. Hierzu kann auf der ps_1 aus 9 verschiedenen Schichtmodellen ($sm_1 \dots sm_9$) und auf der ps_2 aus 3 verschiedenen Schichtmodellen ein jeweils kostenminimales gewählt werden. Die beschränkte Auswahl der zweiten Produktionsstufe ist dadurch begründet, dass keine unterschiedlichen Leistungsgrade gewählt werden können. In der zweiten Untersuchung werden im Gegensatz zur ersten Studie für die Produktionsstufe 1 reihenfolgeabhängige Rüstzeiten unterstellt. Zur Vermeidung zu langer Rechenzeiten wird die Auswahlmöglichkeit an Schichtmodellen in dieser Untersuchung auf zwei verschiedene je Produktionsstufe beschränkt.

⁸⁸ Da mit zunehmender Anzahl von Mikroperioden pro Makroperiode die Zeit zur Lösung des mathematischen Modells steigt, werden die Bucketgröße und sonstigen Werte (Taktzeiten, etc.) so bestimmt, dass die Szenarien generell jeweils in einer Stunde lösbar sind (die Anzahl der in dieser Zeit vollständig lösbar Probleme wird durch den Opt.Status ausgewiesen). Dies erlaubt eine größere Anzahl unterschiedlicher Analysen in kurzer Zeit. Da die Bucketgröße jedoch nur für das mathematische Modell relevant ist, müssen zur Gewährleistung der Vergleichbarkeit der Ergebnisse der unterschiedlichen Verfahren die Bedarfstermine mit dem Start eines Buckets übereinstimmen. Da jeder Analyse nur eine einzige Ausprägung der betrachteten Daten zugrunde liegt, hängt die Genauigkeit der Ergebnisse von der Anzahl der Simulationsläufe ab. Aus diesem Grund werden je Szenario 3 verschiedene Untersuchungen vorgenommen. Deren Ergebnisse werden im Folgenden einzeln und zusammengefasst (\bar{x}) dargestellt. Für jeden Simulationslauf wurde zufällig eine neue Instanz des Problems erzeugt. Die Art der Bedarfsentwicklung kann zudem große Auswirkungen auf das Planungsergebnis haben. Daher werden für jeden Vergleich Untersuchungen ohne und mit Bedarfsveränderungen (BV) vorgenommen.

Untersuchung: Flexibilität in der Schichtmodellauswahl

Die Tabelle 3 und Tabelle 4 zeigen das Ergebnis der ersten Teiluntersuchung. Zur Bewertung der Verfahren werden zeitorientierte und monetäre Kriterien sowie die eingeführten Stabilitätskriterien verwendet. Für die monetären Kriterien werden sowohl die errechneten Kosten (Schichtkosten (SK), Lagerkosten (LK), Verzugskosten (VK), Gesamtkosten (GK)), als auch für die Heuristiken die Lösungsqualität r angegeben. In diesem Vergleich wird deutlich, dass die Heuristiken im Vergleich zu dem Modell der Feinplanung gute Ergebnisse erzielen. Die Gesamtkosten sind nicht signifikant höher als die des Modells. Außerdem sind die Rechenzeiten der Heuristiken um ein Vielfaches geringer und die Stabilitätskriterien weisen bessere Werte auf.

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.) (Opt.Status)	SSM	SLZ
1	Modell der Feinplanung	463.200,00 3.110,40 34.560,00 500.870,40		12.040,04 (8)	0,63	0,43
	Heuristiken der Feinplanung	474.400,00 2.054,40 57.600,00 534.054,40	1,02 0,66 1,67 1,07	72,62	0,74	0,68
2	Modell der Feinplanung	318.800,00 2.073,60 0,00 320.873,60		3.575,37 (10)	0,65	0,39
	Heuristiken der Feinplanung	348.800,00 1.632,00 0,00 350.432,00	1,09 0,79 0 1,09	42,33	0,8	0,71
3	Modell der Feinplanung	266.400,00 3.907,20 0,00 270.307,20		3.941,29 (10)	0,8	0,34
	Heuristiken der Feinplanung	292.800,00 1.555,20 0,00 294.355,20	1,1 0,4 0 1,09	359,52	0,87	0,73
\bar{x}	Modell der Feinplanung	349.466,67 3.030,40 11.520,00 364.017,07		6.518,9	0,69	0,39
	Heuristiken der Feinplanung	372.000,00 1.747,20 19.200,00 392.947,20	1,06 0,58 1,67 1,08	158,16	0,8	0,71

Tabelle 3: Ergebnis AZ 1: Schichtmodellflexibilität (keine BV)

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.) (Opt.Status)	SSM	SLZ
1	Modell der Feinplanung	300.000,00 3.129,60 0,00 303.129,60		5.543,81 (10)	0,75	0,45
	Heuristiken der Feinplanung	339.200,00 1.708,80 0,00 340.908,80	1,13 0,55 0 1,12	38,64	0,85	0,68
2	Modell der Feinplanung	316.000,00 4.416,00 0,00 320.416,00		6.259,91 (10)	0,73	0,31
	Heuristiken der Feinplanung	354.400,00 2.169,60 0,00 356.569,60	1,12 0,49 0 1,11	45,17	0,8	0,61
3	Modell der Feinplanung	352.000,00 4.012,80 0,00 356.012,80		12.775,64 (7)	0,7	0,31
	Heuristiken der Feinplanung	377.600,00 2.035,20 0,00 379.635,20	1,07 0,51 0 1,07	55,73	0,81	0,65
\bar{x}	Modell der Feinplanung	322.666,67 3.852,80 0,00 326.519,47		8.193,12	0,73	0,36
	Heuristiken der Feinplanung	357.066,67 1.971,20 0,00 359.037,87	1,11 0,51 0 1,1	46,51	0,82	0,65

Tabelle 4: Ergebnis AZ 1: Schichtmodellflexibilität (mit BV)

Untersuchung: produktionslinien- und reihenfolgeabhängige Rüstzeiten

Im Stand der Technik wurde deutlich, dass besonders die Integration von produktionslinien- und reihenfolgeabhängigen Rüstzeiten einen starken Zuwachs an Komplexität und damit Rechenzeit zur Lösung des Problems bewirkt. Die Ergebnisse dieser Teiluntersuchung werden in Tabelle 5 und Tabelle 6 zusammengefasst. Es wird analog zu der vorherigen Untersuchung deutlich, dass die Heuristiken gute Ergebnisse erzielen.

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.) (Opt.Status)	SSM	SLZ
1	Modell der Feinplanung	368.000,00 2.784,00 5.760,00 376.544,00	/	2.875,86 (10)	0,77	0,4
	Heuristiken der Feinplanung	368.000,00 2.323,20 34.560,00 404.883,20		1 0,83 6 1,08	53,7	0,86 0,59
2	Modell der Feinplanung	349.600,00 1.958,40 11.520,00 363.078,40	/	1.669,39 (10)	0,79	0,25
	Heuristiken der Feinplanung	368.000,00 2.457,60 28.800,00 399.257,60		1,05 1,25 2,5 1,1	34,17	0,85 0,49
3	Modell der Feinplanung	331.200,00 2.515,20 5.760,00 339.475,20	/	1.565,11 (10)	0,8	0,45
	Heuristiken der Feinplanung	349.600,00 2.342,40 23.039,80 374.982,20		1,06 0,93 4 1,1	37,1	0,89 0,6
\bar{x}	Modell der Feinplanung	349.600,00 2.419,20 7.680,00 359.699,20	/	2.036,79	0,79	0,37
	Heuristiken der Feinplanung	361.866,67 2.374,40 28.799,93 393.041,00		1,04 0,98 3,75 1,09	41,66	0,87 0,56

Tabelle 5: Ergebnis AZ 1: Rüstzeitintegration (keine BV)

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.) (Opt.Status)	SSM	SLZ
1	Modell der Feinplanung	423.000,00 3.148,80 5.760,00 431.908,80		7.514,73 (9)	0,75	0,37
	Heuristiken der Feinplanung	423.200,00 2.860,80 46.080,00 472.140,80	1 0,91 8 1,09	89,03	0,83	0,57
2	Modell der Feinplanung	368.000,00 2.553,60 5.760,00 376.313,60		9.062,08 (9)	0,77	0,4
	Heuristiken der Feinplanung	404.800,00 2.457,60 28.800,00 436.057,60	1,1 0,96 5 1,16	42,68	0,87	0,52
3	Modell der Feinplanung	312.800,00 1.382,40 0,00 314.182,40		3.614,47 (10)	0,85	0,5
	Heuristiken der Feinplanung	312.800,00 1.459,20 0,00 314.259,20	1 1,06 0 1	29,82	0,9	0,63
\bar{x}	Modell der Feinplanung	367.933,33 2.361,60 3.840,00 374.134,93		6.730,43	0,79	0,42
	Heuristiken der Feinplanung	380.266,67 2.259,20 24.960,00 407.485,87	1,03 0,96 6,5 1,09	53,84	0,87	0,57

Tabelle 6: Ergebnis AZ 1: Rüstzeitintegration (mit BV)

Anhand dieser beiden Teiluntersuchungen ist zu konstatieren, dass es aufgrund der enormen Komplexität nicht möglich ist, das mathematische Modell unter praxisrelevanten Bedingungen optimal zu lösen und mit der durch die Heuristik generierten Lösung zu vergleichen. Zudem ist keine optimale Lösung zu einem vergleichbaren Problem bekannt, welches zur Bewertung genutzt werden kann. Aus diesem Grund ist nur eine Dekomposition des Problems in mehrere Teilprobleme möglich. In diesen Untersuchungen hat sich gezeigt, dass die Heuristiken eine hohe Lösungsqualität in einem Bruchteil der erforderlichen Zeit des mathematischen Modells erzielen. Im Folgenden wird analysiert, wie sich das Planungsverfahren bei einem komplexeren Planungsproblem verhält. Zudem wird die Effektivität der einzelnen Teilverfahren ermittelt.

6.2.2 Anwendungsszenario 2

In diesem Szenario (AZ 2) werden 10 unterschiedliche Enderzeugnisse ($E_1 \dots E_{10}$) in einem dreistufigen Produktionsprozess (siehe Abbildung 56) gefertigt.⁸⁹ Zu Anfang dieses Prozesses werden die Grunderzeugnisse hergestellt. Auf dieser Produktionsstufe (ps_3) existieren 3 verschiedene Produktionslinien ($pl_{31}, pl_{32}, pl_{33}$), von denen jedes Erzeugnis genau eine Produktionslinie durchlaufen muss. In Abhängigkeit des Erzeugnisses besitzen die Produktionslinien unterschiedliche Eignung. Im Anschluss daran durchlaufen die Erzeugnisse die zweite Produktionsstufe (ps_2), welche aus 2 Produktionslinien (pl_{21}, pl_{22}) besteht. Aufgrund aufwändiger Werkzeugwechsel und Reinigungszeiten gelten die Produktionsstufen ps_2 und ps_3 als rüstzeitintensiv. Die erste Produktionsstufe (ps_1) besteht aus 2 Produktionslinien (pl_{11}, pl_{12}) mit jeweils 4 identisch ausgestatteten Bearbeitungsplätzen. Die Vorerzeugnisse werden jeweils bedarfsoorientiert gefertigt.⁹⁰ Die drei Produktionsstufen befinden sich am selben Standort, der Transport zwischen ihnen wird durch eine Vorlaufzeit (60 Min.) berücksichtigt. Zudem existieren kapazitierte Puffer (pu_1, pu_2, pu_3) zur Zwischenlagerung der Erzeugnisse.

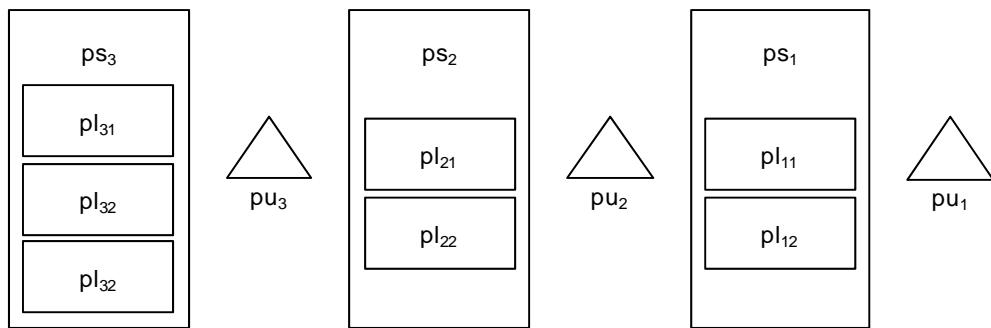


Abbildung 56: Aufbau von Anwendungsszenario 2

Für jede dieser Produktionsstufen gibt es unterschiedliche Schichtmodelle. So kann auf den Produktionsstufen ps_3 und ps_2 aus 8 verschiedenen Schichtmodellen ($sm_1 \dots sm_8$) gewählt werden. Eine Besonderheit stellt die Produktionsstufe 1 dar. In dieser kann nicht nur eine Variation der Zeiten erfolgen, sondern auch der Leistungsgrade, wodurch insgesamt 21 Schichtmodelle ($sm_1 \dots sm_{21}$) zur Verfügung stehen. Das Unternehmen hat einen Feinplanungshorizont von 7 Makroperioden. Die eingefrorene Zone umfasst 1 Makroperiode, die des fixierten Schichthorizontes 2. Analog zu dem ersten Szenario sind 10 Makroperioden benchmarking-relevant.

Die Ergebnisse dieser Untersuchung sind entsprechend der Veränderung der Bedarfe in Tabelle 7 und Tabelle 8 dargestellt. Analog zu dem Anwendungsszenario 1 werden zeitorientierte und monetäre Kriterien sowie die eingeführten Stabilitätskriterien verwendet. Außerdem wird für die Verbesserungsverfahren die Lösungsqualität r in Bezug zu den Ergebnissen des Eröffnungsverfahrens ermittelt. Die Ergebnisse verdeutlichen, dass keines der Verbesserungsverfahren die optimale Lösung findet.

⁸⁹ Für eine Analyse der jeweiligen Daten sei auf Anhang C verwiesen.

⁹⁰ Die erforderlichen Kleinteile (Schrauben, Farbe, etc.) stellen für das Unternehmen C-Teile dar und bedürfen daher keiner weiteren Betrachtung. Diese Teile werden nach der Kanban-Methode disponiert. Die Verfügbarkeit ist immer gewährleistet.

rungsverfahren das andere vollständig dominiert. Beide Verfahren können zu einer Verbesserung eines Ausgangsplans beitragen. Es wird jedoch auch deutlich, dass Situationen auftreten können, in denen eine Verbesserung nicht mehr möglich ist. D.h., dass das Eröffnungsverfahren bereits einen guten Plan ermittelt hat.

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.)	SSM	SLZ
1	Eröffnungsverfahren	473.200,00 4.369,87 37.166,70 514.736,57		96,37	0,78	0,55
	Eröffnungsverfahren & Simulated Annealing	472.400,00 4.469,55 13.663,80 490.533,35	1,00 1,02 0,37 0,95	1.340,4	0,73	0,42
	Eröffnungsverfahren & Tabu Search	444.000,00 4.144,36 8.339,40 456.483,76	0,94 0,95 0,22 0,89	1.233,06	0,78	0,57
2	Eröffnungsverfahren	548.000,00 6.221,68 73.334,70 627.556,38		53,16	0,8	0,46
	Eröffnungsverfahren & Simulated Annealing	564.800,00 5.663,56 6.319,08 576.782,64	1,03 0,91 0,09 0,92	1.431,83	0,74	0,38
	Eröffnungsverfahren & Tabu Search	567.200,00 6.198,45 20.129,40 593.527,85	1,04 1,00 0,27 0,95	1.287,96	0,75	0,42
3	Eröffnungsverfahren	509.600,00 5.339,03 1.265,40 516.204,43		51,36	0,82	0,57
	Eröffnungsverfahren & Simulated Annealing	527.200,00 5.618,20 10.243,80 543.062,00	1,03 1,05 8,10 1,05	1.313,68	0,77	0,45
	Eröffnungsverfahren & Tabu Search	506.800,00 6.296,13 6.669,90 519.766,03	0,99 1,18 5,27 1,01	1.275,26	0,81	0,45

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.)	SSM	SLZ
\bar{x}	Eröffnungsverfahren	510.266,67 5.310,19 37.255,60 552.832,46		66,96	0,8	0,53
	Eröffnungsverfahren & Simulated Annealing	521.466,67 5.250,44 10.075,56 536.792,66	1,02 0,99 0,27 0,97	1.361,97	0,75	0,42
	Eröffnungsverfahren & Tabu Search	506.000,00 5.546,31 11.712,90 523.259,21	0,99 1,04 0,31 0,95	1.265,43	0,78	0,48

Tabelle 7: Ergebnis AZ 2: Vergleich der Feinplanungsverfahren (keine BV)

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.)	SSM	SLZ
1	Eröffnungsverfahren	546.000,00 8.353,13 58.724,25 613.077,38		323,49	0,78	0,34
	Eröffnungsverfahren & Simulated Annealing	470.800,00 8.682,74 37.852,50 517.335,24	0,86 1,04 0,64 0,84	1.460,62	0,74	0,32
	Eröffnungsverfahren & Tabu Search	480.800,00 7.873,26 25.341,60 514.014,86	0,88 0,94 0,43 0,84	1.478,44	0,77	0,39
2	Eröffnungsverfahren	550.800,00 9.084,73 59.435,70 619.320,43		226,55	0,76	0,41
	Eröffnungsverfahren & Simulated Annealing	621.600,00 7.074,28 56.460,90 685.135,18	1,13 0,78 0,95 1,11	1.481,24	0,72	0,31
	Eröffnungsverfahren & Tabu Search	527.600,00 7.782,81 38.688,90 574.071,71	0,96 0,86 0,65 0,93	1.411,29	0,75	0,35

Workflow	Verfahren	SK (€) LK (€) VK (€) GK (€)	SK (r) LK (r) VK (r) GK (r)	RZ (Sek.)	SSM	SLZ
3	Eröffnungsverfahren	524.000,00 6.312,52 83.897,47 614.209,99		204,76	0,79	0,33
	Eröffnungsverfahren & Simulated Annealing	502.000,00 5.340,41 15.960,60 523.301,01	0,96 0,85 0,19 0,85	1.390,31	0,77	0,34
	Eröffnungsverfahren & Tabu Search	504.400,00 6.417,83 25.167,90 535.985,73	0,96 1,02 0,30 0,87	1.461,29	0,77	0,42
\bar{x}	Eröffnungsverfahren	540.266,67 7.916,79 67.352,47 615.535,93		251,6	0,78	0,36
	Eröffnungsverfahren & Simulated Annealing	531.466,67 7.032,48 36.758,00 575.257,14	0,98 0,89 0,55 0,93	1.444,06	0,74	0,32
	Eröffnungsverfahren & Tabu Search	504.266,67 7.357,97 29.732,80 541.357,43	0,93 0,93 0,44 0,88	1.450,34	0,76	0,39

Tabelle 8: Ergebnis AZ 2: Vergleich der Feinplanungsverfahren (mit BV)

Abschließend wird das oben eingeführte Anwendungsszenario beispielhaft für einen Workflow und eine dynamische Planungsumgebung um einen Grobplanungsbereich von 14 Makroperioden erweitert. Die Ergebnisse dieser Untersuchung sind in Tabelle 9 dargestellt⁹¹. Es wird deutlich, dass die Grobplanung nur einen geringen Rechenzeitbedarf hat.

Verfahren	SK (€) LK (€) VK (€) GK (€)	RZ (Feinplanung) (Sek.)	RZ (Grobplanung) (Sek.)	SSM	SLZ
Eröffnungsverfahren & Simulated Annealing & Grobplanung	278.800,00 7.896,60 2.001,60 288.698,20	1.296,47	41,58	0,84	0,53
Eröffnungsverfahren & Tabu Search & Grobplanung	254.800,00 8.438,28 4.168,80 267.407,08	1.019,15	43,39	0,86	0,65

Tabelle 9: Ergebnis AZ 2: Hinzunahme der Verfahren der Grobplanung

⁹¹ Zur Lösung des mathematischen Modells der Grobplanung wurden die Parametereinstellungen des Modells der Feinplanung um einen mipgap von 0,1 erweitert.

Es ist zu konstatieren, dass die Heuristiken auch bei einer großen Anzahl an Bedarfen und Perioden gute Ergebnisse in kurzer Zeit ermitteln. Anhand der Ausprägungen der Stabilitätskriterien ist ersichtlich, dass die jeweiligen Bedarfsveränderungen gut in einen bestehenden Plan eingearbeitet werden können. Zudem zeigt sich, dass nicht eines der Verbesserungsverfahren das andere vollständig dominiert. Vielmehr liefern beide situationsabhängig gute Ergebnisse. Deren Einsatz hat also in Abhängigkeit des konkreten Problems zu erfolgen (z.B. Häufigkeit von Bedarfsveränderungen, Länge der Planungshorizonte, etc.). Schlussendlich kann festgestellt werden, dass auch die Existenz der Grobplanungsverfahren berechtigt ist.

