

**Multikriterielle Ablaufplanung und -steuerung in
dynamischen und stochastischen Umgebungen**
**Ein Beitrag zur Erstellung robuster Ablaufpläne für die
Frachtabfertigung in Luftfrachtterminals**

Der Fakultät für Wirtschaftswissenschaften der
Universität Paderborn
zur Erlangung des akademischen Grades
DOKTOR DER WIRTSCHAFTSWISSENSCHAFTEN
– Doctor rerum politicarum –
vorgelegte Dissertation

von

Simon Boxnick, M. Sc.
geboren am 23.08.1986 in Meschede

Paderborn, Januar 2016

Dekanin: Prof. Dr. Caren Sureth-Sloane

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

Korreferentin: Prof. Dr. Leena Suhl

Für Sarah

Vorwort

Die vorliegende Arbeit ist das Resultat meiner Forschung, die ich als Stipendiat der *International Graduate School „Dynamic Intelligent Systems“* der Universität Paderborn am Lehrstuhl für *Wirtschaftsinformatik, insb. CIM* des Heinz Nixdorf Instituts in Kooperation mit der Firma *Lödige Industries GmbH* in Warburg durchgeführt habe.

Für die engagierte wissenschaftliche Betreuung und vielseitige Unterstützung möchte ich mich bei Herrn Prof. Dr.-Ing. habil. Wilhelm Dangelmaier bedanken. Er ermöglichte mir die Anfertigung dieser Arbeit in Kooperation mit einem Industrieunternehmen und gab mir dadurch die Möglichkeit, Forschung und industrielle Praxis zu verbinden. Diese wertvolle Erfahrung möchte ich nicht missen. Zudem gilt mein Dank Frau Prof. Dr. Leena Suhl für die Übernahme des Zweitgutachtens sowie Herrn Prof. Dr. Friedhelm Meyer auf der Heide und Herrn Prof. Dr. Stefan Betz, die mir als Mitglieder meiner Prüfungskommission wertvolles Feedback gaben.

Dem Team der *International Graduate School „Dynamic Intelligent Systems“* und ihrem Leiter Prof. Dr. Eckhard Steffen danke ich für die Förderung durch ein Stipendium. Der interdisziplinäre Austausch mit anderen Stipendiaten in den zahlreichen Veranstaltungen des Doktorandenprogramms hat meine Promotionszeit sehr bereichert.

Mein Dank gilt zudem Herrn Prof. Dr