7 Zusammenfassung und Ausblick

*„Wenn ein Reisender nach Hause zurückkehrt,
soll er nicht die Bräuche seiner Heimat eintauschen
gegen die des fremden Landes. Nur einige Blumen von dem,
was er in der Ferne gelernt hat, soll er in die Gewohnheiten
seines eigenen Landes einpflanzen.“*

(Francis Bacon)

7.1 Ergebnis der Arbeit

In der realen Unternehmenswelt lässt sich ein Flexible Flow Shop häufig direkt oder mit kleineren Abweichungen vorfinden und stellt somit einen wesentlichen Anteil praxisüblicher Probleme dar. Aus diesem Grund dient diese Problemklasse auch als Basis für die in dieser Arbeit entwickelten Verfahren. Um möglichst viele praxisrelevante Aspekte abbilden zu können, wurde die klassische Definition des Flexible Flow Shop erweitert. Dies umfasst bspw. verschiedenartige parallele Produktionslinien, beschränkte Pufferkapazitäten, produktionslinien- und reihenfolgeabhängige Rüstzeiten sowie eine flexible Allokation von Schichtmodellen.

Aus mathematischer Sicht können diese Probleme bereits seit Jahren modelliert und theoretisch gelöst werden. Praktisch scheitert dies jedoch an der Komplexität des Planungsprozesses. So existiert ein entscheidender Zielkonflikt. Damit ein möglichst gutes Ergebnis erreicht werden kann, muss das Modell sehr genau sein. Ein detailliertes und damit umfangreiches Modell verursacht wiederum nicht zu bewältigenden Rechenaufwand. Aus diesem Grund wurde ein heuristisches Verfahren entwickelt, das zwar keine Optimalität garantieren kann, aber im Allgemeinen eine relativ gute Lösung des Problems in kurzer Zeit liefert. Dieses Vorgehen erscheint auch ausreichend, da das Optimum in der Realität ohnehin unerreichbar bleibt.

Um die mit dem Planungsproblem verbundenen unterschiedlichen Probleme effektiv zu lösen, wurden mehrere Teilverfahren entwickelt, deren Aufbau und Ablauf durch eine Reihe von Parametern beeinflusst werden können. In Abhängigkeit des Detaillierungsgrads der einzelnen Informationen und des zeitlichen Horizontes erfolgte eine erste Unterteilung in Grob- und Feinplanungsverfahren. Das Ziel der Grobplanungsverfahren besteht in der Betrachtung eines längeren Planungshorizontes und einer damit insgesamt vorausschauenden und sicheren Planung. So können auf dieser Basis frühzeitig Maßnahmen zur Vermeidung von Engpässen ergriffen werden. Der Fokus dieser Arbeit liegt jedoch auf den Feinplanungsverfahren, die der Planung eines vergleichsweise kurzen Planungshorizontes dienen. Zur Planung dieses Bereichs werden zwei unterschiedliche Verfahrenstypen verwendet. Ein Eröffnungsverfahren verfolgt das Ziel, die mit dem dynamischen Planungsumfeld verbundenen Unsicherheiten so in einen bestehenden Plan einzuarbeiten, dass eine erste gültige Lösung unter Berücksichtigung der Interdependenzen auf und zwischen den Produktionsstufen resultiert. Diese fungiert als Basis des Verbesserungsverfahrens, welches aus zwei nachbarschaftsbasierten metaheuris-

tischen Verfahren besteht. In diesen wird bis zum Erreichen der Abbruchbedingung in jeder Iteration auf Grundlage der beiden Faktoren „Kosten“ und „Flexibilität“ ein Basislos bestimmt, welches an einer besten als gültig definierten Position eingeplant und in Abhängigkeit der jeweiligen Metaheuristik übernommen wird.

Entscheidend für den Einsatz eines Planungsverfahrens in einem Unternehmen ist die Kenntnis, wie gut die Planungsergebnisse unter bestimmten Bedingungen sind. Um dies evaluieren zu können, wurde eine Benchmarkingplattform erstellt. Diese erlaubt vor dem Hintergrund eines rollierenden Planungshorizontes anhand verschiedener Kriterien einen direkten Vergleich der einzelnen Verfahren untereinander sowie eine transparente Darstellung der Ergebnisse. Mittels dieser Benchmarkingkomponente wurden auf Grundlage zweier unterschiedlich komplexer Anwendungsszenarien die entwickelten Verfahren evaluiert. Das erste Anwendungsszenario ist von geringer Komplexität und dient als Basis zum direkten Vergleich der Planungsheuristiken mit den Ergebnissen eines mathematischen Modells. In diesem konnte nachgewiesen werden, dass sich die Ergebnisse nicht signifikant voneinander unterscheiden, die Lösungszeit des heuristischen Vorgehens indes um ein Vielfaches geringer ist. Das zweite Anwendungsszenario diente zur Simulation der Planung anhand praxisrelevanter Instanzen. Es konnte gezeigt werden, dass unter den verschiedenen Bedingungen die Laufzeiten immer noch akzeptabel sind. Zudem wurde dargestellt, dass die einzelnen Teilverfahren ihre Existenzberechtigung haben und unter jeweils verschiedenen Bedingungen gute Planungsergebnisse resultieren.

7.2 Ausblick

Die Ergebnisse dieser Arbeit können als Basis für weitere Entwicklungen dienen. Die in Kapitel 6.1 beschriebene Implementierung diente bislang primär der Validierung und Evaluation der Planungsverfahren. In einem nächsten Schritt kann diese Implementierung die Grundlage zur Integration in betriebliche ERP-/PPS-Systeme bilden. Es wurde zudem deutlich, dass die Parametrisierung starken Einfluss auf die Lösungsqualität bzw. -ermittlung besitzen kann. Vor diesem Hintergrund könnte ein Vorgehen erarbeitet werden, dass automatisch für ein Unternehmen diese Werte bestimmt. Dies bezieht sich sowohl auf die Länge der einzelnen Planungsbereiche (dem Fein- und Grobplanungshorizont) als auch auf die planungsverfahrensspezifischen Parameter.

Eine weitere Möglichkeit ist die Erweiterung des Verfahrens um die Lösung zusätzlicher Aufgaben. So wurde in dieser Arbeit von einer begrenzten Verfügbarkeit der Arbeitskräfte abstrahiert. Der ausreichende Umfang wird entweder durch die strategische Planung gewährleistet oder es besteht die Möglichkeit, diesen kurzfristig durch z.B. Personaldienstleister zu erhöhen. Außerdem werden keine Qualifikationen der einzelnen Arbeitskräfte berücksichtigt. In einem weiteren Schritt könnte das in dieser Arbeit vorgestellte Verfahren um entsprechende Annahmen erweitert werden. Hierbei wären sowohl gesetzliche (z.B. Ruhezeiten zwischen zwei Schichten einer Arbeitskraft) als auch geltende tarifvertragliche Bestimmungen (z.B. eine maximale Anzahl an Arbeitsstunden in einer Woche) zu integrieren.

Der bisherige Ansatz fokussiert die Planung innerhalb eines Produktionsunternehmens. In diesem wird von der begrenzten Verfügbarkeit notwendiger Roh- oder Hilfsstoffe abstrahiert. Es wird vielmehr unterstellt, dass diese zur geplanten Fertigung zur Verfügung stehen, also vorrätig sind bzw. durch den Lieferanten fristgerecht geliefert werden können. Da diese Flexibilität jedoch erkauft werden muss (z.B. durch Lager- und Transportkosten), kann das in dieser Arbeit vorgestellte Verfahren in einem weiteren Schritt als ein Baustein einer kooperativen Planung dienen. In dieser erfolgt die Planung der einzelnen Unternehmen in einem Supply Chain Network nicht isoliert voreinander, sondern kooperativ durch die Abstimmung der individuellen Produktionspläne.

Literaturverzeichnis

- [AAO01] Arslan, H.; Ayhan, H.; Olsen, T. L.: Analytic models for when and how to expedite in make-to-order systems. In IIE Transactions, 2001, 33, S. 1019-1030.
- [ACK01] Azizoğlu, M.; Cakmak, E.; Kondakci, S.: A flexible flowshop problem with total flow time minimization. In European Journal of Operational Research, 2001, 132, S. 528-538.
- [AFI+08] Arnold, D.; Furmans, K.; Isermann, H.; Kuhn, A.; Tempelmeier, H.: Handbuch Logistik. Springer-Verlag, Berlin, Heidelberg, 2008.
- [AFK+93] Adler, L.; Fraiman, N.; Kobacker, E.; Pinedo, M.; Plotnicoff, J. C.; Wu, T. P.: BPSS: a scheduling support system for the packaging industry. In Operations Research, 1993, S. 641-648.
- [AkGo99] Akturk, M. S.; Gorgulu, E.: Match-up scheduling under a machine breakdown. In European Journal of Operational Research, 1999, 112, S. 81-97.
- [ALM+05] Aytug, H.; Lawley, M. A.; McKay, K.; Mohan, S.; Uzsoy, R.: Executing production schedules in the face of uncertainties: A review and some future directions. In European Journal of Operational Research, 2005, 161, S. 86-110.
- [APR+97] Agnetis, A.; Pacifici, A.; Rossi, F.; Lucertini, M.; Nicoletti, S.; Nicolo, F.; Oriolo, G.; Pacciarelli, D.; Pesaro, E.: Scheduling of flexible flow lines in an automobile assembly plant. In European Journal of Operational Research, 1997, 97, S. 348-362.
- [ArRa71] Arthanari, T. S.; Ramamurthy, K. G.: An extension of two machines sequencing problem. In Opsearch, 1971, 8, S. 10-22.
- [BDH97] Brockmann, K.; Dangelmaier, W.; Holthöfer, N.: Parallel Branch & Bound Algorithm for Makespan Optimal Scheduling in Flow Shops with Multiple Processors. In Operations Research Proceedings, 1997, S. 428-433.
- [BDS04] Berens, W.; Delfmann, W.; Schmitting, W.: Quantitative Planung: Grundlagen, Fallstudien, Lösungen. Schäffer-Poeschel, Stuttgart, 2004.
- [BeAl98] Ben-Daya, M.; Al-Fawzan, M.: A tabu search approach for the flow shop scheduling problem. In European Journal of Operational Research, 1998, 109, S. 88-95.
- [BeBi04] Bertel, S.; Billaut, J. C.: A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. In European Journal of Operational Research, 2004, 159, S. 651-662.
- [Bere92] Berens, W.: Beurteilung von Heuristiken: Neuorientierung und Vertiefung am Beispiel logistischer Probleme. Gabler, Wiesbaden, 1992.
- [BEW+01] Blazewicz, J.; Ecker, H. K.; Weglarz, J.; Schmidt, G.; Pesch, E.: Scheduling computer and manufacturing processes. Springer, Berlin, 2001.
- [BKK+04] Bilge, Ü.; Kiraç, F.; Kurtulan, M.; Pekgün, P.: A tabu search algorithm for parallel machine total tardiness problem. In Computers and Operations Research, 2004, 31, S. 397-414.

- [BMT83] Billington, P. J.; McClain, J. O.; Thomas, L. J.: Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. In *Management Science*, 1983, 29, S. 1126-1141.
- [Bott00] Botta-Genoulaz, V.: Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. In *International Journal of Production Economics*, 2000, 64, S. 101-111.
- [BrHu91] Brah, S.; Hunsucker, J. L.: Branch and bound algorithm for the flow shop with multiple processors. In *European Journal of Operational Research*, 1991, 51, S. 88-99.
- [Brock98] Brockmann, K. D. W.: A parallel branch & bound algorithm for makespan optimal sequencing in flow shops with parallel machines. In *Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications*, 1998, 3, S. 431-436.
- [BSH+08] Buschühl, L.; Sahling, F.; Helber, S.; Tempelmeier, H.: Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. In *OR Spectrum*, 2008, S. 1-31.
- [BSW06] Beutelspacher, A.; Schwenk, J.; Wolfenstetter, K.-D.: Moderne Verfahren der Kryptographie. Von RSA zu Zero-Knowledge. Vieweg, Wiesbaden, 2006.
- [CaNé00] Carlier, J.; Néron, E.: An exact method for solving the multi-processor flow-shop. In *RAIRO-Recherche Opérationnelle-Operations Research*, 2000, 34, S. 1-25.
- [CDS70] Campbell, H. G.; Dudek, R. A.; Smith, M. L.: A heuristic algorithm for the n job, m machine sequencing problem. In *Management Science*, 1970, S. 630-637.
- [CGT95] Cheng, R.; Gen, M.; Tozawa, T.: Minmax earliness/tardiness scheduling in identical parallel machine system using genetic algorithms. In *Computers & Industrial Engineering*, 1995, 29, S. 513-517.
- [COP04] Cowling, P. I.; Ouelhadj, D.; Petrovic, S.: Dynamic scheduling of steel casting and milling using multi-agents. In *Production Planning & Control*, 2004, 15, S. 178-188.
- [Cors04] Corsten, H.: Produktionswirtschaft. Einführung in das industrielle Produktionsmanagement. Oldenbourg, München, 2004.
- [Cors07] Corsten, H.: Produktionswirtschaft. Einführung in das industrielle Produktionsmanagement. Oldenbourg, München, 2007.
- [CuPe05] Curry, J.; Peters, B.: Rescheduling parallel machines with stepwise increasing tardiness and machine assignment stability objectives. In *International Journal of Production Research*, 2005, 43, S. 3231-3246.
- [DaMa94] Daniels, R. L.; Mazzola, J. B.: Flow shop scheduling with resource flexibility. In *Operations Research*, 1994, 42, S. 504-522.
- [Dang01] Dangelmaier, W.: Fertigungsplanung. Planung von Aufbau und Ablauf der Fertigung; Grundlagen, Algorithmen und Beispiele. Springer, Berlin, 2001.
- [Dang09] Dangelmaier, W.: Theorie der Produktionsplanung und -steuerung. Im Sommer keine Kirschpralinen? Springer, Berlin, 2009.

- [Daub94] Daub, A.: Ablaufplanung: Modellbildung, Kapazitätsabstimmung und Unsicherheit. Josef Eul, Köln, 1994.
- [DaWa97] Dangelmaier, W.; Warnecke, H.-J.: Fertigungslenkung. Planung und Steuerung des Ablaufs der diskreten Fertigung. Springer, Berlin, 1997.
- [DiKi94] Ding, F. Y.; Kittichartphayak, D.: Heuristics for scheduling flexible flow lines. In Computers and Industrial Engineering, 1994, 26, S. 27-34.
- [DMS04] Daniels, R. L.; Mazzola, J. B.; Shi, D.: Flow shop scheduling with partial resource flexibility. In Management Science, 2004, 50, S. 658-669.
- [DoDr05] Domschke, W.; Drexel, A.: Einführung in Operations Research. Springer, Berlin, Heidelberg, 2005.
- [DoSc05] Domschke, W.; Scholl, A.: Grundlagen der Betriebswirtschaftslehre. Eine Einführung aus entscheidungsorientierter Sicht. Springer, Berlin, 2005.
- [DSV93] Domschke, W.; Scholl, A.; Voß, S.: Produktionsplanung. Ablauforganisatorische Aspekte. Springer, Berlin, 1993.
- [FFG94] Fandel, G.; François, P.; Gubitz, K. M.: PPS-Systeme: Grundlagen, Methoden, Software, Marktanalyse. Springer, Berlin, 1994.
- [Fisc07] Fischäder, H.: Störungsmanagement in netzwerkförmigen Produktionssystemen. Deutscher Universitäts-Verlag | GWV Fachverlage GmbH, Wiesbaden, 2007.
- [Flei90] Fleischmann, B.: The discrete lot-sizing and scheduling problem. In European Journal of Operational Research, 1990, 44, S. 337-348.
- [Frei99] Frei, R.: Entstörungsmanagement in Unternehmen der chemischen Industrie. Dt. Univ.-Verl. [u.a.], Wiesbaden, 1999.
- [GaJo79] Garay, M. R.; Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP-completeness. In WH Freeman, 1979.
- [GGN00] Gourgand, M.; Grangeon, N.; Norre, S.: Meta-Heuristics for the Deterministic Hybrid Flow Shop Problem. In Journal Européen des Systèmes Automatisés, 2000, 34, S. 1107-1136.
- [GHP97] Gupta, J. N.; Hariri, A. M.; Potts, C. N.: Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. In Annals of Operations Research, 1997, 69, S. 171-191.
- [GKA07] Gienke, H.; Kämpf, R.; Aldinger, L.: Handbuch Produktion. Innovatives Produktionsmanagement: Organisation, Konzepte, Controlling. Hanser, München, 2007.
- [GKL+02] Gupta, J. N. D.; Krüger, K.; Lauff, V.; Werner, F.; Sotskov, Y. N.: Heuristics for hybrid flow shops with controllable processing times and assignable due dates. In Computers and Operations Research, 2002, 29, S. 1417-1439.
- [GKL95] Glover, F.; Kelly, J. P.; Laguna, M.: Genetic algorithms and tabu search: hybrids for optimization. In Computers and Operations Research, 1995, 22, S. 111-134.
- [Glov98] Glover, F.: A template for scatter search and path relinking. In Lecture Notes in Computer Science, 1998, 1363, S. 13-54.
- [GoSt85] Golden, B. L.; Stewart, W. R.: Empirical analysis of heuristics. In the traveling salesman problem, 1985, S. 207-249.

- [GSK+96] Guinet, A.; Solomon, M. M.; Kedia, P. K.; Dussauchoy, A.: A computational study of heuristics for two-stage flexible flowshops. In International Journal of Production Research, 1996, 34, S. 1399-1416.
- [GuNo99] Guo, B.; Nonaka, Y.: Rescheduling and optimization of schedules considering machine failures. In International Journal of Production Economics, 1999, 60, S. 503-513.
- [Gupt71] Gupta, J. N. D.: A functional heuristic algorithm for the flowshop scheduling problem. In Operational Research Quarterly, 1971, 22, S. 39-47.
- [Gupt72] Gupta, J. N. D.: Heuristic algorithms for multistage flowshop scheduling problem. In IIE Transactions, 1972, 4, S. 11-18.
- [Gupt88] Gupta, J. N. D.: Two-stage, hybrid flowshop scheduling problem. In the Journal of the Operational Research Society, 1988, 39, S. 359-364.
- [GüTe00] Günther, H.-O.; Tempelmeier, H.: Produktion und Logistik. Springer, Berlin, 2000.
- [GüTe05] Günther, H. O.; Tempelmeier, H.: Produktion und Logistik. Springer, Berlin, 2005.
- [Gute83] Gutenberg, E.: Grundlagen der Betriebswirtschaftslehre. Springer, Berlin, 1983.
- [GuTu91] Gupta, J. N.; Tunc, E. A.: Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. In International Journal of Production Research, 1991, 29, S. 1489-1502.
- [GuTu94] Gupta, J. N.; Tunc, E. A.: Scheduling a two-stage hybrid flowshop with separable setup and removal times. In European Journal of Operational Research, 1994, 77, S. 415-428.
- [GZA09] Gholami, M.; Zandieh, M.; Alem-Tabriz, A.: Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. In the International Journal of Advanced Manufacturing Technology, 2009, 42, S. 189-201.
- [HCM04] Huq, F.; Cutright, K.; Martin, C.: Employee scheduling and makespan minimization in a flow shop with multi-processor work stations: a case study. In Omega, 2004, 32, S. 121-129.
- [Heil95] Heil, M.: Entstörung betrieblicher Abläufe. Deutscher Universitäts-Verlag, Wiesbaden, 1995.
- [Helb94] Helber, S.: Kapazitätsorientierte Losgrößenplanung in PPS-Systemen. M & P Verl. für Wiss. und Forschung, Stuttgart, 1994.
- [Heue04] Heuer, J.: Das Multiprocessor Scheduling-Problem mit reihenfolgeabhängigen Rüstzeiten. Heuristische Lösungsverfahren. Dt. Univ.-Verl., Wiesbaden, 2004.
- [HYY+05] Hu, Y. H.; Yan, J. Q.; Ye, F. F.; Yu, J. H.: Flow shop rescheduling problem under rush orders. In Journal of Zhejiang University(Science), 2005, 6, S. 1040-1046.
- [IMT95] Ishibuchi, H.; Misaki, S.; Tanaka, H.: Modified simulated annealing algorithms for the flow shop sequencing problem. In European Journal of Operational Research, 1995, 81, S. 388-398.

- [JaBi99] Jahnke, H.; Biskup, D.: Planung und Steuerung der Produktion. mi Verl. Moderne Industrie, Landsberg/Lech, 1999.
- [Jäge00] Jäger, M.: Kennliniengestützte Parametereinstellung von PPS-Systemen, Universität Hannover, 2000.
- [Jens01] Jensen, M. T.: Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. In *Applied Soft Computing Journal*, 2001, 1, S. 35-52.
- [JMR98] Józefowska, J.; Mika, M.; Róeglarz, J.: Local search metaheuristics for discrete-continuous scheduling problems. In *European Journal of Operational Research*, 1998, 107, S. 354-370.
- [John54] Johnson, S. M.: Optimal two-and three-stage production schedules with setup times included. In *Naval research logistics quarterly*, 1954, 1, S. 61-68.
- [JOI+02] Jin, Z. H.; Ohno, K.; Ito, T.; Elmaghraby, S. E.: Scheduling hybrid flowshops in printed circuit board assembly lines. In *Production and Operations Management*, 2002, 11, S. 216-230.
- [JRC+08] Jungwattanakit, J.; Reodecha, M.; Chaovallwongse, P.; Werner, F.: Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. In *the International Journal of Advanced Manufacturing Technology*, 2008, 37, S. 354-370.
- [JRC+09] Jungwattanakit, J.; Reodecha, M.; Chaovallwongse, P.; Werner, F.: A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. In *Computers and Operations Research*, 2009, 36, S. 358-378.
- [Jung07] Junge, M.: Simulationsgestützte Entwicklung und Optimierung einer energieeffizienten Produktionssteuerung. Univ. Press, Kassel, 2007.
- [JYI06] Jin, Z.; Yang, Z.; Ito, T.: Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. In *International Journal of Production Economics*, 2006, 100, S. 322-334.
- [KaÖs94] Karlöf, B.; Östblom, S.: Das Benchmarking-Konzept. Wegweiser zur Spitzenleistung in Qualität und Produktivität. Vahlen, München, 1994.
- [KEH07] Kulcsár, G.; Erdélyi, F.; Hornyák, O.: Multi-Objective Optimization and Heuristic Approaches for Solving Scheduling Problems. In *IFAC Workshop on Manufacturing Modeling, Management and Control*, 2007, S. 127-132.
- [Kimm96] Kimms, A.: Multi-level, single-machine lot sizing and scheduling (with initial inventory). In *European Journal of Operational Research*, 1996, 89, S. 86-99.
- [KiSt01] Kistner, K.-P.; Steven, M.: Produktionsplanung. Physica-Verl., Heidelberg, 2001.
- [Klei71] Klein, H. K.: Heuristische Entscheidungsmodelle: Neue Techniken des Programmierens und Entscheidens für das Management. Gabler, Wiesbaden, 1971.
- [KMO+06] Kiener, S.; Maier-Scheubeck, N.; Obermaier, R.; Weiß, M.: Produktions-Management. Grundlagen der Produktionsplanung und -steuerung. Oldenbourg, München, 2006.

- [KoMo87] Kochhar, S.; Morris, R. J.: Heuristic methods for flexible flow line scheduling. In *Journal of Manufacturing Systems*, 1987, 6, S. 299-314.
- [KOY95] Kobayashi, S.; Ono, I.; Yamamura, M.: An efficient genetic algorithm for job shop scheduling problems. In *Proceedings of the 6th International Conference on Genetic Algorithms*, 1995, S. 506-511.
- [KuAs04] Kurz, M. E.; Askin, R. G.: Scheduling flexible flow lines with sequence-dependent setup times. In *European Journal of Operational Research*, 2004, 159, S. 66-82.
- [Kurb05] Kurbel, K.: *Produktionsplanung und -steuerung im Enterprise Resource Planning und Supply Chain Management*. Oldenbourg, München, 2005.
- [Laux05] Laux, H.: *Entscheidungstheorie*. Springer, Berlin, Heidelberg, 2005.
- [LiCh00] Liu, C. Y.; Chang, S. C.: Scheduling flexible flow shops with sequence-dependent setup effects. In *IEEE Transactions on Robotics and Automation*, 2000, 16, S. 408-419.
- [LiZh99] Linn, R.; Zhang, W.: Hybrid flow shop scheduling: a survey. In *Computers & Industrial Engineering*, 1999, 37, S. 57-61.
- [Low05] Low, C.: Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. In *Computers and Operations Research*, 2005, 32, S. 2013-2025.
- [MoPo00] Moursli, O.; Pochet, Y.: A branch-and-bound algorithm for the hybrid flowshop. In *International Journal of Production Economics*, 2000, 64, S. 113-125.
- [MRR+53] Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E.; others: Equation of state calculations by fast computing machines. In *the journal of chemical physics*, 1953, 21, S. 1087-1092.
- [NaMa87] Narasimhan, S. L.; Mangiameli, P. M.: A comparison of sequencing rules for a two-stage hybrid flow shop. In *Decision Sciences*, 1987, 18, S. 250-265.
- [NaPa84] Narasimhan, S. L.; Panwalkar, S. S.: Scheduling in a two-stage manufacturing process. In *International Journal of Production Research*, 1984, 22, S. 555-564.
- [NBG01] Néron, E.; Baptiste, P.; Gupta, J. N. D.: Solving hybrid flow shop problem using energetic reasoning and global operations. In *Omega*, 2001, 29, S. 501-511.
- [Near04] Nearchou, A. C.: A novel metaheuristic approach for the flow shop scheduling problem. In *Engineering Applications of Artificial Intelligence*, 2004, 17, S. 289-300.
- [NEH83] Nawaz, M.; Enscore Jr, E. E.; Ham, I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. In *Omega*, 1983, 11, S. 91-95.
- [Neuh08] Neuhaus, U.: *Reaktive Planung in der chemischen Industrie: Verfahren zur operativen Plananpassung für Mehrzweckanlagen*. Gabler, Wiesbaden, 2008.
- [Newe69] Newell, A.: Heuristic programming: III structured problems. In *Progress in Operations Research*, 1969, 3, S. 360-414.
- [Norm99] Norman, B. A.: Scheduling flowshops with finite buffers and sequence-dependent setup times. In *Computers & Industrial Engineering*, 1999, 36, S. 163-177.

- [NoSm96a] Nowicki, E.; Smutnicki, C.: A fast tabu search algorithm for the permutation flow-shop problem. In European Journal of Operational Research, 1996a, 91, S. 160-175.
- [NoSm96b] Nowicki, E.; Smutnicki, C.: A fast taboo search algorithm for the job shop problem. In Management Science, 1996b, 42, S. 797-813.
- [NoSm98] Nowicki, E.; Smutnicki, C.: The flow shop with parallel machines: A tabu search approach. In European Journal of Operational Research, 1998, 106, S. 226-253.
- [OgSm90] Ogbu, F.; Smith, D. K.: The application of the simulated annealing algorithm to the solution of the n/m/Cmax flowshop problem. In Computers & Operations Research, 1990, 17, S. 243-253.
- [OgSm91] Ogbu, F. A.; Smith, D. K.: Simulated annealing for the permutation flowshop problem. In Omega, 1991, 19, S. 64-67.
- [OsPo89] Osman, I. H.; Potts, C. N.: Simulated annealing for permutation flow-shop scheduling. In Omega, 1989, 17, S. 551-557.
- [OvUz94] Ovacik, I. M.; Uzsoy, R.: Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times. In International Journal of Production Research, 1994, 32, S. 1243-1263.
- [Pati01] Patig, S.: Flexible Produktionsfeinplanung mit Hilfe von Planungsschritten. Ein Planungsansatz zum Umgang mit Störungen bei der Produktion. Lang, Frankfurt am Main, 2001.
- [Pati99] Patig, S.: Ansatzpunkte und Rechnerunterstützung des produktionsorientierten Störungsmanagements: Ergebnis einer Literaturanalyse. Preprint Nr. 6, Fakultät für Informatik, Universität Magdeburg. Magdeburg, 1999.
- [PfHe82] Pfohl, H. C.; Hebel, R.: Bewertung heuristischer Methoden. In Mathematical Methods of Operations Research, 1982, 26, S. 123-139.
- [Pine95] Pinedo, M.: Scheduling. Theory, algorithms, and systems. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [PTR+04] Pugazhendhi, S.; Thiagarajan, S.; Rajendran, C.; Anantharaman, N.: Generating non-permutation schedules in flowline-based manufacturing systems with sequence-dependent setup times of jobs: a heuristic approach. In The International Journal of Advanced Manufacturing Technology, 2004, 23, S. 64-78.
- [Puch04] Puchta, M.: Optimierung von Problemstellungen aus der diskreten und der Prozess-Industrie unter Verwendung physikalischer Verfahren. Dissertation, Universität Regensburg, 2004.
- [PVD+98] Portmann, M. C.; Vignier, A.; Dardilhac, D.; Dezalay, D.: Branch and bound crossed with GA to solve hybrid flowshops. In European Journal of Operational Research, 1998, 107, S. 389-400.
- [Quad04] Quadt, D.: Lot-sizing and scheduling for flexible flow lines. Springer, Berlin, 2004.
- [QuKu07] Quadt, D.; Kuhn, H.: A taxonomy of flexible flow line scheduling procedures. In European Journal of Operational Research, 2007, 178, S. 686-698.

- [RaCh92a] Rajendran, C.; Chaudhuri, D.: A multi-stage parallel-processor flowshop problem with minimum flowtime. In European Journal of Operational Research, 1992a, 57, S. 111-122.
- [RaCh92b] Rajendran, C.; Chaudhuri, D.: Scheduling in n-job, m-stage flowshop with parallel processors to minimize makespan. In International Journal of Production Economics, 1992b, 27, S. 137-143.
- [RaHo99] Rajendran, C.; Holthaus, O.: A comparative study of dispatching rules in dynamic flowshops and jobshops. In European Journal of Operational Research, 1999, 116, S. 156-170.
- [Reck00] Recker, A.: Losgrößenplanung in PPS-Systemen. Optimierende versus heuristische Verfahren. Dt. Univ.-Verl. [u.a.], Wiesbaden, 2000.
- [Reev95] Reeves, C. R.: A genetic algorithm for flowshop sequencing. In Computers and Operations Research, 1995, 22, S. 5-13.
- [RFK04] Rangaswamy, R.; Ferrell, W. G.; Kurz, M. B.: Dynamic rescheduling that simultaneously considers efficiency and stability. In Computers & Industrial Engineering, 2004, 46, S. 1-15.
- [Rian98] Riane, F.: Scheduling hybrid flow shop: Algorithms and applications. Facultés Universitaires Catholiques de Mons, Belgium, 1998.
- [Ross03] Rossi, H.: Ein heuristisches Dekompositionsverfahren für mehrstufige Losgrößenprobleme. Freie Universität Berlin, 2003.
- [RŞU08] Ruiz, R.; Şerifoğlu, F. S.; Urlings, T.: Modeling realistic hybrid flexible flowshop scheduling problems. In Computers and Operations Research, 2008, 35, S. 1151-1175.
- [RuMa06] Ruiz, R.; Maroto, C.: A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. In European Journal of Operational Research, 2006, 169, S. 781-800.
- [SaBa00] Sabuncuoglu, I.; Bayiz, M.: Analysis of reactive scheduling problems in a job shop environment. In European Journal of Operational Research, 2000, 126, S. 567-586.
- [Salv73] Salvador, M. S.: A solution to a special case of flow shop scheduling problems. In Elmaghraby, S.E., Editor, 1973. Symposium on the Theory of Scheduling and Its Applications, 1973, S. 83-91.
- [Saue02] Sauer, J.: Multi-Site Scheduling Hierarchisch koordinierte Ablaufplanung auf mehreren Ebenen. Universität Oldenburg, 2002.
- [Sawi00] Sawik, T.: Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. In Mathematical and computer modelling, 2000, 31, S. 39-52.
- [Sawi02] Sawik, T.: An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers. In Mathematical and computer modelling, 2002, 36, S. 461-471.
- [Sawi95] Sawik, T. J.: Scheduling flexible flow lines with no in-process buffers. In International Journal of Production Research, 1995, 33, S. 1357-1367.

- [SBT+09] Sahling, F.; Buschkühl, L.; Tempelmeier, H.; Helber, S.: Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. In *Computers and Operations Research*, 2009, 36, S. 2546-2553.
- [Sche97] Scheer, A.-W.: *Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse*. Springer, Berlin, 1997.
- [Schm99] Schmitting, W.: Das Traveling-Salesman-Problem-Anwendung und heuristische Nutzung von Voronoi-/Delaunay-Strukturen zur Lösung euklidischer, zweidimensionaler Traveling-Salesman-Probleme. Universität Düsseldorf, 1999.
- [Schn03] Schneeweiss, C. A.: *Distributed Decision Making*. Springer, Berlin, 2003.
- [Schu06] Schuh, G.: *Produktionsplanung und -steuerung. Grundlagen, Gestaltung und Konzepte*. Springer, Berlin, 2006.
- [SeUl04] Serifoğlu, F. S.; Ulusoy, G.: Multiprocessor task scheduling in multistage hybrid flow-shops: a genetic algorithm approach. In *the Journal of the Operational Research Society*, 2004, 55, S. 504-512.
- [ShBr99a] Shafaei, R.; Brunn, P.: Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. In *International Journal of Production Research*, 1999a, 37, S. 4105-4118.
- [ShBr99b] Shafaei, R.; Brunn, P.: Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. In *International Journal of Production Research*, 1999b, 37, S. 3913-3925.
- [SHD95] Santos, D. L.; Hunsucker, J. L.; de Deal: Global lower bounds for flow shops with multiple processors. In *European Journal of Operational Research*, 1995, 80, S. 112-120.
- [SHD96] Santos, D. L.; Hunsucker, J. L.; de Deal: An evaluation of sequencing heuristics in flow shops with multiple processors. In *Computers & Industrial Engineering*, 1996, 30, S. 681-691.
- [Sieg74] Siegel, T.: *Optimale Maschinenbelegungsplanung, Zweckmäßigkeit der Zielkriterien und Verfahren zur Lösung des Reihenfolgeproblems*. E. Schmidt, Berlin, 1974.
- [Sill08] Sillekens, T.: *Aggregierte Produktionsplanung in der Automobilindustrie unter besonderer Berücksichtigung von Personalflexibilität*. Universität Paderborn, 2008.
- [SrLa86] Sriskandarajah, C.; Ladet, P.: Some no-wait shops scheduling problems: Complexity aspect. In *European Journal of Operational Research*, 1986, 24, S. 424-438.
- [SRR05] Subramaniam, V.; Raheja, A. S.; Reddy, K. R. B.: Reactive repair tool for job shop schedules. In *International Journal of Production Research*, 2005, 43, S. 1-23.

- [SrRa93] Sridhar, J.; Rajendran, C.: Scheduling in a cellular manufacturing system: a simulated annealing approach. In International Journal of Production Research, 1993, 31, S. 2927-2945.
- [SrSe89] Sriskandarajah, C.; Sethi, S. P.: Scheduling algorithms for flexible flowshops: Worst and average case performance. In European Journal of Operational Research, 1989, 43, S. 143-160.
- [Stad88] Stadtler, H.: Hierarchische Produktionsplanung bei losweiser Fertigung. Physica, Heidelberg, 1988.
- [Stam02] Stammen-Hegener, C.: Simultane Losgrößen-und Reihenfolgeplanung bei ein- und mehrstufiger Fertigung. Dt. Univ.-Verl., Wiesbaden, 2002.
- [Stüt98] Stützle, T.: An ant approach to the flow shop problem. In Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing, 1998, S. 1560-1564.
- [Suer05] Suerie, C.: Time continuity in discrete time models: new approaches for production planning in process industries. Springer, Berlin, 2005.
- [SuMe06] Suhl, L.; Mellouli, T.: Optimierungssysteme: Modelle, Verfahren, Software, Anwendungen. Springer, Berlin, 2006.
- [Tail90] Taillard, E.: Some efficient heuristic methods for the flow shop sequencing problem. In European Journal of Operational Research, 1990, 47, S. 65-74.
- [Teic98] Teich, T.: Optimierung von Maschinenbelegungsplänen unter Benutzung heuristischer Verfahren. Köln, 1998.
- [Temp06] Tempelmeier, H.: Material-Logistik. Modelle und Algorithmen für die Produktionsplanung und -steuerung in Advanced Planning-Systemen. Springer, Berlin, 2006.
- [TZV09] Tabriz, A. A.; Zandieh, M.; Vaziri, Z.: A Novel Simulated Annealing Algorithm to Hybrid Flow Shops Scheduling with Sequence-Dependent Setup Times. In Journal of Applied Sciences, 2009, 9, S. 1943-1949.
- [vAL92] van Laarhoven, P. J. M.; Aarts, E. H. L.; Lenstra, J. K.: Job shop scheduling by simulated annealing. In Operations Research, 1992, 40, S. 113-125.
- [VDI 93] VDI Richtlinie: 3633, Blatt 1: Simulation von Logistik-, Materialfluß-und Produktionssystemen-Grundlagen. In Berlin, Beuth, 1993.
- [VeDe99] Verma, S.; Dessouky, M.: Multistage hybrid flowshop scheduling with identical jobs and uniform parallel machines. In Journal of Scheduling, 1999, 2, S. 135-150.
- [VHL03] Vieira, G. E.; Herrmann, J. W.; Lin, E.: Rescheduling manufacturing systems: a framework of strategies, policies, and methods. In Journal of Scheduling, 2003, 6, S. 39-62.
- [WaFa04] Wardono, B.; Fathi, Y.: A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. In European Journal of Operational Research, 2004, 155, S. 380-401.
- [Wang05] Wang, H.: Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. In Expert Systems, 2005, 22, S. 78-85.

- [Weng00] Weng, M. X.: Scheduling flow-shops with limited buffer spaces. In Proceeding of the 2000 Winter Simulation Conference, 2000, 2, S. 1359-1363.
- [Whit84] White, S. R.: Concepts of scale in simulated annealing. In Proceedings of the IEEE International Conference on Computer Design, 1984, 122, S. 261-270.
- [WiHe89] Widmer, M.; Hertz, A.: A new heuristic method for the flow shop sequencing problem. In European Journal of Operational Research, 1989, 41, S. 186-193.
- [Wild91] Wildemann, H.: Flexible Arbeits-und Betriebszeiten-wettbewerbs-und mitarbeiterorientiert. In Bayerisches Staatsministerium München, 1991.
- [Witt85] Wittrock, R. J.: Scheduling algorithms for flexible flow lines. In IBM Journal of Research and Development, 1985, 29, S. 401-412.
- [Witt88] Wittrock, R. J.: An adaptable scheduling algorithm for flexible flow lines. In Operations Research, 1988, 36, S. 445-453.
- [WöDö05] Wöhe, G.; Döring, U.: Einführung in die allgemeine Betriebswirtschaftslehre. Vahlen, München, 2005.
- [WuLi95] Wu, H. H.; Li, R. K.: A new rescheduling method for computer based scheduling systems. In International Journal of Production Research, 1995, 33, S. 2097-2110.
- [YaRe97] Yamada, T.; Reeves, C. R.: Permutation flowshop scheduling by genetic local search. In Proceedings of the 2nd IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems, 1997, S. 232-238.
- [YBT09] Yaurima, V.; Burtseva, L.; Tchernykh, A.: Hybrid flowshop with unrelated machines, sequence-dependent setup time, availability constraints and limited buffers. In Computers & Industrial Engineering, 2009, 56, S. 1452-1463.
- [YKP00] Yang, Y.; Kreipl, S.; Pinedo, M.: Heuristics for minimizing total weighted tardiness in flexible flow shops. In Journal of Scheduling, 2000, 3, S. 89-108.
- [YMA01] Youssef, H.; M. Sait, S.; Adiche, H.: Evolutionary algorithms, simulated annealing and tabu search: a comparative study. In Engineering Applications of Artificial Intelligence, 2001, 14, S. 167-181.
- [ZäBr05] Zäpfel, G.; Braune, R.: Moderne Heuristiken der Produktionsplanung: am Beispiel der Maschinenbelegung. Vahlen, München, 2005.
- [Zäpf94] Zäpfel, G.: Entwicklungsstand und-tendenzen von PPS-Systemen. In Handbuch Produktionsmanagement, hrsg. Corsten, H. Wiesbaden: Gabler, 1994, S. 719-745.
- [Zele99] Zelewski, S.: Flexibilitätsorientierte Koordinierung von Prozessen. Arbeitsbericht Nr. 2, Universität Essen, 1999.

Anhang A: Eröffnungs- und Verbesserungsverfahren

Anhang A: Symbolverzeichnis

Symbol	Definition
A[b]	Auftrag des Bedarfs
A[los]	Auftrag des Loses
Abs(zahl)	Absoluter Wert einer Zahl
AktuelleZeit()	Triviale Funktion, gibt die aktuelle Systemzeit zurück
Änderung[b]	Änderungsmenge eines Bedarfs für das Eröffnungsverfahren
Anzahl(liste)	Triviale Funktion, gibt die Anzahl an Elementen in einer Liste zurück
Aufträge[plan]	Menge der Aufträge des Produktionsplans
Bedarf[los]	Bedarf zum Los
Bedarfe[plan]	Menge der Bedarfe des Produktionsplans
Bedarfe[ps]	Menge der Bedarfe der Produktionsstufe
BestimmeZug(p)	Triviale Funktion, die den Zug liefert, der zu Plan p führte
break	Schlüsselwort zum Beenden der aktuellen Schleife
continue	Schlüsselwort zum Überspringen der restlichen Anweisungen in der aktuellen Schleife und Fortführen im Schleifenkopf mit dem nächsten Iterationsschritt
DBK[e]	Direktbedarfskoeffizient des Erzeugnisses "e" zu seinem strukturellen Vorgänger
E[b]	Erzeugnistyp des Bedarfs
E[los]	Erzeugnistyp des Loses
EB[los]	Bedarf, der durch das Los verursacht wird
EingefroreneZone	Parameter (der letzten Periode) der eingefrorenen Zone
Ende[los]	Endzeitpunkt eines Loses
Ende[p]	Endzeitpunkt einer Periode
Ende[PlanungshorizontFP]	Endzeitpunkt des Planungshorizontes
Ende[s]	Endzeitpunkt einer Schicht
ErstesElement(liste)	Triviale Funktion, die das erste Element einer Liste zurück gibt
exists(x)	Triviale Funktion, die die Existenz des Objektes "x" prüft
Exp(zahl)	Berechnet die Exponentialfunktion einer Zahl
FixierterSchichthorizont	Parameter (der letzten Periode) des fixierten Schichthorizontes
Flexibilität[ps]	Speichert die Flexibilität der Produktionsstufe
Floor(zahl)	Berechnet die größte Ganzzahl, die kleiner oder gleich einer gegeben Zahl ist
GewähltesSchichtmodell[plan, p, l]	Gewähltes Schichtmodell von Produktionsplan "plan" in Periode "p" auf Linie "l"
InitialesSchichtmodell[p]	Initiales Schichtmodell einer Periode
InitStart[los]	Startzeitpunkt des Loses im Originalplan
K_AnzahlTeillose	Konfigurationswert für maximale Anzahl Losteilungen
K_Epsilon	Konfigurationswert für marginale Einheit
K_LagerKostenGewicht	Konfigurationswert zur Gewichtung der Lagerkosten
K_MaxPeriodenänderung	Konfigurationswert für die maximale Anzahl an Perioden, um die ein Los insgesamt verschoben werden darf
K_MaxSchichtmodelländerung	Konfigurationswert für die maximale Anzahl an Schritten, um die ein Schichtmodell verändert werden darf
K_Periodenlänge	Konfigurationswert für die Länge einer Makroperiode

Symbol	Definition
K_SA_Alpha	Konfigurationswert für den Abkühlungsfaktor
K_SA_InitialeTemperatur	Konfigurationswert der initialen Temperatur für das Simulated Annealing
K_SA_MaxIterationen	Konfigurationswert für die maximale Anzahl an Iterationen
K_SA_MaxLaufzeit	Konfigurationswert der maximalen Laufzeit
K_SA_MKL	Konfigurationswert der Markow-Ketten-Länge
K_SchichtmodellKostenGewicht	Konfigurationswert zur Gewichtung der Schichtmodellkosten
K_TS_MaxAnzahlTabuListe	Konfigurationswert für die Größe der Tabu-Liste
K_VerzugErlaubt	Konfigurationswert legt fest, ob Verzug auf PS-1 erlaubt ist
K_VerzugsKostenGewicht	Konfigurationswert zur Gewichtung der Verzugskosten
KalkBruttodauerR(plan, sz, nd, l)	Liefert die tatsächliche Dauer für die Nettodauer "nd" auf Linie "l", wenn ausgehend von Startzeitpunkt "sz" rückwärts gerechnet wird. Analog zu KalkBruttodauer, daher kein Pseudocode.
Kapazität[sm]	Kapazität des Schichtmodells "sm"
KapazitätInPuffer	Benötigte Kapazität zur Lagerung einer Einheit von Erzeugnistyp e auf Stufe ps
Kopiere(plan)	Triviale Funktion, zum Kopieren/Klonen eines Produktionsplans. Alle darin abhängigen und änderbaren Objekte werden ebenfalls kopiert/geklont.
Kosten[a]	Kosten eines Auftrags
Kosten[los]	Kosten eines Loses
Kosten[sm]	Kosten des Schichtmodells "sm"
LagerKosten[los]	Lagerkosten eines Loses
Lagerverbrauch[b]	Lagerverbrauch eines Bedarfs aus einem vorherigen Planungslauf
LetztesElement(liste)	Triviale Funktion die das letzte Element einer Liste zurück gibt
LG[s]	Leistungsgrad der Schicht
Linie[los]	Linie, der das Los zugewiesen ist
Linien[ps]	Menge der Linien in der Produktionsstufe
LKFaktor[e]	Lagerkostensatz je Zeiteinheit des Erzeugnistyps
Lose[l, p]	Menge der Lose auf Linie "l" in Periode "p"
Lose[l]	Menge der Lose auf Linie "l"
Lose[plan]	Menge der Lose des Produktionsplans
Lose[ps]	Menge der Lose der Produktionsstufe
Max(liste)	Triviale Funktion, die das größte Element einer Liste oder Menge ermittelt
MaxLosKosten[plan]	Speichert die Kosten des teuersten Loses des Plans
MaxPSFlexibilität[plan]	Speichert die Flexibilität der flexibelsten Produktionsstufe im Plan
Menge[b]	Bedarfsmenge
Menge[los]	Losgröße
Min(liste)	Triviale Funktion, die das kleinste Element einer Liste oder Menge ermittelt
Nachfolger[los]	Nachfolgerlos unter zeitlicher Betrachtung der Linie des Loses
Nachfolger[p]	Nachfolgerperiode von "p" unter zeitlicher Betrachtung
new Bedarf()	Erzeugt einen neuen Bedarf
new Los()	Erzeugt ein neues Los
Periode[t]	Periode, in der der Zeitpunkt "t" liegt
Perioden[los]	Perioden, die das Los mit seiner Start- und Endzeit berührt
PlanungshorizontFP	Planungshorizont der Feinplanung
Produzierbar[e, l]	Zeigt an, ob Erzeugnistyp "e" auf Linie "l" zu bearbeiten ist

Symbol	Definition
PS[e]	Produktionsstufe eines Erzeugnistyps
PS[los]	Produktionsstufe eines Erzeugnistyps des Loses
PS[plan]	Menge der Produktionsstufen im Produktionsplan
Puffer[e, ps]	Puffer, der Erzeugnistyp "e" und Produktionsstufe "ps" zugewiesen ist
Pufferbelegung[e, puffer]	Mengenbelegung des Puffers "puffer" von Erzeugnis "e"
PufferKapazität[puffer]	Kapazität des Puffers
Random(0,1)	Gibt eine Zufallszahl zwischen 0 und 1 zurück
Restmenge[los]	Restmenge eines Loses
Rüstzeit[e1, e1]	Nettorüstzeit von Erzeugnistyp e1 auf Erzeugnistyp e2
Schichten[sm, t]	Schicht des Schichtmodells "sm" zum Zeitpunkt "t"
Schichten[sm]	Schichten des Schichtmodells
Schichtmodelle[i, p, l]	Schichtmodelle in Periode "p" auf Linie "l", indizierbar durch "i"
Schichtmodelle[plan]	Menge der Schichtmodelle im Produktionsplan
SchichtmodellKosten[los]	Schichtmodellkosten eines Loses
Splitlos[los]	Gibt an, ob es sich bei dem Los um ein geteiltes Los handelt
Splitteil[los]	Gibt die Teile des Loses an
Start[los]	Startzeitpunkt eines Loses
Start[p]	Startzeitpunkt einer Periode
Start[PlanungshorizontFP]	Startzeitpunkt des Planungshorizontes
Start[s]	Startzeitpunkt einer Schicht
StrukturNachfolger[los]	Nachfolgerlos auf Ebene der Erzeugnisstruktur
StrukturVorgänger[los]	Vorgängerlos auf Ebene der Erzeugnisstruktur
TZ[e, l]	Taktzeit (Zykluszeit) von Erzeugnis "e" auf Linie "l"
UngeplanteBedarfe[plan]	Ungeplante Bedarfe eines Produktionsplans
VerzugLose[plan]	Liste von Losen, die mit Verzug durch den Produktionsstufenabgleich eingeplant werden
VerzugsKosten[los]	Verzugskosten eines Loses
VKFaktor[e]	Verzugskostensatz je Zeiteinheit des Erzeugnistyps
VLZ[e]	Vorlaufzeit des Erzeugnistyps
Vorgänger[b]	Vorgängerbedarf zum Bedarf "b". Der Vorgängerbedarf muss zum selben Auftrag gehören und kann über die Fertigungsstruktur identifiziert werden.
Vorgänger[e]	Struktureller Vorgänger des Erzeugnistyps
Vorgänger[los]	Vorgängerlos unter zeitlicher Betrachtung der Linie des Loses
Vorgänger[p]	Vorgängerperiode von "p" unter zeitlicher Betrachtung
Zeit[b]	Bedarfszeitpunkt des Bedarfs b
zeitberechnungMöglich	Globale Variable als Erfolgskennzeichen der Zeitberechnungsfunktionen
Zeitraum[p]	Zeitraum zwischen dem Start- und Endzeitpunkt der Periode
Zeitraum[start, ende]	Zeitraum zwischen zwei Zeitpunkten
zug [^] (-1)	Komplement eines Zuges

Anhang A: Pseudocode Eröffnungsverfahren

Algorithmus Eröffnungsverfahren(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Angepasster und erweiterter Produktionsplan "plan"

```

1.   // Es werden alle Produktionsstufen beginnend mit der ersten durchgegangen.
2.   for each Produktionsstufe ps ∈ PS[plan] do begin
3.       // Aufruf der Funktion Feinplanung für die Produktionsstufe ps.
4.       Feinplanung(plan, ps)
5.
6.       // Falls Lose mit Verzug eingeplant wurden, dazugehörige Lose auf Produktionsstufe ps-1 mit
7.       // Verzug einplanen und auf Produktionsstufe ps neu einplanen.
8.       Produktionsstufenabgleich(plan)
9.   end

```

Algorithmus Feinplanung(plan, ps)

Eingabe: Produktionsplan "plan", Produktionsstufe "ps"

Ausgabe: Angepasster und erweiterter Produktionsplan "plan"

```

1.   // Alle Bedarfssenkungen durchgehen. Diese sind zeitlich aufsteigend sortiert. Eine Bedarfssenkung
2.   // sagt aus, dass zu Termin x, von Erzeugnis e eine Menge von y Einheiten weniger benötigt wird.
3.   for each Bedarf b ∈ Bedarfe[ps] with Änderung[b] < 0 order by Zeit[b] do begin
4.       // Zu reduzierende Menge bestimmen.
5.       senkung = Abs(Änderung[b])
6.       // Startzeitpunkt für die Suche ist die erste planbare Makroperiode nach der eingefrorenen Zone.
7.       p = Nachfolger[EingefroreneZone]
8.       // Die Suche beginnt bei "periode" und reicht bis zum Ende des Feinplanungshorizontes.
9.       // Sie bricht ab, wenn die Bedarfssenkung komplett eingeplant wurde.
10.      while Start[p] < Ende[PlanungshorizontFP] and senkung > 0 do begin
11.          // Durchlaufe alle Linien der Produktionsstufe "ps" auf denen das Erzeugnis "E[b]" gefertigt
12.          // werden kann. Die Linien sind aufsteigend sortiert nach Taktzeit. Dies hat den Vorteil, dass
13.          // falls es mehrere Lose gleichen Erzeugnisses und Bedarfstermins gibt, werden zuerst die
14.          // unwirtschaftlichsten Belegungen gelöscht.
15.          for each Linie l ∈ Linien[ps] with Produzierbar[E[b], l] order by TZ[E[b], l] do begin
16.              // Durchlaufe jedes Los auf Linie "l" in Periode "p". Der Startzeitpunkt des Loses muss
17.              // außerhalb der eingefrorenen Zone liegen. Erzeugnisart und Auftrag des Loses muss
18.              // identisch zur Bedarfssenkung sein.
19.              for each Los los ∈ Lose[l, p] with Periode[Start[los]] ≥ Nachfolger[EingefroreneZone]
20.                  // and E[los] = E[b] and A[los] = A[b]
21.                  do begin
22.                      // Das gefundene Los ist kleiner/gleich der Bedarfssenkung und entfällt ganz.
23.                      if Menge[los] ≤ senkung then begin
24.                          // Bedarfssenkung wird um Losgröße reduziert.
25.                          senkung = senkung - Menge[los]
26.                          // Los aus dem Plan entfernen.
27.                          Lose[plan] = Lose[plan] \ { los }
28.                      end
29.                  else begin
30.                      // Das gefundene Los ist größer als die Bedarfssenkung, es ist zu reduzieren.
31.                      Menge[los] = Menge[los] - senkung
32.                      // Zeitraum des Loses entsprechend der Änderung verkleinern.
33.                      Start[los] = KalkStartzeit(plan, los, Ende[los])
34.                      senkung = 0
35.                  end
36.                  // Freigabe von Kapazität.
37.                  DekremSchichtmodell(plan, Perioden[los], l)
38.                  if senkung = 0 then break

```

```

39.           end
40.           if senkung = 0 then break
41.       end
42.       // Nächste Periode betrachten.
43.       p = Nachfolger[p]
44.   end
45.
46.   // Bedarfe auf Vorgänger-Produktionsstufe  $ps+1$  anpassen. Der Vorgängerbedarf muss zum
47.   // selben Auftrag gehören und kann über die Fertigungsstruktur identifiziert werden.
48.   if exists(Vorgänger[b]) = true then begin
49.       Änderung[Vorgänger[b]] = Menge[b] * DBK[E[b]] - Menge[Vorgänger[b]]
50.       Menge[Vorgänger[b]] = Menge[b] * DBK[E[b]]
51.   end
52. end
53.
54. // Gehe alle positiven Bedarfsänderungen durch. Eine Bedarfserhöhung heißt, dass ein vorheriger
55. // Auftrag um eine bestimmte Menge erhöht wird. Zudem gibt es komplett neue Bedarfe (im
56. // bekannten Horizont und im neuen Horizont). Sortierung nach Anzahl Alternativlinien und dann
57. // absteigend nach Datum. Somit erst Erzeugnistypen mit wenig Alternativlinien planen um hier
58. // flexibler zu sein.
59.
60. // Für eventuelle Lagerbestände die Bedarfserhöhungen (= neue Bedarfe) nach aufsteigendem
61. // Bedarfszeitpunkt abarbeiten, da bei den frühesten die geringsten Lagerkosten zu erwarten sind.
62. for each Bedarf b ∈ Bedarfe[ps] with Änderung[b] > 0 order by Zeit[b] do begin
63.     // Lagerbestände (Pufferbelegung) aus dem Plan verbrauchen.
64.     pufferMenge = Pufferbelegung[E[b], Puffer[E[b], ps]]
65.     if pufferMenge > 0 then begin
66.         // Ist ein echter Lagerbestand (z.B. durch Bedarfssenkung) vorhanden oder nur entfernte
67.         // Lose aufgrund der rollierenden Planung? Dazu alle konsumierenden Bedarfe des
68.         // Erzeugnistyps mit allen erzeugenden Losen des Erzeugnisses verrechnen.
69.         for each Bedarf k ∈ Bedarfe[plan] with E[k] = E[b] do begin
70.             pufferMenge = pufferMenge - Menge[k] - Änderung[k]
71.             for each Los l ∈ Lose[plan] with E[l] = E[b] do begin
72.                 pufferMenge = pufferMenge + Menge[l]
73.             end
74.             if pufferMenge > 0 then begin
75.                 restBedarf = 0
76.                 if Änderung[b] > pufferMenge then begin
77.                     // Bedarfserhöhung wird teilweise abgedeckt.
78.                     restBedarf = Menge[b] - pufferMenge
79.                     Menge[b] = pufferMenge
80.                     Änderung[b] = 0
81.                     pufferMenge = 0
82.                 end
83.                 else begin
84.                     // Bedarfserhöhung wird komplett abgedeckt.
85.                     restBedarf = 0
86.                     pufferMenge = pufferMenge - Änderung[b]
87.                     Änderung[b] = 0
88.                 end
89.             end
90.             // Der Bedarf wird direkt durch den Lagerbestand befriedigt und eine eventuelle Restmenge der
91.             // Bedarfserhöhung als neuer Bedarf mit neuem Auftrag eingeplant.
92.             if restBedarf > 0 then begin
93.                 y = new Bedarf()
94.                 A[y] = new Auftrag
95.                 E[y] = E[x]
96.                 Menge[y] = restBedarf
97.                 Änderung[y] = restBedarf

```

```

98.     Zeit[y] = Zeit[x]
99.     Bedarfe[ps] = Bedarfe[ps] ∪ { y }
100.    end
101.    end
102.
103.    // Eilaufträge besser verplanen, indem aus vorherigen Planungsläufen verwendete
104.    // Lagerbestände für neue Bedarfe mit früheren Bedarfszeitpunkten genutzt und anschließend
105.    // neue Bedarfe für die späteren Zeitpunkte angelegt werden.
106.    for each Bedarf b ∈ Bedarfe[ps] with Änderung[b] > 0 order by Zeit[b] do begin
107.        for each Bedarf x ∈ Bedarfe[ps] with E[x] = E[b] and Zeit[x] > Zeit[b]
108.            and Periode[Zeit[x]] > Nachfolger[EingefroreneZone] do begin
109.                benutztePufferMenge = Menge[x] - Änderung[x]
110.                for each Los los ∈ Lose[plan] with A[los] = A[x] and E[los] = E[x] do begin
111.                    benutztePufferMenge = benutztePufferMenge - Menge[los]
112.                    if benutztePufferMenge > 0 then begin
113.                        neuerBedarf = 0
114.                        if Änderung[b] > benutztePufferMenge then begin
115.                            // Bedarfserhöhung wird teilweise abgedeckt.
116.                            neuerBedarf = benutztePufferMenge
117.                            Änderung[b] = Änderung[b] - benutztePufferMenge
118.                            benutztePufferMenge = 0
119.                            Menge[x] = 0
120.                        end
121.                    else begin
122.                        // Bedarfserhöhung wird komplett abgedeckt.
123.                        neuerBedarf = Änderung[b]
124.                        benutztePufferMenge = benutztePufferMenge - Änderung[b]
125.                        Änderung[b] = 0
126.                        Menge[x] = Menge[x] - neuerBedarf
127.                    end
128.                    // Für den geänderten Bedarf, bzw. die Umschichtung der Nutzung von Lagerbeständen,
129.                    // wird eine neue Bedarfserhöhung angelegt, die nachfolgend eingeplant wird.
130.                    // So wird eine direkte Bedarfserhöhung vermieden, welche durch das Verfahren nicht
131.                    // eingeplant werden könnte.
132.                    if neuerBedarf > 0 then begin
133.                        y = new Bedarf()
134.                        A[y] = new Auftrag
135.                        E[y] = E[x]
136.                        Menge[y] = neuerBedarf
137.                        Änderung[y] = neuerBedarf
138.                        Zeit[y] = Zeit[x]
139.                        Bedarfe[ps] = Bedarfe[ps] ∪ { y }
140.                    end
141.                end
142.            end
143.        end
144.
145.        // Bedarfserhöhungen (= neue Bedarfe) nach absteigendem Bedarfszeitpunkt abarbeiten.
146.        for each Bedarf b ∈ Bedarfe[ps] with Änderung[b] > 0 order by desc Zeit[b] do begin
147.            // Bedarfstermin der Erhöhung liegt in der eingefrorenen Zone.
148.            gültigePeriode = true
149.            if Änderung[b] > 0 and Zeit[b] ≤ EingefroreneZone then begin
150.                // Bedarfserhöhung kann nicht automatisch eingeplant werden und kommt in Liste.
151.                gültigePeriode = false
152.            end
153.            einfügenMöglich = (Änderung[b] = 0)
154.            if Änderung[b] > 0 then begin
155.                // Neues Los auf dieser Produktionsstufe und dadurch erzeugten Bedarf auf der vorliegenden
156.                // Produktionsstufe erzeugen. Die Hierarchie entsteht mit Fortlauf des Algorithmus.

```

```

157.     los = KalkLosdetails(plan, ps, b, Änderung[b], true)
158.     if gültigePeriode then begin
159.         // Falls es eine gültige Einfügeposition gibt, wird das Los eingefügt und falls erforderlich,
160.         // das Schichtmodell angepasst.
161.         einfügenMöglich = KalkLosposition(plan, ps, b, los, false)
162.         if einfügenMöglich then begin
163.             // Einfügen der Bedarfserhöhung ist möglich.
164.             Änderung[b] = 0
165.         end
166.         else begin
167.             // Einfügen der Bedarfserhöhung nicht möglich. Insofern soll versucht werden,
168.             // das Los zu teilen.
169.             anzTeilungen = 0
170.             anzTeilLose = 2
171.             while anzTeilungen ≤ K_AnzahlTeillose do begin
172.                 // Arbeitet auf Kopie des Plans.
173.                 tempPlan = Kopiere(plan)
174.                 Lose[tempPlan] = Lose[tempPlan] \ { los }
175.                 for i = 0 to anzTeilLose do begin
176.                     teilLos = KalkLosdetails(plan, ps, b, (Änderung[b] / anzTeilLose), false)
177.                     einfügenMöglich = KalkLosposition(plan, ps, b, teilLos, false)
178.                     if einfügenMöglich = false then break
179.                 end
180.                 // Wenn alle Teillose verplant werden konnten, wird der Plan übernommen und das
181.                 // Teilen der Lose beendet.
182.                 if einfügenMöglich = true then begin
183.                     // Neuen gültigen Plan übernehmen.
184.                     plan = tempPlan
185.                     Änderung[b] = 0
186.                     break
187.                 end
188.                 anzTeilLose = anzTeilLose * 2
189.                 anzTeilungen = anzTeilungen + 1
190.             end
191.         end
192.     end
193. end
194. if einfügenMöglich = false then begin
195.     // Auf den anderen Produktionsstufen werden die Lose zu den Bedarfen für spätere
196.     // Verplanung gespeichert.
197.     VerzugLose[plan] = VerzugLose[plan] ∪ { los }
198. end
199. end

```

Algorithmus KalkLosposition(plan, ps, b, los, stufenintegration)

Eingabe: Produktionsplan "plan", Produktionsstufe "ps", Bedarf "b", Los "los",
Kennzeichen "stufenintegration", Kennzeichen "rekusiverAufruf" (Standard "false")

Ausgabe: Angepasster und erweiterter Produktionsplan "besterPlan",
Erfolgsrückmeldung "einfügenMöglich"

Beschreibung: Diese Funktion durchsucht den Lösungsraum nach einer passenden Stelle zum
Einfügen des Loses und plant dieses dort ein.

1. besterPlan = Kopiere(plan)
2. einfügenMöglich = false
3. minGesamtKosten = ∞
4. // Der Kenner "stufenintegration" bestimmt den Ablauf der Funktion in Hinblick auf Rückwärts-
- 5. // oder Vorwärtssuche. Zum Auflösen von Verzug wird diese Funktion aus der Funktion
- 6. // "Produktionsstufenabgleich" zur Vorwärtspannung aufgerufen.
- 7. **if** stufenintegration = false **then begin**

```

8. // Rückwärtssuche beginnend auf im Verlauf späteren Stufen. Neue Lose werden möglichst nah
9. // am Bedarfstermin eingeplant. Dazu wird eine Reihenfolge der Lose gebildet, in der die Lose
10. // beginnend beim spätesten Los eingeplant werden (Left-Shifting).
11. // Planungsbeginn festlegen.
12. if ps = 1 then begin
13.   if K_VerzugErlaubt then begin
14.     // Wenn Verzug erlaubt ist, wird der komplette Feinplanungshorizont berücksichtigt.
15.     t = Ende[PlanungshorizontFP]
16.   end
17.   else begin
18.     // Andernfalls ist der Horizont durch die Bedarfszeit begrenzt.
19.     t = Zeit[b]
20.   end
21. end
22. else begin
23.   t = Zeit[b]
24. end
25. // Für das Erzeugnis e erlaubte Produktionslinien, aufsteigend sortiert nach Taktzeit und darauf
26. // verplanter Kapazität.
27. for each Linie l ∈ Linien[ps] with Produzierbar[E[b], l] order by TZ[E[b], l]
28. then by BelegteKapazität[l] do begin
29.   // Pro Linie wird auf einer Kopie des Ursprungsplans nach der besten Belegung gesucht.
30.   modifizierterPlan = Kopiere(plan)
31.   // Los auf aktuelle Linie setzen.
32.   Linie[los] = l
33.   p = Periode[t]
34.   while p ≥ Nachfolger[EingefroreneZone] do begin
35.     // Es ist das günstigste Schichtmodell mit genügend Kapazität für eine Makroperiode
36.     // zu wählen.
37.     if p > FixierterSchichthorizont then begin
38.       // Außerhalb des fixierten Schichthorizonts werden alle Schichtmodelle geprüft,
39.       // die größer/gleich dem aktuell gewählten Schichtmodell sind.
40.       for each Schichtmodell sm ∈ Schichtmodelle[p, l] do begin
41.         if Kapazität[sm] ≥ GewähltesSchichtmodell[plan, p, l] then
42.           schichtmodelle = schichtmodelle ∪ { sm }
43.         end
44.       else begin
45.         // Im fixierten Schichthorizont kann das aktuelle Schichtmodell getestet werden,
46.         // sonst keines.
47.         schichtmodelle = { GewähltesSchichtmodell[plan, p, l] }
48.       end
49.     // Schichtmodelle sind aufsteigend nach Kapazität sortiert.
50.     for each Schichtmodell sm ∈ schichtmodelle order by Kapazität[sm] do begin
51.       // Für jedes Schichtmodell wird auf einer Kopie des modifizierten Plans für diese
52.       // Linie nach der besten Belegung gesucht.
53.       aktuellerPlan = Kopiere(modifizierterPlan)
54.       // Schichtmodell im Plan auswählen.
55.       GewähltesSchichtmodell[aktuellerPlan, p, l] = sm
56.       // Exakte Reihenfolge der Losabarbeitung berechnen.
57.       sortierteLose = KalkLosReihenfolgeR(aktuellerPlan, l, p, los)
58.       // Zeiten der Lose und der abhängigen Bedarfe für die gegebene Reihenfolge
59.       // berechnen.
60.       erfolg = KalkKapazitätseignungR(aktuellerPlan, sortierteLose)
61.       // Sofern der Plan bisher erfolgreich berechnet werden konnte, die gesamte
62.       // Konsistenz prüfen.
63.       if erfolg = true then
64.         erfolg = KalkKonsistenz(tempPlan, Lose[tempPlan])
65.       // Sofern der Plan bisher erfolgreich berechnet werden konnte, die Pufferauslastung
66.       // prüfen.

```

```

67.         if erfolg = true then
68.             erfolg = KalkPufferauslastung(plan, ps)
69.         if erfolg = true then begin
70.             gesamtKosten = KalkKosten(aktuellerPlan)
71.             if gesamtKosten < minGesamtKosten then begin
72.                 minGesamtKosten = gesamtKosten
73.                 besterPlan = Kopiere(aktuellerPlan)
74.                 einfügenMöglich = true
75.             end
76.         end
77.     end
78.     if einfügenMöglich = false then begin
79.         p = Vorgänger[p]
80.     else begin
81.         break
82.     end
83.   end
84. end
85. end
86. else begin
87.     // Gehe alle Perioden vorwärts durch und suche die nächste Periode, in der die Einplanung
88.     // möglich ist. Dabei wird das bisher nicht einplanbare Los frühestmöglich geplant und
89.     // sukzessive über alle nachfolgenden Produktionsstufen alle davon abhängigen Termine neu
90.     // berechnet (Right-Shifting), sowie größere Schichtmodelle geprüft. Die insgesamt günstigste
91.     // Möglichkeit wird abschließend übernommen.
92.     if exists(EB[los]) = true and exists(Zeit[EB[los]]) = true then begin
93.         // Sofern ein durch das Los erzeugter Bedarf existiert und für diesen eine gültige Zeit
94.         // berechnet wurde, wird diese als frühester Startzeitpunkt genutzt.
95.         t = Zeit[EB[los]] + VLZ[E[los]]
96.     end
97.     else begin
98.         // Andernfalls wird direkt nach der eingefrorenen Zone begonnen.
99.         t = Start[Nachfolger[EingefroreneZone]] + VLZ[E[los]]
100.    end
101.    // Für das Erzeugnis e erlaubte Produktionslinien, aufsteigend sortiert nach Taktzeit und
102.    // darauf verplanter Kapazität.
103.    for each Linie l ∈ Linien[ps] with Produzierbar[E[b], l] order by TZ[E[b], l]
104.    then by BelegteKapazität[l] do begin
105.        // Pro Linie wird auf einer Kopie des Ursprungsplans nach der besten Belegung gesucht.
106.        modifizierterPlan = Kopiere(plan)
107.        // Los auf aktuelle Linie setzen
108.        Linie[los] = l
109.        Start[los] = t
110.        p = Periode[t]
111.        while p < Ende[PlanungshorizontFP] do begin
112.            // Es ist das günstigste Schichtmodell mit genügend Kapazität für eine Makroperiode
113.            // zu wählen.
114.            if p > FixierterSchichthorizont then begin
115.                // Außerhalb des fixierten Schichthorizontes werden alle Schichtmodelle geprüft,
116.                // die größer/gleich dem aktuell gewählten Schichtmodell sind.
117.                for each Schichtmodell sm ∈ Schichtmodelle[p, l] do begin
118.                    if Kapazität[sm] ≥ GewähltesSchichtmodell[plan, p, l] then
119.                        schichtmodelle = schichtmodelle ∪ { sm }
120.                    end
121.                else begin
122.                    // Im fixierten Schichthorizont kann das aktuelle Schichtmodell getestet werden,
123.                    // sonst keines.
124.                    schichtmodelle = { GewähltesSchichtmodell[plan, p, l] }
125.                end

```

```

126.      // Schichtmodelle sind aufsteigend nach Kapazität sortiert.
127.      for each Schichtmodell sm ∈ schichtmodelle order by Kapazität[sm] do begin
128.          // Für jedes Schichtmodell wird auf einer Kopie des modifizierten Plans für diese
129.          // Linie nach der besten Belegung gesucht.
130.          aktuellerPlan = Kopiere(modifizierterPlan)
131.          // Schichtmodell im Plan auswählen.
132.          GewähltesSchichtmodell[aktuellerPlan, p, l] = sm
133.          // Exakte Reihenfolge der Losabarbeitung berechnen.
134.          sortierteLose = KalkLosReihenfolgeV(aktuellerPlan, l, los)
135.          // Zeiten der Lose und der abhängigen Bedarfe für die gegebene Reihenfolge
136.          // berechnen.
137.          erfolg = KalkKapazitätseignungV(aktuellerPlan, sortierteLose)
138.          // Sofern der Plan bisher erfolgreich berechnet werden konnte und es sich aktuell um
139.          // keinen rekusiven Aufruf der Funktion (durch "KalkKapazitätseignungV") handelt,
140.          // die gesamte Konsistenz prüfen.
141.          if erfolg = true and rekusiverAufruf = false then
142.              erfolg = KalkKonsistenz(tempPlan, Lose[tempPlan])
143.              // Sofern der Plan bisher erfolgreich berechnet werden konnte, die Pufferauslastung
144.              // prüfen.
145.              if erfolg = true then
146.                  erfolg = KalkPufferauslastung(plan, ps)
147.              if erfolg = true then begin
148.                  gesamtKosten = KalkKosten(aktuellerPlan)
149.                  if gesamtKosten < minGesamtKosten then begin
150.                      minGesamtKosten = gesamtKosten
151.                      besterPlan = Kopiere(aktuellerPlan)
152.                      einfügenMöglich = true
153.                  end
154.              end
155.          end
156.          if einfügenMöglich = false then begin
157.              p = Nachfolger[p]
158.          else begin
159.              break
160.          end
161.      end
162.  end
163. end
164. return besterPlan

```

Algorithmus Produktionsstufenabgleich(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Angepasster und erweiterter Produktionsplan "plan"

Beschreibung: Falls auf der Produktionsstufe "ps" Lose mit Verzug eingeplant wurden, müssen dazugehörige Lose auf Produktionsstufe "ps-1" neu eingeplant werden. Wenn hierbei Lose auf Produktionsstufe "ps-1" mit Verzug eingeplant werden, dann wird das Verfahren für "ps-1" wiederholt usw. D.h., auf höherer Stufe wird Belegung nach stufenspezifischen Aspekten optimiert und nach unten (vorgelagerte Stufe) weitergegeben.

```

1.  while Anzahl(VerzugLose[plan]) > 0 do begin
2.      // Arbeite auf Kopie des Plans.
3.      tempPlan = Kopiere(plan)
4.      los = ErstesElement(VerzugLose[plan])
5.      einfügenMöglich = false
6.      // Versuchen, das Los durch Stufenintegration (=Vorwärtsplanung) neu einzuplanen.
7.      einfügenMöglich = KalkLosposition(tempPlan, PS[E[los]], Bedarf[los], los, true)
8.      if einfügenMöglich = true then begin
9.          VerzugLose[tempPlan] = VerzugLose[tempPlan] \ { los }
10.         plan = tempPlan

```

```

11.   end
12.   else begin
13.     // Alle Bedarfe des Auftrags als ungeplant speichern.
14.     for each Bedarf x ∈ Bedarfe[plan] with A[x] = A[los] do begin
15.       UngeplanteBedarfe[plan] = UngeplanteBedarfe[plan] ∪ { x }
16.     end
17.     // Anschließend werden die zugehörigen Lose aus dem Plan entfernt.
18.     for each Los x ∈ Lose[plan] with A[x] = A[los] do begin
19.       Lose[plan] = Lose [plan] \ { x }
20.     end
21.   end
22. end

```

Algorithmus KalkLosdetails(plan, ps, b, menge, bedarfeAnlegen)

Eingabe: Produktionsplan "plan", Produktionsstufe "ps", Bedarf "b", Losgröße "menge", Kennzeichen "bedarfeAnlegen"

Ausgabe: Angepasster und erweiterter Produktionsplan "plan", neu erstelltes Los "los"

Beschreibung: Erstellt ein neues Los mit gegebener Losgröße ohne es zu planen, fügt es in die strukturelle Hierarchie ein und erstellt ggf. abhängige Bedarfe.

```

1.   // Ein ungeplantes Los erstellen.
2.   los = new Los()
3.   Menge[los] = menge
4.   E[los] = E[b]
5.   A[los] = A[b]
6.   Start[los] = Ende[PlanungshorizontFP]
7.   Ende[los] = Ende[PlanungshorizontFP]
8.   Linie[los] = null
9.   // Strukturhierarchie der Lose herstellen.
10.  if ps > 1 then begin
11.    // Erstes Los des Auftrags auf der Nachfolgerproduktionsstufe (PS-1) suchen.
12.    losNachfolger = null
13.    for each Los x ∈ Lose[plan] with A[x] = A[b] and PS[E[x]] = ps-1 order by Start[x] do begin
14.      losNachfolger = x
15.      break
16.    end
17.    if exists(losNachfolger) = true then begin
18.      losVorgänger = StrukturVorgänger[losNachfolger]
19.      // Das neue Los wird direkt vor den strukturellen Vorgänger des ersten Loses dieses
20.      // Auftrags auf der Nachfolgerproduktionsstufe (PS-1) gehängt.
21.      if exists(losVorgänger) = true then begin
22.        StrukturNachfolger[losVorgänger] = los
23.        StrukturVorgänger[los] = losVorgänger
24.      end
25.      // Das neue Los wird direkt hinter das erste Los dieses Auftrags auf der
26.      // Nachfolgerproduktionsstufe (PS-1) gehängt
27.      StrukturVorgänger[losNachfolger] = los
28.      StrukturNachfolger[los] = losNachfolger
29.    end
30.  end
31.  Lose[plan] = Lose[plan] ∪ { los }
32.  if bedarfeAnlegen = true and exists(Vorgänger[E[b]]) = true and ps < Anzahl(PS[plan])
33.  then begin
34.    vorgängerBedarf = new Bedarf()
35.    A[vorgängerBedarf] = A[b]
36.    E[vorgängerBedarf] = Vorgänger[E[b]]
37.    Menge[vorgängerBedarf] = Menge[b] * DBK[E[b]]
38.    Änderung[vorgängerBedarf] = Menge[vorgängerBedarf]

```

```

39.      Bedarfe[plan] = Bedarfe[plan] ∪ { vorgängerBedarf }
40.      end

```

Algorithmus KalkLosReihenfolgeR(plan, l, p, los)

Eingabe: Produktionsplan "plan", Linie "l", Periode "aktuellePeriode", Los "los"

Ausgabe: Sortierte Liste von Losen "sortierteLose"

Beschreibung: Exakte Reihenfolge zur Losabarbeitung durch Rückwärtsterminierung berechnen

```

1.      minRüstzeit = ∞
2.      sortierteLoseTemp = { }
3.      p = Ende[PlanungshorizontFP]
4.      while p ≥ Nachfolger[EingefroreneZone] do begin
5.          lose = { }
6.          for each Los x ∈ Lose[plan] with x != los and Linie[x] = l
7.              and Ende[x] ≥ Start[p] and Ende[x] < Ende[p] and Start[l] > Ende[EingefroreneZone]
8.              order by desc Start[x] do begin
9.                  lose = lose ∪ { x }
10.             end
11.             if p = aktuellePeriode then begin
12.                 // In der aktuellen Planungsperiode das neue Los rüstzeitoptimal einreihen.
13.                 for each Los x ∈ lose do begin
14.                     sortierteLoseTemp = sortierteLoseTemp ∪ { x }
15.                     losNachfolger = Nachfolger[x]
16.                     rüstzeitN = Rüstzeit[E[x], E[los]]
17.                     rüstzeitV = Rüstzeit[E[los], E[losNachfolger]]
18.                     rüstzeitZ = Rüstzeit[E[x], E[losNachfolger]]
19.                     rüstzeit = rüstzeitN + rüstzeitV - rüstzeitZ
20.                     // Sichern der günstigsten Position.
21.                     if rüstzeit < minRüstzeit then begin
22.                         minRüstzeit = rüstzeit
23.                         bestePosition = x
24.                     end
25.                 end
26.                 // Los an die beste Position einfügen.
27.                 sortierteLose = { }
28.                 for 0 to x = Anzahl(sortierteLoseTemp) do begin
29.                     sortierteLose = sortierteLose ∪ { sortierteLoseTemp[x] }
30.                     if x = bestePosition then
31.                         sortierteLose = sortierteLose ∪ { los }
32.                     end
33.                 end
34.             else begin
35.                 // Außerhalb der aktuellen Planungsperiode die Reihenfolge der Lose übernehmen.
36.                 for each Los x ∈ lose do begin
37.                     sortierteLose = sortierteLose ∪ { x }
38.                 end
39.             end
40.             p = Vorgänger[p]
41.         end
42.     return sortierteLose

```

Algorithmus KalkLosReihenfolgeV(plan, l, los)

Eingabe: Produktionsplan "plan", Linie "l", Los "los"

Ausgabe: Sortierte Liste von Losen "sortierteLose"

Beschreibung: Exakte Reihenfolge zur Losabarbeitung durch Vorwärtsterminierung berechnen.

```

1.  sortierteLose = KalkLosReihenfolgeR(plan, l, los)
2.  // Sortierung der Liste umdrehen.
3.  sortierteLose = Reverse(sortierteLose)
4.  return sortierteLose

```

Algorithmus KalkKapazitätseignungR(plan, lose)

Eingabe: Produktionsplan "plan", Menge von sortierten Losen "lose"

Ausgabe: Angepasster und erweiterter Produktionsplan "plan",

Erfolgsrückmeldung "erfolg"

Beschreibung: Führt die Rückwärtsplanung auf der Linie für eine Menge von Losen durch. Dabei werden die Zeitpunkte aller Lose für die gegebene Situation komplett neu berechnet.

```

1.  erfolg = true
2.  losNachfolger = null
3.  for each Los los ∈ lose do begin
4.    if exists(losNachfolger) = true then begin
5.      rüstzeitN = Rüstzeit[E[los], E[losNachfolger]]
6.      rüstzeitB = KalkBruttodauerR(plan, Start[losNachfolger], rüstzeitN, Linie[los])
7.      Ende[los] = Min(Zeit[Bedarf[los]], Start[losNachfolger] - rüstzeitB)
8.    else begin
9.      Ende[los] = Zeit[Bedarf[los]]
10.    end
11.    Start[los] = KalkStartzeit(plan, los, Ende[los])
12.    if Periode[Start[los]] ≤ EingefroreneZone then
13.      erfolg = false
14.    if Ende[los] > Ende[PlanungshorizontFP] then
15.      erfolg = false
16.    if erfolg = true then begin
17.      // Bedarfszeitpunkte für Vorgänger-Produktionsstufe PS+1 setzen.
18.      Zeit[EB[los]] = Start[los] - VLZ[E[los]]
19.      // Gültigkeit des Bedarfszeitpunktes prüfen.
20.      if Periode[Zeit[EB[los]]] ≤ EingefroreneZone then begin
21.          erfolg = false
22.      end
23.      if erfolg = false then break
24.      losNachfolger = los
25.    end
26.  return erfolg

```

Algorithmus KalkKapazitätseignungV(plan, lose)

Eingabe: Produktionsplan "plan", Menge von sortierten Losen "lose"

Ausgabe: Angepasster und erweiterter Produktionsplan "plan",

Erfolgsrückmeldung "erfolg"

Beschreibung: Führt die Vorwärtsplanung auf der Linie für eine Menge von Losen durch. Dabei werden die Zeitpunkte aller Lose für die gegebene Situation komplett neu berechnet.

```

1.  erfolg = true
2.  losVorgänger = Vorgänger[ErstesElement[lose]]
3.  EinplanungNächsteStufe = { }
4.  for each Los los ∈ lose do begin
5.    if exists(losVorgänger) = true then begin
6.      rüstzeitN = Rüstzeit[E[losVorgänger], E[los]]
7.      rüstzeitB = KalkBruttodauer(plan, Start[losVorgänger], rüstzeitN, Linie[los])

```

```

8.      // Nur anpassen, falls keine ausreichende Lücke vorhanden war.
9.      if Ende[losVorgänger] + rüstzeitB > Start[los] then begin
10.         if exists(EB[los]) = true then begin
11.             Start[los] = Max(Zeit[EB[los]] + VLZ[E[los]], Ende[losVorgänger] + rüstzeitB)
12.             else begin
13.                 Start[los] = Ende[losVorgänger] + rüstzeitB
14.             end
15.         end
16.     end
17.     Ende[los] = KalkEndzeit(plan, los)
18.     if Periode[Start[los]] ≤ EingefroreneZone then
19.         erfolg = false
20.     if Ende[los] > Ende[PlanungshorizontFP] then
21.         erfolg = false
22.     // Wenn kein Verzug erlaubt ist, müssen die Endzeiten der Lose auf PS 1 kleiner/gleich den
23.     // Bedarfsterminen sein.
24.     if PS[E[los]] = 1 and K_VerzugErlaubt = false and Ende[los] > Zeit[Bedarf[los]] then
25.         erfolg = false
26.     // Kann der Bedarfszeitpunkt nicht eingehalten werden, wird rekursiv über die nachfolgenden
27.     // Produktionsstufen die Gültigkeit des Plans wiederhergestellt.
28.     if erfolg = true and Ende[los] > Zeit[Bedarf[los]] and PS[E[los]] > 1 then begin
29.         // Bedarfszeitpunkte für diese Produktionsstufe setzen.
30.         EinplanungNächsteStufe = EinplanungNächsteStufe ∪ { los }
31.     end
32.     if erfolg = true and PS[E[los]] < Anzahl(PS[plan]) then begin
33.         // Bedarfszeitpunkte für Vorgänger-Produktionsstufe PS+1 setzen.
34.         Zeit[EB[los]] = Start[los] - VLZ[E[los]]
35.         // Gültigkeit des Bedarfszeitpunktes prüfen
36.         if Periode[Zeit[EB[los]]] ≤ EingefroreneZone then begin
37.             erfolg = false
38.         end
39.         if erfolg = false then break
40.         losVorgänger = los
41.     end
42.     // Start der rekursiven Einplanung auf nächster Stufe erst nachdem die aktuelle Stufe
43.     // abgearbeitet ist.
44.     if erfolg = true and Anzahl(EinplanungNächsteStufe) > 0 then begin
45.         for each Los los ∈ EinplanungNächsteStufe do begin
46.             // Bedarfszeitpunkte für diese Produktionsstufe setzen.
47.             Zeit[Bedarf[los]] = Ende[los]
48.             // Gültigkeit des Bedarfszeitpunktes prüfen.
49.             if Periode[Zeit[Bedarf[los]]] ≤ EingefroreneZone then begin
50.                 erfolg = false
51.                 // Wenn der Bedarfszeitpunkt nicht mehr eingehalten werden kann, müssen die zugehörigen
52.                 // Lose der PS-1 neu geplant werden.
53.                 for each Los x ∈ Lose[plan] with A[x] = A[Nachfolger[Bedarf[los]]]
54.                 and E[x] = E[Nachfolger[Bedarf[los]]] do begin
55.                     // Rekursiv die abhängigen Lose auf PS-1 planen.
56.                     erfolg = KalkLosposition(plan, PS[E[x]], Nachfolger[Bedarf[los]], x, true, true)
57.                     if erfolg = false then break
58.                 end
59.                 if erfolg = false then break
60.             end
61.         end
62.     return erfolg

```

Anhang A: Pseudocode Verbesserungsverfahren

Algorithmus SimulatedAnnealing (plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Verbesserter Produktionsplan "besterPlan"

Beschreibung: Ablauf der übergeordneten Metaheuristik des Simulated Annealing.

```

1.  // Initialisierung
2.  besterPlan = plan
3.  k = 0
4.  j = 0
5.  InitLose(plan)
6.  InitSchichtmodelle(plan)
7.  startTime = AktuelleZeit()
8.  temperatur = K_SA_InitialeTemperatur
9.  // Optimiere, bis Abbruchkriterium erfüllt.
10. while j < K_SA_MaxIterationen and (AktuelleZeit() – startTime) ≤ K_SA_MaxLaufeit do begin
11.   i = 0
12.   while i < K_SA_MKL do begin
13.     nachbarn = { }
14.     while Anzahl(nachbarn) = 0 do begin
15.       basislos = KalkBasislos(plan, ausgeschlosseneLose)
16.       nachbarn = KalkNachbarn(plan, basislos)
17.       if Anzahl(nachbarn) = 0 then
18.         ausgeschlosseneLose = ausgeschlosseneLose ∪ { basislos }
19.       end
20.       minKosten = ∞
21.       // Ermitteln des günstigsten Plans aller Nachbarpläne.
22.       for each Plan nachbar ∈ nachbarn do begin
23.         kosten = KalkKosten(nachbar)
24.         if kosten < minKosten then begin
25.           minKosten = kosten
26.           besterNachbar = nachbar
27.         end
28.       end
29.       kostenNachbar = KalkKosten(besterNachbar)
30.       kostenPlan = KalkKosten(plan)
31.       if kostenNachbar < kostenPlan then begin
32.         // Bester Nachbarplan ist besser als aktueller Plan.
33.         plan = besterNachbar
34.         ausgeschlosseneLose = { }
35.       end
36.       else begin
37.         // Prüfe auf Übernahme durch Zufall.
38.         zufallsZahl = Random(0,1)
39.         if kostenNachbar != kostenPlan
40.           and zufallsZahl < Exp((-kostenNachbar - kostenPlan)) / temperatur then begin
41.             plan = besterNachbar
42.             ausgeschlosseneLose = { }
43.           end
44.           else begin
45.             ausgeschlosseneLose = ausgeschlosseneLose ∪ { basislos }
46.           end
47.         end
48.       if KalkKosten(plan) < KalkKosten(besterPlan) then begin
49.         // Neues Optimum gefunden.
50.         besterPlan = plan
51.         j = 0
52.       end

```

```

53.      else begin
54.          j = j + 1
55.      end
56.      i = i + 1
57.  end
58.  k = k + 1
59.  temperatur = temperatur * K_SA_Alpha
60. end
61. return besterPlan

```

Algorithmus TabuSearch(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Verbesserter Produktionsplan "besterPlan"

Beschreibung: Ablauf der übergeordneten Metaheuristik der Tabu Search.

```

1.  // Initialisierung
2.  besterPlan = plan
3.  j = 0
4.  InitLose(plan)
5.  InitSchichtmodelle(plan)
6.  startTime = AktuelleZeit()
7.  // Optimiere, bis Abbruchkriterium erfüllt.
8.  while j < K_TS_MaxIterationen and (AktuelleZeit() – startTime) ≤ K_TS_MaxLaufzeit do begin
9.      while Anzahl(Nachbarn) = 0 do begin
10.         basislos = KalkBasislos(plan, ausgeschlosseneLose)
11.         nachbarn = KalkNachbarn(plan, basislos)
12.         if Anzahl(nachbarn) = 0 then
13.             ausgeschlosseneLose = ausgeschlosseneLose ∪ { basislos }
14.         end
15.         // Versuche möglichst guten Plan der Nachbarschaft zu nutzen. Nachbarn sind nach
16.         // aufsteigenden Kosten sortiert.
17.         for each Plan nachbar ∈ nachbarn order by KalkKosten(nachbar) do begin
18.             zug = BestimmeZug(nachbar)
19.             kostenNachbar = KalkKosten(nachbar)
20.             kostenBesterPlan = KalkKosten(besterPlan)
21.             if kostenNachbar < kostenBesterPlan then begin
22.                 // Neues Optimum gefunden.
23.                 besterPlan = nachbar
24.                 plan = nachbar
25.                 if zug ∉ tabuListe then begin
26.                     if Anzahl(tabuListe) = K_TS_MaxAnzahlTabuListe then begin
27.                         tabuListe = tabuListe \ { ErstesElement(tabuListe) }
28.                     end
29.                     tabuListe = tabuListe ∪ { zug-1 }
30.                 end
31.                 j = 0
32.                 ausgeschlosseneLose = { }
33.                 break
34.             end
35.             else if zug ∉ tabuListe then begin
36.                 // Übernehme besten Nachbarn als aktuellen Plan.
37.                 plan = nachbar
38.                 if Anzahl(tabuListe) = K_TS_MaxAnzahlTabuListe then begin
39.                     tabuListe = tabuListe \ { ErstesElement(tabuListe) }
40.                 end
41.                 tabuListe = tabuListe ∪ { zug-1 }
42.                 j = j + 1
43.                 ausgeschlosseneLose = { }

```

```

44.         break
45.     end
46.   end
47. end
48. return besterPlan

```

Algorithmus KalkBasislos(plan, ausgeschlosseneLose)

Eingabe: Produktionsplan "plan", Liste von ausgeschlossenen Basislosen „ausgeschlosseneLose“

Ausgabe: Los mit maximalem Potential „basislos“

Beschreibung: Die Funktion liefert das Los mit maximalem Potential zurück, d.h. es weiß die höchste Kombination aus Kosten und Flexibilität auf.

```

1.  InitKosten(plan)
2.  // Berechnen der Auftragskosten.
3.  Kosten[ ] = 0
4.  for each Los los ∈ Lose[plan] do begin
5.    if ErstesElement[Perioden[los] ] > EingefroreneZone then begin
6.      Kosten[A[los]] = Kosten[A[los] ] + Kosten[los]
7.    end
8.  end
9.  // Bestimmung von „basislos“ auf normierten Kosten und Flexibilitäten.
10. InitFlexibilität(plan)
11. maxPotential = -∞
12. for each Auftrag auftrag ∈ Aufträge[plan] order by desc Kosten[auftrag] do begin
13.   for each Produktionsstufe ps ∈ PS[plan] do begin
14.     for each Los los ∈ Lose[ps] with A[los] = auftrag do begin
15.       if los ∈ ausgeschlosseneLose then begin
16.         if ErstesElement[Perioden(los)] > EingefroreneZone then begin
17.           normierungsfaktor = MaxLosKosten[plan] / MaxPSFlexibilität[plan]
18.           aktPotential = Kosten[los] + Flexibilität[ps] * normierungsfaktor
19.           if maxPotential < aktPotential then begin
20.             maxPotential = aktPotential
21.             basislos = los
22.           end
23.         end
24.       end
25.     end
26.   end
27.   if exists(basislos) = true then break
28. end
29. return basislos

```

Algorithmus InitFlexibilität(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Angepasster Produktionsplan "plan"

Beschreibung: Berechne grobe Flexibilität aller Produktionsstufen, um dies bei der Auswahl des Basisloses zu berücksichtigen. Hier ist noch nicht bekannt, wohin ein Los verschoben werden soll, daher grob mittels 3 Indikatoren ganze Stufe betrachten, statt genau losbezogen.

```

1.  MaxPSFlexibilität[plan] = 0
2.  for each Produktionsstufe ps ∈ PS[plan] do begin
3.    minFlexibilität = 0
4.    kapazitätsDifferenz = 0
5.    freiraumSumme = 0
6.    for each Linie l ∈ Linien[ps] do begin
7.      periodenBearbeitet = { }
8.      for each Los los ∈ Lose[l] do begin
9.        if ErstesElement[Perioden[los] ] > EingefroreneZone then begin

```

```

10.      // Berechne Losflexibilität in Minuten.
11.      // Berechnung der Flexibilität ohne Berücksichtigung des direkten
12.      // Vorgänger- bzw. Nachfolgerloses.
13.      minFlexibilität = minFlexibilität + Start[los] - KalkFAZ(plan, los, false) +
14.          KalkSEZ(plan, los, false) - Ende[los]
15.      // Berechne maximale Schichtmodelldifferenz.
16.      for each Periode p ∈ Perioden(los) do begin
17.          if p ∉ periodenBearbeitet and p > FixierterSchichthorizont then begin
18.              maxSchichtmodell = Max(Schichtmodelle[_, p, l])
19.              aufschlag = Kapazität[maxSchichtmodell] -
20.                  Kapazität[GewähltesSchichtmodell[plan, p, l]]
21.              kapazitätsDifferenz = kapazitätsDifferenz + aufschlag
22.              periodenBearbeitet = periodenBearbeitet ∪ { p }
23.          end
24.      end
25.      // Berechne Summe der Leeräume und beachte ggf. Freiraum vor erstem und nach
26.      // letztem Los.
27.      if los = ErstesElement[Lose[l]] then begin
28.          freiraumSumme = freiraumSumme + Start[los] - Start[PlanungshorizontFP]
29.      end
30.      if exists[Nachfolger[los]] then begin
31.          freiraumSumme = freiraumSumme + Ende[los] - Start[Nachfolger[los]]
32.      end
33.      else begin
34.          freiraumSumme = freiraumSumme + Ende[PlanungshorizontFP] - Ende[los]
35.      end
36.      end
37.  end
38.  end
39.  // Speichern der ermittelten Flexibilität der Produktionsstufe und ggf. flexibelste
40.  // Produktionsstufe des Plans aktualisieren.
41.  Flexibilität[ps] = minFlexibilität + kapazitätsDifferenz + freiraumSumme
42.  if Flexibilität [ps] > MaxPSFlexibilität[plan] then begin
43.      MaxPSFlexibilität[plan] = Flexibilität[ps]
44.  end
45. end

```

Algorithmus KalkNachbarn(plan, basislos)

Eingabe: Produktionsplan "plan", Basislos "basislos"

Ausgabe: Liste von Nachbarn "nachbarschaft"

Beschreibung: Berechnet für einen Plan und ein Basislos die gesamte Nachbarschaft.

```

1.  // Arbeit auf Kopie des Plans.
2.  tempPlan = Kopiere(plan)
3.  // Initialisierung der Daten.
4.  nachbarschaft = { }
5.  linieBasislos = Linie[basislos]
6.  periodenBasislos = Perioden(basislos)
7.  alternativPositionen = KalkAlternativpositionen(tempPlan, basislos)
8.  // Entferne Basislos und reduziere in diesem Bereich das Schichtmodell.
9.  Lose[linieBasislos] = Lose[linieBasislos] \ { basislos }
10. DekremSchichtmodell(tempPlan, periodenBasislos, linieBasislos)
11. for each los ∈ alternativPositionen \ { Vorgänger[basislos] } do begin
12.     // Arbeit auf Kopie des Ursprungsplans.
13.     aktPlan = Kopiere(tempPlan)
14.     // Füge das Basislos "basislos" hinter die aktuelle alternative Position "los" ein.
15.     Linie[basislos] = Linie[los]
16.     rüstzeitN = Rüstzeit[E[los], E[basislos]]

```

```

17.     rüstzeitB = KalkBruttodauer(aktPlan, Ende[los], rüstzeitN, Linie[los])
18.     Start[basislos] = Ende[los] + rüstzeitB
19.     Ende[basislos] = KalkEndzeit(aktPlan, basislos)
20.     konsistent = true
21.     // Aktualisiere ggf. Bedarfssituation auf vorgelagerter Stufe.
22.     if PS[basislos] != Anzahl(PS[aktPlan]) then begin
23.         if SplitLos[basislos] = true then begin
24.             Zeit[EB[basislos]] = Min(Start[Splitteile[basislos]])
25.         end
26.         else begin
27.             Zeit[EB[basislos]] = Start[basislos] - VLZ[E[basislos]]
28.         end
29.     end
30.     // Prüfe Zulässigkeit des Plans für das Basislos.
31.     if konsistent = true then begin
32.         konsistent = KalkKonsistenz(aktPlan, { basislos })
33.     end
34.     // Schließe ggf. Lücke bei Konsistenz, versuche sonst Konsistenz herzustellen.
35.     if konsistent = false then begin
36.         konsistent = SpezifischeReparatur(aktPlan, basislos)
37.     end
38.     else begin
39.         DekremSchichtmodell(aktPlan, Perioden(basislos), linie[basislos])
40.     end
41.     // Übernehme aktuellen gültigen Plan in die Menge der Nachbarn.
42.     if konsistent = true then begin
43.         // Pufferbelegung nur prüfen, wenn beschränkte Puffer vorhanden sind.
44.         if PS[basislos] = Anzahl(PS[aktPlan]) then begin
45.             // Wenn es sich um die erste Produktionsstufe handelt, nur deren Puffer prüfen.
46.             pufferPrüfung = KalkPufferauslastung(aktPlan, PS[basislos])
47.         end
48.         else begin
49.             // Wenn es sich nicht um die erste Produktionsstufe handelt, deren Puffer und den
50.             // Puffer der vorgelagerten Stufe prüfen.
51.             pufferPrüfung = KalkPufferauslastung(aktPlan, PS[basislos]) and
52.                 KalkPufferauslastung(aktPlan, PS[basislos]+1)
53.         end
54.         // Konsistenten Plan übernehmen, wenn Pufferbelegung in Ordnung ist.
55.         if pufferPrüfung = true then begin
56.             nachbarschaft = nachbarschaft ∪ { aktPlan }
57.         end
58.     end
59. end
60. return nachbarschaft

```

Algorithmus KalkAlternativpositionen(plan, basislos)

Eingabe: Produktionsplan "plan", Basislos "basislos"

Ausgabe: Liste von Alternativpositionen "lose"

Beschreibung: Durchläuft alle Lose der aktuellen Produktionsstufe und wählt die erfolgversprechenden unter den möglichen aus.

```

1.     lose = { }
2.     for each Linie l ∈ Linien[PS[basislos]] do begin
3.         if Produzierbar[E[basislos], l] = true then begin
4.             for each Los los ∈ Lose[l] do begin
5.                 // Berechne Distanz zwischen in Frage kommender Startposition und LMax.
6.                 rüstzeitN = Rüstzeit[E[los], E[basislos]]
7.                 rüstzeitB = KalkBruttodauer(plan, Ende[los], rüstzeitN, l)
8.                 distanz = Abs(InitStart[basislos] - Ende[los] + rüstzeitB) / K_Periodenlänge

```

```

9.         if distanz ≤ K_MaxPeriodenänderung
10.        and LetztesElement[Perioden(los)] > EingefroreneZone then begin
11.            lose = lose ∪ { los }
12.        end
13.    end
14.  end
15. end
16. return lose

```

Algorithmus SpezifischeReparatur(plan, problemlos)

Eingabe: Produktionsplan "plan", Problemlos "problemlos"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "erfolg"

Beschreibung: Bewertet die derzeitige Situation und löst eine geeignet erscheinende

Reparaturfunktion aus. Analog zu "InitFlexibilität", allerdings hier nur Berücksichtigung relevanter Daten statt ganzer Produktionsstufe.

```

1.  // Bestimme Schichtmodell Flexibilität für die relevanten Perioden.
2.  perioden = Perioden[problemlos]
3.  if Anzahl(perioden) ≤ 2 then begin
4.    perioden = perioden ∪
5.      Vorgänger[ErstesElement[perioden]] ∪ Nachfolger[LetztesElement[perioden]]
6.  end
7.  kapazitätsDifferenz = 0
8.  for each Periode p ∈ perioden do begin
9.    if p > FixierterSchichthorizont then
10.      maxSM = Max(Schichtmodelle[_, p, Linie[problemlos]])
11.      kapazitätsDifferenz = kapazitätsDifferenz + Kapazität[maxSM] -
12.        Kapazität[GewähltesSchichtmodell[plan, p, Linie[problemLos]]]
13.    end
14.  end
15.
16.  minFlexibilität = 0
17.  freiraumSumme = 0
18.  for each Los los ∈ Lose[PS[problemLos]] with Periode[Start[los]] > EingefroreneZone do begin
19.    if Linie[los] = Linie[problemLos] then begin
20.      minFlexibilität = minFlexibilität + Start[los] -
21.        KalkFAZ(plan, los, true) + KalkSEZ(plan, los, true) - Ende[los]
22.    end
23.    // Für erstes Los auf der Linie des Loses.
24.    if los = ErstesElement[Lose[Linie[los]]] then begin
25.      freiraumSumme = freiraumSumme + Start[los] - Start[PlanungshorizontFP]
26.    end
27.    // Wenn das Los einen Nachfolger hat.
28.    if exists[Nachfolger[los]] then begin
29.      freiraumSumme = freiraumSumme + Ende[los] - Start[Nachfolger[los]]
30.    end
31.    else begin
32.      freiraumSumme = freiraumSumme + Ende[PlanungshorizontFP] - Ende[los]
33.    end
34.  end
35.  if minFlexibilität + freiraumSumme ≥ kapazitätsDifferenz then begin
36.    erfolg = LoseVerschiebenOderTeilen(plan, problemLos)
37.  end
38.  else begin
39.    erfolg = InkremSchichtmodell(plan, problemLos)
40.  end
41. return erfolg

```

Algorithmus LoseVerschiebenOderTeilen(plan, problemlos)

Eingabe: Produktionsplan "plan", Problemlos "problemlos"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "erfolg"

Beschreibung: Bestimme maximale Ausdehnung der Lücke nach vorne, falls nötig auch nach hinten.

Wenn so keine Konsistenz hergestellt werden konnte, splitte Problemteil ab und prüfe auf
passende Lücke bei paralleler Linie.

```

1. aktLinie = Linie[problemlos]
2. erfolg = false
3. // Nach vorne schieben.
4. los = ErstesElement[Lose[aktLinie]]
5. while los != Nachfolger[problemlos] do begin
6.   if Periode[Start[los]] > EingefroreneZone then begin
7.     Start[los] = KalkFAZ(plan, los, true)
8.     if PS[los] != Anzahl(PS[plan]) then begin
9.       if SplitLos[los] = true then begin
10.         Zeit[EB[los]] = Min(Start[Splitteile[los]])
11.       end
12.     else begin
13.       Zeit[EB[los]] = Start[los] - VLZ[E[los]]
14.     end
15.   end
16.   Ende[los] = KalkEndzeit(plan, los)
17.   los = Nachfolger[los]
18. end
19. // Prüfe auf Konsistenz.
20. if Ende[problemlos] ≤ KalkSEZ(plan, problemlos, true) then
21.   return true
22. // Nach hinten schieben.
23. los = LetztesElement[Lose[aktLinie]]
24. while los != problemlos do begin
25.   if Periode[Start[los]] > EingefroreneZone then begin
26.     Start[los] = KalkSAZ(plan, aktLos, true)
27.     if PS[los] != Anzahl(PS[plan]) then begin
28.       if SplitLos[los] = true then begin
29.         Zeit[EB[los]] = Min(Start[Splitteile[los]])
30.       end
31.     else begin
32.       Zeit[EB[los]] = Start[los] - VLZ[E[los]]
33.     end
34.   end
35.   Ende[los] = KalkEndzeit(plan, los)
36.   los = Vorgänger[los]
37. end
38. // Prüfe auf Konsistenz.
39. if Ende[problemlos] ≤ KalkSEZ(plan, problemlos, true) then
40.   return true
41. // Verschieben der Lose war nicht erfolgreich, nun plane möglichst viel des Problemlosen in die
42. // Lücke ein (nur ganze Stücke) und splitte den Rest ab.
43. zeitraumN = KalkSEZ(plan, problemlos, true) - Start[problemlos]
44. zeitraumB = KalkBruttodauer(plan, Start[los], zeitraumN, Linie[los])
45. lückeStück = Floor(zeitraumB / TZ[E[problemlos], Linie[los]])
46. if lückeStück > 0 then begin
47.   teilStück = Menge[problemlos] - lückeStück
48.   Menge[problemlos] = lückeStück
49.   Ende[problemlos] = KalkEndzeit[plan, problemlos]
50.   // Bilde Menge relevanter Linien und sortiere später nach Eignung.
51.
52.

```

```

53. relevanteLinien = { }
54. for each Linie l ∈ Linien[PS[problemlos]] with l != aktLinie do begin
55.   if Produzierbar[E[problemlos], l] then begin
56.     relevanteLinien = relevanteLinien ∪ { l }
57.   end
58. end
59. // Bestimme FAZ und SEZ des Problemloses ohne Berücksichtigung des direkten Vorgängers
60. // oder Nachfolgers.
61. teilLosFAZ = KalkFAZ(plan, problemlos, false)
62. teilLosSEZ = KalkSEZ(plan, problemlos, false)
63. // Suche passende Lücke für abgespalteten Teil.
64. for each Linie l ∈ relevanteLinien order by TZ[E[problemlos], l] do begin
65.   for each Los los ∈ Lose[l] do begin
66.     if Periode[Ende[los]] > EingefroreneZone and Ende[los] < teilLosSEZ then begin
67.       rüstzeitV = Rüstzeit[E[los], E[problemlos]]
68.       rüstzeitA = Rüstzeit[E[problemlos], E[Nachfolger[los]]]
69.       rüstzeitN = Rüstzeit[E[los], E[Nachfolger[los]]]
70.       rüstzeitNB = KalkBruttodauer(plan, Ende[los], rüstzeitN, l)
71.       // Prüfe ob "echte Lücke" vorhanden oder nur Rüstzeit.
72.       if Ende[los] + rüstzeitNB < Start[Nachfolger[los]] then begin
73.         rüstzeitVB = KalkBruttodauer(plan, Ende[los], rüstzeitV, l)
74.         rüstzeitAB = KalkBruttodauerR(plan, Start[Nachfolger[los]], rüstzeitA, l)
75.         startLücke = Ende[los] + rüstzeitVB
76.         endeLücke = Start[Nachfolger[los]] + rüstzeitAB
77.         zeitraum = Zeitraum[startLücke, endeLücke] ∩ Zeitraum[teilLosFAZ, teilLosSEZ]
78.         teilLos = new Los()
79.         Menge[teilLos] = teilStück
80.         E[teilLos] = E[problemlos]
81.         Start[teilLos] = Start[zeitraum]
82.         Linie[los] = l
83.         // Speicherung der Split-Informationen.
84.         SplitLos[problemlos] = true
85.         // Prüfe, ob "echte Lücke" für SplitLos ausreicht.
86.         if KalkEndzeit[plan, teilLos] ≤ Ende[zeitraum] then begin
87.           // Plane neues TeilLos ein.
88.           A[teilLos] = A[problemlos]
89.           Ende[teilLos] = KalkEndzeit[plan, teilLos]
90.           // Aktualisiere ggf. Bedarfssituation auf vorgelagerter Stufe.
91.           if PS[problemlos] != Anzahl(PS[plan]) then begin
92.             if Start[teilLos] < Min(Start(Splittteile[problemlos])) then begin
93.               Zeit[EB[problemlos]] = Start[problemlos] - VLZ[E[problemlos]]
94.             end
95.           end
96.           // Erzeugnisbasierte Vorgänger-/Nachfolgerhierarchie um das Splitlos
97.           // erweitern.
98.           if exists(Nachfolger[problemlos]) = true then begin
99.             Vorgänger[Nachfolger[problemlos]] = teilLos
100.            Nachfolger[teilLos] = Nachfolger[problemlos]
101.           end
102.           Nachfolger[problemlos] = teilLos
103.           Vorgänger[teilLos] = problemlos
104.           // Los in den Plan einfügen.
105.           Lose[plan] = Lose[plan] ∪ { teilLos }
106.           return true
107.         end
108.       end
109.     end
110.   end
111. end

```

```

112. end
113. return false

```

Anhang A: Pseudocode gemeinsame Hilfsfunktionen

Algorithmus InitSchichtmodelle(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Angepasster Produktionsplan "plan"

Beschreibung: Bestimmt die gewählten Schichtmodelle der Perioden des Planungshorizontes.

```

1. for each Produktionsstufe ps ∈ PS[plan] do begin
2.   for each Linie l ∈ Linien[ps] do begin
3.     for each Periode p ∈ Perioden[plan] do begin
4.       InitialesSchichtmodell[p] = GewähltesSchichtmodell[plan, p, l]
5.     end
6.   end
7. end

```

Algorithmus InitLose(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Angepasster Produktionsplan "plan"

Beschreibung: Ermittelt die Position jedes Loses im Ausgangsplan.

```

1. for each Los los ∈ Lose[plan] do begin
2.   InitStart[los] = Start[los]
3. end

```

Algorithmus InitKosten(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Angepasster Produktionsplan "plan"

Beschreibung: Berechnet kundenneutrale Kosten und speichert sie separat sowie in Summe im Los.

Dies sind Lagerkosten, Verzugskosten, Schichtmodellkosten.

```

1. // Initialisierung, d.h. zurücksetzen aller Kostenarten.
2. MaxLosKosten[plan] = 0
3. for each Los los ∈ Lose[plan] do begin
4.   SchichtmodellKosten[los] = -1
5.   LagerKosten[los] = 0
6.   VerzugsKosten[los] = 0
7.   Restmenge[los] = Menge[los]
8. end
9. // Start der Kostenberechnung.
10. for each Produktionsstufe ps ∈ PS[plan] do begin
11.   // Gegebene Pufferbelegung einbeziehen.
12.   virtuellerPuffer[ ] = 0
13.   for each Erzeugnis e ∈ Erzeugnisse[ps] do begin
14.     virtuellerPuffer[e] = tmpPuffer[e] + Pufferbelegung[e, Puffer[e, ps]]
15.   end
16.   // Alle Bedarfe durchlaufen.
17.   for each Bedarf b ∈ Bedarfe[ps] do begin
18.     restBedarf = Menge[b]
19.     if virtuellerPuffer[E[b]] > 0 then begin
20.       restBedarf = restBedarf - virtuellerPuffer[E[b]]
21.       virtuellerPuffer[E[b]] = virtuellerPuffer[E[b]] - Menge[b]
22.     end
23.     if restBedarf = 0 then continue
24.     for each Los los ∈ Lose[ps] order by Start[los] do begin

```

```

25.      // Einmalig pro Los die Schichtmodellkosten bestimmen.
26.      if SchichtmodellKosten[los] = -1 then begin
27.          rüstzeitN = Rüstzeit[E[Vorgänger[los]], E[los]]
28.          rüstzeitB = KalkBruttodauerR(plan, Start[los], rüstzeitN, Linie[los])
29.          zeitraum = Zeitraum[Start[los] - rüstzeitB, Ende[los]]
30.          perioden = Perioden(zeitraum)
31.          SchichtmodellKosten[los] = 0
32.          for each Periode p ∈ perioden do begin
33.              anteil = (zeitraum ∩ Zeitraum[p]) / Zeitraum[p]
34.              summand = Kosten[GewähltesSchichtmodell[plan, p, Linie[los]]] * anteil
35.              SchichtmodellKosten[los] = SchichtmodellKosten[los] + summand
36.          end
37.      end
38.      // Bestimmen der Lagerkosten und Verzugskosten.
39.      if E[los] = E[b] then begin
40.          zeitDifferenz = Zeit[b] - Ende[los]
41.          mengenDifferenz = restBedarf - Restmenge[los]
42.          // Lagerkosten entstehen bei positiver Zeitdifferenz.
43.          if zeitDifferenz ≥ 0 then begin
44.              if mengenDifferenz > 0 then begin
45.                  // Los reicht nicht um Bedarf zu decken.
46.                  restBedarf = mengenDifferenz
47.                  aufschlag = zeitDifferenz * LKFaktor[E[los]] * Restmenge[los]
48.                  LagerKosten[los] = LagerKosten[los] + aufschlag
49.                  Restmenge[los] = 0
50.              end
51.              else begin
52.                  aufschlag = zeitDifferenz * LKFaktor[E[los]] *
53.                               (Restmenge[los] + mengenDifferenz)
54.                  LagerKosten[los] = LagerKosten[los] + aufschlag
55.                  Restmenge[los] = mengenDifferenz * -1
56.                  break
57.              end
58.          end
59.          // Verzugskosten entstehen bei negativer Zeitdifferenz.
60.          else begin
61.              if mengenDifferenz > 0 then begin
62.                  // Los reicht nicht um Bedarf zu decken.
63.                  restBedarf = mengenDifferenz
64.                  aufschlag = zeitDifferenz * VKFaktor[E[los]] * Restmenge[los]
65.                  VerzugsKosten[los] = VerzugsKosten[los] + aufschlag
66.                  Restmenge[los] = 0
67.              end
68.              else begin
69.                  aufschlag = zeitDifferenz * VKFaktor[E[los]] *
70.                               (Restmenge[los] + mengenDifferenz)
71.                  VerzugsKosten[los] = VerzugsKosten[los] + aufschlag
72.                  Restmenge[los] = mengenDifferenz * -1
73.                  break
74.              end
75.          end
76.      end
77.      // Speichern der Kostenpotentiale im Los und ggf. das maximale Loskostenpotential
78.      // des Plans aktualisieren.
79.      Kosten[los] = SchichtmodellKosten[los] * K_SchichtmodellKostenGewicht +
80.                     LagerKosten[los] * K_LagerKostenGewicht +
81.                     VerzugsKosten[los] * K_VerzugsKostenGewicht
82.      if Kosten[los] > MaxLosKosten[Plan] then begin
83.          MaxLosKosten[plan] = Kosten[los]

```

```

84.           end
85.       end
86.   end
87. end

```

Algorithmus KalkKosten(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Kosten des Produktionsplans

Beschreibung: Berechnet die gesamten realen Kosten des Plans, d.h. Verzugskosten, Lagerkosten, Schichtmodellkosten. Dazu müssen die Lose zuvor mit "InitKosten" initialisiert worden sein.

```

1.  InitKosten(plan)
2.  kosten = KalkLagerKosten(plan)
3.  kosten = kosten + KalkVerzugsKosten(plan)
4.  for each Periode p ∈ Perioden[plan] do begin
5.    for each Linie l ∈ Linien[plan] do begin
6.      summe = summe + Kosten[GewähltesSchichtmodell[plan, p, l]]
7.    end
8.  end
9.  return summe

```

Algorithmus KalkLagerkosten(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Lagerkosten des Produktionsplans

```

1.  kosten = 0
2.  for each Los los ∈ Lose[plan] do begin
3.    kosten = kosten + LagerKosten[los] * K_LagerKostenGewicht
4.  end
5.  return kosten

```

Algorithmus KalkVerzugskosten(plan)

Eingabe: Produktionsplan "plan"

Ausgabe: Verzugskosten des Produktionsplans

```

1.  kosten = 0
2.  for each Los los ∈ Lose[plan] do begin
3.    kosten = kosten + VerzugsKosten[los] * K_VerzugsKostenGewicht
4.  end
5.  return kosten

```

Algorithmus KalkBruttodauer(plan, startzeit, nettodauer, l)

Eingabe: Produktionsplan "plan", Startzeitpunkt "startzeit", Nettodauer "nettodauer", Linie "l"

Ausgabe: Angepasster Produktionsplan "plan"

Beschreibung: Liefert die Bruttodauer für eine Nettodauer auf dieser Linie, wenn ausgehend vom Startzeitpunkt vorwärts gerechnet wird.

```

1.  originalStartzeit = startzeit
2.  p = Periode[startzeit]
3.  // Solange Nettodauer nicht vollständig verrechnet wurde, wird in den Schichten der späteren
4.  // Makroperioden nach Kapazität gesucht.
5.  while nettodauer > 0 do begin
6.    schichten = Schichten[GewähltesSchichtmodell[plan, p, l]]
7.    for each Schicht s ∈ schichten do begin
8.      // Wenn Leistungsgrad der Schicht null ist, dann überspringen.
9.      if LG[s] = 0 then
10.        continue

```

```

11.      if startzeit = savedStart then begin
12.          schichtKapazität = Ende[s] - startzeit
13.      end
14.      else begin
15.          schichtKapazität = Ende[s] - Start[s]
16.      end
17.      // Prüfe, ob Nettodauer in aktueller Schicht verrechnet werden kann und gebe Ergebnis
18.      // zurück oder aktualisiere Variablen für die nächste Iteration.
19.      if schichtKapazität > (nettodauer / LG[s]) then begin
20.          return (startzeit + nettodauer / LG[s]) - originalStartzeit
21.      end
22.      else begin
23.          nettodauer = nettodauer - schichtKapazität * LG[s]
24.          startzeit = startzeit + schichtKapazität
25.      end
26.      end
27.      p = Nachfolger[p]
28.  end

```

Algorithmus KalkFAZ(plan, los, vorgängerBetrachten)

Eingabe: Produktionsplan "plan", zu betrachtendes Los "los", Kennzeichen "vorgängerBetrachten"
 Ausgabe: Frühester Anfangszeitpunkt (FAZ)
 Beschreibung: Berechnet den frühesten Anfangszeitpunkt (FAZ) des Loses.

```

1.      // Bestimme Ende des Vorgängers.
2.      if vorgängerBetrachten and exists[Vorgänger[los]] then begin
3.          rüstzeitN = Rüstzeit[E[Vorgänger[los]], E[los]]
4.          endeVorgänger = Ende[Vorgänger[los]] +
5.              KalkBruttodauer(plan, Ende[Vorgänger[los]], rüstzeitN, Linie[los])
6.      end
7.      else begin
8.          endeVorgänger = Start[PlanungshorizontFP]
9.      end
10.     // Bestimme Ende der vorgelagerten Stufe.
11.     endeStufe = 0
12.     for each Los aktLos ∈ Lose[PS[los]+1] do begin
13.         if A[aktLos] = A[los] and Ende[aktLos] > endeStufe then begin
14.             endeStufe = Ende[aktLos]
15.         end
16.     end
17.     // Berücksichtigung der maximalen Planungsflexibilität, der Nachlaufzeit und der
18.     // eingefrorenen Zone.
19.     tempFAZ = Max(endeVorgänger, endeStufe + VLZ[E[los]], Ende[EingefroreneZone])
20.     if (InitStart[los] - tempFAZ) / K_Periodenlänge > K_MaxPeriodenänderung then begin
21.         return InitStart[los] - K_MaxPeriodenänderung * K_Periodenlänge
22.     end
23.     else begin
24.         return tempFAZ
25.     end

```

Algorithmus KalkSEZ(plan, los, nachfolgerBetrachten)

Eingabe: Produktionsplan "plan", zu betrachtendes Los "los", Kennzeichen "nachfolgerBetrachten"
 Ausgabe: Spätester Endzeitpunkt (SEZ)
 Beschreibung: Berechnet den spätesten Endzeitpunkt (SEZ) des Loses.

```

1.      // Bestimme Start des Nachfolgers.
2.      if nachfolgerBetrachten and exists[Nachfolger[los]] then begin
3.          rüstzeitN = Rüstzeit[E[los], E[Nachfolger[los]]]

```

```

4.     startNachfolger = Start[Nachfolger[los]] -
5.         KalkBruttodauerR(plan, Start[Nachfolger[los]], rüstzeitN, Linie[los])
6.     end
7.     else begin
8.         startNachfolger = Ende[PlanungshorizontFP]
9.     end
10.    // Bestimme Anfang der nachgelagerten Stufe.
11.    startStufe =  $\infty$ 
12.    for each Los aktLos  $\in$  Lose[PS[los]-1] do begin
13.        if A[aktLos] = A[los] and Start[aktLos] < startStufe then begin
14.            startStufe = Start[aktLos]
15.        end
16.    end
17.    // Berücksichtigung der maximalen Planungsflexibilität und der Nachlaufzeit
18.    tempSEZ = Min(startNachfolger, startStufe - VLZ[E[los]])
19.    if (tempSEZ - InitStart[los]) / K_Periodenlänge > K_MaxPeriodenänderung then begin
20.        return InitStart[los] + K_MaxPeriodenänderung * K_Periodenlänge
21.    end
22.    else begin
23.        return tempSEZ
24.    end

```

Algorithmus DekremSchichtmodell(plan, perioden, l)

Eingabe: Produktionsplan "plan", Liste relevanter Perioden "perioden", Linie "l"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "einfügenMöglich"

Beschreibung: Suche kleineres Schichtmodell bis gerade nicht mehr alle Lose einplanbar sind, um dann vorheriges auszuwählen.

```

1.     einfügenMöglich = true
2.     do
3.         // Arbeitet auf Kopie des Plans.
4.         tempPlan = Kopiere(plan)
5.         geändertePeriode = null
6.         maxKosten = 0
7.         // Suche nächstes Schichtmodell, welches die meisten Kosten einspart.
8.         for each Periode p  $\in$  perioden do begin
9.             aktSM = GewähltesSchichtmodell[tempPlan, p, l]
10.            nächstesVerfügbar = exists[Schichtmodelle[aktSM-1, p, l]]
11.            nächstesMöglich = Abs(InitialesSchichtmodell[p] - (aktSM-1))  $\leq$ 
12.                K_MaxSchichtmodelländerung
13.            if nächstesVerfügbar and nächstesMöglich and p > FixierterSchichthorizont then begin
14.                kostenAktSM = Kosten[Schichtmodelle[aktSM, p, l]]
15.                kostenVorSM = Kosten[Schichtmodelle[aktSM-1, p, l]]
16.                if kostenAktSM - kostenVorSM > maxKosten then begin
17.                    maxKosten = kostenAktSM - kostenVorSM
18.                    maxSM = Schichtmodelle[aktSM-1, p, l]
19.                    geändertePeriode = p
20.                end
21.            end
22.        end
23.        if geändertePeriode != null then begin
24.            GewähltesSchichtmodell[tempPlan, geändertePeriode, l] = maxSM
25.            einfügenMöglich = KalkLoszeiten(tempPlan, l, perioden)
26.            if einfügenMöglich = true then begin
27.                // Bei Erfolg wird die Änderung im Originalplan übernommen, das spart das spätere
28.                // Zurücksetzen.
29.                GewähltesSchichtmodell[plan, geändertePeriode, l] = maxSM
30.                KalkLoszeiten(plan, l, perioden)

```

```

31.      end
32.      end
33.      while einfügenMöglich = true and geändertePeriode != null
34.      return einfügenMöglich

```

Algorithmus KalkLoszeiten(plan, l, perioden)

Eingabe: Produktionsplan "plan", Linie "l", Liste relevanter Perioden "perioden"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "erfolg"

Beschreibung: Plant Lose in der Linie im Bereich der übergebenen Perioden frühestmöglich neu ein und zeigt den Erfolg an.

```

1.  erfolg = true
2.  for each Periode p ∈ perioden do begin
3.    if p > EingefroreneZone then begin
4.      ersterDurchlauf = true
5.      for each Los los ∈ Lose[l, p] do begin
6.        // Plane kein Los vor dem Startzeitpunkt der Periode ein.
7.        if ersterDurchlauf = true then begin
8.          Start[los] = Max(Start[p], KalkFAZ(plan, los, true))
9.        end
10.       else begin
11.         Start[los] = KalkFAZ(plan, los, true)
12.       end
13.       // Berechne Ende neu und aktualisiere Bedarfszeitpunkt.
14.       Ende[los] = KalkEndzeit(plan, los)
15.       // Auf aktueller Linie prüfen, ob es sich nicht um PS-n handelt.
16.       if PS[l] != Anzahl(PS[plan]) then begin
17.         if SplitLos[los] = true then begin
18.           Zeit[EB[los]] = Min(Start[Splitteile[los]])
19.         end
20.         else begin
21.           Zeit[EB[los]] = Start[los] - VLZ[E[los]]
22.         end
23.       end
24.       // Prüfe ob konsistent eingeplant wurde.
25.       if Ende[los] > KalkSEZ(plan, los, true) then begin
26.         erfolg = false
27.       end
28.       ersterDurchlauf = false
29.     end
30.   end
31. end
32. return erfolg

```

Algorithmus KalkEndzeit(plan, los)

Eingabe: Produktionsplan "plan", Los "los"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "erfolg"

Beschreibung: Berechnet für gegebenes Startdatum eines Loses das zugehörige Enddatum und gibt es zurück. Zudem wird der Erfolg der Berechnung über die globale Variable "zeitberechnungMöglich" angezeigt.

```

1.  restMenge = Menge[los]
2.  endzeit = Start[los]
3.  zeitberechnungMöglich = false
4.  // Solange das Los nicht vollständig eingeplant ist
5.  while restMenge > 0 do begin
6.    // Prüfen, ob die aktuelle Periode überhaupt noch im Plan vorkommt. Über
7.    // "zeitberechnungMöglich" wird der Misserfolg zurückgemeldet.

```

```

8.  if exists(Periode[endzeit]) = false then
9.    return undefined
10.   // Prüfen, ob die aktuelle Periode noch planbar ist. Über "zeitberechnungMöglich" wird der
11.   // Misserfolg zurückgemeldet.
12.   if Periode[endzeit] ≤ EingefroreneZone then
13.     return undefined
14.     sm = GewähltesSchichtmodell[plan, Periode[endzeit], Linie[los]]
15.     schicht = Schichten[sm, endzeit]
16.     kapInMin = Ende[schicht] - endzeit
17.     kapInStück = Floor((kapInMin * LG[schicht]) / TZ[E[los], Linie[los]])
18.     if kapInStück ≥ restMenge then begin
19.       // Los passt in Schicht.
20.       restZeit = (restMenge * TZ[E[los], Linie[los]]) / LG[schicht]
21.       endzeit = endzeit + restZeit
22.       restMenge = 0
23.     end
24.   else begin
25.     // Los passt nicht in Schicht. Epsilon stellt eine marginale Einheit dar, damit wird
26.     // sichergestellt, dass die Endzeit nicht genau auf den Zeitpunkt zwischen Ende dieser Schicht
27.     // und Anfang nächster Schicht gesetzt wird.
28.     endzeit = endzeit + kapInMin + K_Epsilon
29.     restMenge = restMenge - kapInStück
30.   end
31. end
32. // Wenn die Zeitberechnung erfolgreich war, wird die globale Variable "zeitberechnungMöglich"
33. // mit "true" belegt.
34. zeitberechnungMöglich = true
35. return endzeit

```

Algorithmus KalkStartzeit(plan, los, endzeit)

Eingabe: Produktionsplan "plan", Los "los", Endzeitpunkt "endzeit"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "erfolg"

Beschreibung: Berechnet für gegebenes Enddatum eines Loses das zugehörige Startdatum und gibt es zurück. Zudem wird der Erfolg der Berechnung über die globale Variable "zeitberechnungMöglich" angezeigt.

```

1.  restMenge = Menge[los]
2.  startzeit = endzeit
3.  zeitberechnungMöglich = false
4.  // Solange das Los nicht vollständig eingeplant ist
5.  while restMenge > 0 do begin
6.    // Prüfen, ob die aktuelle Periode überhaupt noch im Plan vorkommt. Über
7.    // "zeitberechnungMöglich" wird der Misserfolg zurückgemeldet.
8.    if exists(Periode[startzeit]) = false then
9.      return undefined
10.     // Prüfen, ob die aktuelle Periode noch planbar ist. Über "zeitberechnungMöglich" wird der
11.     // Misserfolg zurückgemeldet.
12.     if Periode[startzeit] ≤ EingefroreneZone then
13.       return undefined
14.       sm = GewähltesSchichtmodell[plan, Periode[startzeit], Linie[los]]
15.       schicht = Schichten[sm, startzeit]
16.       kapInMin = startzeit - Start[schicht]
17.       kapInStück = Floor((kapInMin * LG[schicht]) / TZ[E[los], Linie[los]])
18.       if kapInStück ≥ restMenge then begin
19.         // Los passt in Schicht.
20.         restZeit = (restMenge * TZ[E[los], Linie[los]]) / LG[schicht]
21.         startzeit = startzeit - restZeit
22.         restMenge = 0

```

```

23.    end
24. else begin
25.    // Los passt nicht in Schicht. Epsilon stellt eine marginale Einheit dar, damit wird
26.    // sichergestellt, dass die Endzeit nicht genau auf den Zeitpunkt zwischen Ende dieser Schicht
27.    // und Anfang nächster Schicht gesetzt wird.
28.    startzeit = startzeit - kapInMin - K_Epsilon
29.    restMenge = restMenge - kapInStück
30.    end
31. end
32. // Wenn die Zeitberechnung erfolgreich war, wird die globale Variable "zeitberechnungMöglich"
33. // mit "true" belegt.
34. zeitberechnungMöglich = true
35. return startzeit

```

Algorithmus KalkSAZ(plan, los, nachfolgerBetrachten)

Eingabe: Produktionsplan "plan", Los "los", Kennzeichen "nachfolgerBetrachten"

Ausgabe: Spätester Anfangszeitpunkt (SAZ)

Beschreibung: Berechnet den spätesten Anfangszeitpunkt (SAZ) eines Loses und gibt ihn zurück.

```

1.  sez = KalkSEZ(plan, los, nachfolgerBetrachten)
2.  // Wenn die Zeitberechnung erfolgreich war, wird die Variable "zeitberechnungMöglich"
3.  // durch die Methode "KalkSEZ" entsprechend belegt.
4.  if zeitberechnungMöglich = true then begin
5.    KalkStartzeit(plan, los, sez)
6.  end

```

Algorithmus InkremSchichtmodell(plan, los)

Eingabe: Produktionsplan "plan", Los "los"

Ausgabe: Angepasster Produktionsplan "plan", Erfolgsrückmeldung "einfügenMöglich"

Beschreibung: Wähle schrittweise nächstgrößeres Schichtmodell, bis Lose aus relevanten Perioden einplanbar sind. Relevante Perioden sind die des Problemloses sowie ggf. die vorherige und die nachfolgende Periode.

```

1.  einfügenMöglich = false
2.  perioden = Perioden(los)
3.  if Anzahl(perioden) ≤ 2 then begin
4.    perioden = perioden ∪
5.    Vorgänger[ErstesElement[perioden]] ∪ Nachfolger[LetztesElement[perioden]]
6.  end
7.  do
8.    geändertePeriode = null
9.    minKosten = ∞
10.   // Erhöhe Schichtmodell in der Periode, in der Erhöhung am günstigsten ist.
11.   for each Periode p ∈ perioden do begin
12.     aktSM = GewähltesSchichtmodell[plan, p, l]
13.     nächstesVerfügbar = exists[Schichtmodelle[aktSM+1, p, l]]
14.     nächstesMöglich = Abs(InitialesSchichtmodell[p] - (aktSM+1)) ≤
15.       K_MaxSchichtmodelländerung
16.     if nächstesVerfügbar and nächstesMöglich and p > FixierterSchichthorizont then begin
17.       kostenAktSM = Kosten[Schichtmodelle[aktSM, p, l]]
18.       kostenNächstesSM = Kosten[Schichtmodelle[aktSM+1, p, l]]
19.       if kostenNächstesSM - kostenAktSM < minKosten then begin
20.         minKosten = kostenNächstesSM - kostenAktSM
21.         minSM = Schichtmodelle[aktSM+1, p, l]
22.         geändertePeriode = p
23.       end
24.     end
25.   end

```

```

26.  if geändertePeriode != null then begin
27.      GewähltesSchichtmodell[plan, geändertePeriode, l] = minSM
28.      einfügenMöglich = KalkLoszeiten(plan, l, perioden)
29.  end
30.  while einfügenMöglich = false and geändertePeriode != null
31.  return einfügenMöglich

```

Algorithmus KalkPufferauslastung(plan, ps)

Eingabe: Produktionsplan "plan", Produktionsstufe "ps"

Ausgabe: Erfolgsrückmeldung

Beschreibung: Prüft, ob die maximalen Kapazitäten der Puffer zwischen aktueller und nachgelagerter Stufe eingehalten werden. Dazu werden die Bedarfe und Lose zeitlich sortiert, sodass eine Menge von Events durchlaufen werden kann. Dabei wird nach jedem Zugang auf Einhaltung der Pufferkapazität getestet.

```

1.  tmpPuffer[ ] = 0
2.  // Bereits belegte Puffer einbeziehen.
3.  for each Erzeugnis e ∈ Erzeugnisse[ps] do begin
4.      puffer = Puffer[e, ps]
5.      tmpPuffer[puffer] = tmpPuffer[puffer] + Pufferbelegung[e, puffer] * KapazitätInPuffer[e, ps]
6.  end
7.  // Ein- und Ausgänge der Puffer einbeziehen.
8.  for each Objekt event ∈ Lose[ps] ∪ Bedarfe[ps] order by Zeit[event] do begin
9.      puffer = Puffer[E[event], ps]
10.     if Typ(event) = Typ(Los) then begin
11.         // Zugang, da Los fertiggestellt.
12.         tmpPuffer[puffer] := tmpPuffer[puffer] + Menge[event] * KapazitätInPuffer[E[event], ps]
13.         // Ist die Puffergröße überschritten, kann abgebrochen werden.
14.         if tmpPuffer[puffer] > PufferKapazität[puffer] then
15.             return false
16.     end
17.     else begin
18.         // Abgang, da Bedarf vorliegen muss.
19.         tmpPuffer[puffer] = tmpPuffer[puffer] - Menge[event] * KapazitätInPuffer[E[event], ps]
20.     end
21.  end
22.  return true

```

Algorithmus KalkKonsistenz(plan, lose)

Eingabe: Produktionsplan "plan", Liste von betroffenen Losen "lose"

Ausgabe: Erfolgsrückmeldung "konsistent"

Beschreibung: Bestimmt die Gültigkeit eines Produktionsplans.

```

1.  konsistent = true
2.  for each Los los ∈ lose do begin
3.      // Prüfe auf Überschneidung mit Nachfolger (zeitlich auf selber Linie).
4.      if konsistent = true then begin
5.          if exists[Nachfolger[los]] then begin
6.              rüstzeitN = Rüstzeit[E[los], E[Nachfolger[los]]]
7.              rüstzeitB = KalkBruttodauer(plan, Ende[los], rüstzeitN, Linie[los])
8.              // Wenn die Zeitberechnung erfolgreich war, wird die Variable "zeitberechnungMöglich"
9.              // durch die Methode "KalkBruttodauer" entsprechend belegt.
10.             if zeitberechnungMöglich = false then
11.                 konsistent = false
12.                 if Ende[los] + rüstzeitB > Start[Nachfolger[los]] then
13.                     konsistent = false
14.                 end
15.             end

```

```

16. // Prüfe auf Überschneidung mit Vorgänger (zeitlich auf selber Linie).
17. if konsistent = true then begin
18.   if exists[Vorgänger[los]] then begin
19.     rüstzeitN = Rüstzeit[E[Vorgänger[los]], E[los]]
20.     rüstzeitB = KalkBruttodauer(plan, Ende[Vorgänger[los]], rüstzeitN, Linie[los])
21.     // Wenn die Zeitberechnung erfolgreich war, wird die Variable "zeitberechnungMöglich"
22.     // durch die Methode "KalkBruttodauer" entsprechend belegt.
23.     if zeitberechnungMöglich = false then
24.       konsistent = false
25.     if Ende[Vorgänger[los]] + rüstzeitB > Start[los] then
26.       konsistent = false
27.     end
28.   end
29. // Prüfe auf Überschneidung mit vorgelagerter Produktionsstufe.
30. if konsistent = true then begin
31.   if exists[PS[los]+1] then begin
32.     for each Los losPS ∈ Lose[PS[los]+1] with A[losPS] = A[los] do begin
33.       if Ende[losPS] > Start[los] - VLZ[E[los]] then begin
34.         konsistent = false
35.         break
36.       end
37.     end
38.   end
39. end
40. // Prüfe auf Überschneidung mit nachgelagerter Produktionsstufe.
41. if konsistent = true then begin
42.   if exists[PS[los]-1] then begin
43.     for each Los losPS ∈ Lose[PS[los]-1] with A[losPS] = A[los] do begin
44.       if Ende[los] > Start[losPS] - VLZ[E[losPS]] then begin
45.         konsistent = false
46.         break
47.       end
48.     end
49.   end
50. end
51. return konsistent

```

Anhang B: Bedarfsglättungsverfahren

Anhang B: Symbolverzeichnis

Symbol	Definition
b[erzeugnis, periode]	Bedarf von Erzeugnis in Makroperiode
endFP	Ende der Feinplanung ($T^{e,FP}$)
E[los]	Erzeugnistyp von Los
j	Verzugsmengen, zeitlich aufsteigend sortiert
L[los]	Produktionslinie, auf der das Los gefertigt wird
Linie[e]	Menge der gültigen Produktionslinien von Erzeugnis e
Lose[periode]	Alle Lose einer Makroperiode absteigend sortiert nach Alternativlinien
Menge[los]	Losgröße
TZ[erzeugnis,linie]	Takzeit von Erzeugnis auf Linie
periode	Gültige Makroperiode in einem Grobplanungslauf
Planungshorizont	Zeitraum aufsteigend von $T^{b,GP}$ bis $T^{e,GP}$

Anhang B: Pseudocode

Algorithmus Bedarfsglättungsverfahren(plan, b, j)

Eingabe: Produktionsplan "plan", Bedarf "b", Verzugslose "j"

Ausgabe: Angepasster Produktionsplan "plan", zu verschiebene Lose in die Feinplanung

Beschreibung: Bestimme Lose, die in den Feinplanungshorizont verschoben werden.

```

1.      j=1
2.      while j != null do begin
3.          for each periode P ∈ Planungshorizont do begin
4.              for each Los L ∈ Lose[P] do begin
5.                  if Linie[E[L]]∩Linie[E[j]] !=null then begin
6.                      for each periode T ∈ Planungshorizont do begin
7.                          if b[E[L],T] >0 then begin
8.                              if b[E[L],T]*TZ[E[L],L[L]]≥Menge[j]*TZ[E[j],L[L]] then begin
9.                                  b[E[L], endFP]=Ceiling(Menge[j]*TZ[E[j],L[L]]/TZ[E[L],L[L]])
10.                                 b[E[Los],T]=b[E[L],T]-Ceiling(Menge[j]*TZ[E[j],L[L]]/TZ[E[L],L[L]])
11.                                 break
12.                             else
13.                                 Menge[j]= Menge[j]-Floor(b[E[L],T]*TZ[E[L],L[L]]/TZ[E[j],L[L]])
14.                                 b[E[L],endFP]=b[E[L],T]
15.                                 b[E[L],T]=0
16.                             end
17.                         end
18.                     end
19.                 end
20.             end
21.         end
22.         j++
23.     end

```


Anhang C: Daten des Benchmarkings

Anhang C: Anwendungsszenario 1

Schichtname	Beginn	Ende	Leistungsgrad	Kosten
FS-0	00:00:00	08:00:00	0	0
SS-0	08:00:00	16:00:00	0	0
NS-0	16:00:00	23:59:59	0	0
FS-50	00:00:00	08:00:00	50	3.600
SS-50	08:00:00	16:00:00	50	3.200
NS-50	16:00:00	23:59:59	50	2.400
FS-100	00:00:00	08:00:00	100	7.200
SS-100	08:00:00	16:00:00	100	6.400
NS-100	16:00:00	23:59:59	100	4.800

Tabelle 10: AZ 1: Verfügbare Schichten mit Leistungsgrad und Kosten

Schichtmodellname	Schichtzuordnung			Kapazität	Kosten
SM 1	FS-0	SS-0	NS-0	0	0
SM 2	FS-100	SS-100	NS-100	1440	18.400
SM 3	FS-0	SS-100	NS-100	960	11.200
SM 4	FS-100	SS-0	NS-0	480	7.200
SM 5	FS-0	SS-100	NS-0	480	6.400
SM 6	FS-100	SS-100	NS-0	960	13.600
SM 7	FS-50	SS-50	NS-50	720	9.200
SM 8	FS-100	SS-100	NS-50	1200	16.000
SM 9	FS-50	SS-100	NS-100	1200	14.800

Tabelle 11: AZ 1: Schichtmodelle mit Schichtzuordnung

Produktionsstufe	Produktionslinie	Erzeugnis	Taktzeit (Min.)	Rüstzeit (Min.)	Lagerkosten (€ Min.)	Verzugskosten (€ Min.)	Vorlaufzeit (Min.)	Schichtmodelle
PS 2	PL 21	E 21 - E 40	12	0	0,002	0,2	0	SM 1 - SM 3
PS 2	PL 22	E 21 - E 40	12	0	0,002	0,2	0	SM 1 - SM 3
PS 1	PL 11	E 1 - E 10	12	0	0,003	0,3	0	SM 1 - SM 9
PS 1	PL 12	E 11 - E 20	12	0	0,003	0,3	0	SM 1 - SM 9

Tabelle 12: AZ 1: Aufbau des Produktionssystems (Schichtflexibilität)

Produktionsstufe	Produktionslinie	Erzeugnis	Taktzeit (Min.)	Rüstzeit (Min.)	Lagerkosten (€ Min.)	Verzugskosten (€ Min.)	Vorlaufzeit (Min.)	Schichtmodelle
PS 2	PL 21	E 21 - E 40	12	0	0,002	0,2	0	SM 1, SM 2
PS 2	PL 22	E 21 - E 40	12	0	0,002	0,2	0	SM 1, SM 2
PS 1	PL 11	E 1 - E 10	12	0/480	0,003	0,3	0	SM 1, SM 2
PS 1	PL 12	E 11 - E 20	12	0/480	0,003	0,3	0	SM 1, SM 2

Tabelle 13: AZ 1: Aufbau des Produktionssystems (Rüstzeitanalyse)

Parametername	Wert
Anzahl Bedarfe	30-50 (20-40 bei Rüstzeitanalyse)
Bedarf / Periode	40
Peaks einbauen	0
Erstellungswahrscheinlichkeit	0,05 (einheitlich für alle Erzeugnisse)

Tabelle 14: AZ 1: Konfiguration der Bedarfe

Parametername	Wert
Zeitlimit	3600 Sek.
Umlanungskosten	0
EpGap	0,00
Probe	2

Tabelle 15: AZ 1: Einstellungen des Modells der Feinplanung

Parametername	Wert
Flexibilität Umlanung	4
Flexibilität Schichtmodell	Max. Anzahl Schichtmodelle des Szenarios (9/ 2)
Verzug erlaubt	Ja
Max. Lossplittungen	3

Tabelle 16: AZ 1: Einstellungen des Eröffnungsverfahrens

Parametername	Wert
Flexibilität Umlanung	4
Flexibilität Schichtmodell	Max. Anzahl Schichtmodelle des Szenarios (9/ 2)
SA Abkühlungsfaktor	0.99
SA Initialtemperatur	100
SA Max. Iterationen	1.000
SA Max. Laufzeit	01:00:00
SA Länge Markov-Kette	10
TS Max. Iterationen	1.000
TS Max. Laufzeit	01:00:00
TS Länge Tabu-Liste	10

Tabelle 17: AZ 1: Einstellungen des Verbesserungsverfahrens

Anhang C: Anwendungsszenario 2

Schichtname	Beginn	Ende	Leistungsgrad	Kosten
FS-0	00:00:00	08:00:00	0	0
SS-0	08:00:00	16:00:00	0	0
NS-0	16:00:00	23:59:59	0	0
FS-50	00:00:00	08:00:00	50	3.600
SS-50	08:00:00	16:00:00	50	3.200
NS-50	16:00:00	23:59:59	50	2.400
FS-100	00:00:00	08:00:00	100	7.200
SS-100	08:00:00	16:00:00	100	6.400
NS-100	16:00:00	23:59:59	100	4.800

Tabelle 18: AZ 2: Verfügbare Schichten mit Leistungsgrad und Kosten

Schichtmodellname	Schichtzuordnung			Kapazität	Kosten
SM 1	FS-0	SS-0	NS-0	0	0
SM 2	FS-100	SS-0	NS-0	480	7.200
SM 3	FS-0	SS-100	NS-0	480	6.400
SM 4	FS-0	SS-0	NS-100	480	4.800
SM 5	FS-100	SS-100	NS-0	960	13.600
SM 6	FS-0	SS-100	NS-100	960	11.200
SM 7	FS-100	SS-0	NS-100	960	12.000
SM 8	FS-100	SS-100	NS-100	1440	18.400
SM 9	FS-50	SS-0	NS-0	240	3.600
SM 10	FS-0	SS-50	NS-0	240	3.200
SM 11	FS-0	SS-0	NS-50	240	2.400
SM 12	FS-50	SS-50	NS-0	480	6.800
SM 13	FS-0	SS-50	NS-50	480	5.600
SM 14	FS-50	SS-0	NS-50	480	6.000
SM 15	FS-50	SS-50	NS-50	720	9.200
SM 16	FS-100	SS-50	NS-50	960	12.800
SM 17	FS-50	SS-100	NS-50	960	12.400
SM 18	FS-50	SS-50	NS-100	960	11.600
SM 19	FS-100	SS-100	NS-50	1200	16.000
SM 20	FS-50	SS-100	NS-100	1200	14.800
SM 21	FS-100	SS-50	NS-100	1200	15.200

Tabelle 19: AZ 2: Schichtmodelle mit Schichtzuordnung

Produktionsstufe	Produktionslinie	Erzeugnis	Taktzeit (Min.)	Rüstzeit (Min.)	Lagerkosten (€ Min.)	Verzugskosten (€ Min.)	Vorlaufzeit (Min.)	Schichtmodelle
PS 3	PL 31	E 21 - E 30	12	120	0,001	0,1	0	SM 1 - SM 8
PS 3	PL 32	E 21 - E 30	12	120	0,001	0,1	0	SM 1 - SM 8
PS 3	PL 33	E 21 - E 30	12	120	0,001	0,1	0	SM 1 - SM 8
PS 2	PL 21	E 11 - E 20	12	60	0,002	0,2	60	SM 1 - SM 8
PS 2	PL 22	E 11 - E 20	12	60	0,002	0,2	60	SM 1 - SM 8
PS 1	PL 11	E 1 - E 10	12	0	0,003	0,3	60	SM 1 - SM 21
PS 1	PL 12	E 1 - E 10	12	0	0,003	0,3	60	SM 1 - SM 21

Tabelle 20: AZ 2: Aufbau des Produktionssystems

Parametername	Wert
Anzahl Bedarfe	100
Mindestbedarf / Periode	0
Peaks einbauen	5
Erstellungswahrscheinlichkeit (initial)	0,10 (einheitlich für alle Erzeugnisse)
Wahrscheinlichkeit Terminänderung	0,03
Wahrscheinlichkeit Mengenänderung	0,03
Wahrscheinlichkeit Stornierung	0,03
Wahrscheinlichkeit Eilauftrag	0,03

Tabelle 21: AZ 2: Konfiguration der Bedarfe

Parametername	Wert
Flexibilität Umplanung	7
Flexibilität Schichtmodell	21
Verzug erlaubt	Ja
Max. Lossplittungen	3

Tabelle 22: AZ 2: Einstellungen des Eröffnungsverfahrens

Parametername	Wert
Flexibilität Umplanung	7
Flexibilität Schichtmodell	21
SA Abkühlungsfaktor	0.99
SA Initialtemperatur	100
SA Max. Iterationen	1.000
SA Max. Laufzeit	00:02:00
SA Länge Markov-Kette	10
TS Max. Iterationen	1.000
TS Max. Laufzeit	00:02:00
TS Länge Tabu-Liste	10

Tabelle 23: AZ 2: Einstellungen des Verbesserungsverfahrens

Parametername	Wert
Anzahl Bedarfe	250
Mindestbedarf / Periode	0
Peaks einbauen	5
Erstellungswahrscheinlichkeit (initial)	0,10 (einheitlich für alle Erzeugnisse)
Wahrscheinlichkeit Terminänderung	0,03
Wahrscheinlichkeit Mengenänderung	0,03
Wahrscheinlichkeit Stormierung	0,03
Wahrscheinlichkeit Eilauftrag	0,03
Standardschichtmodell	8

Tabelle 24: AZ 2: Einstellungen der Grobplanung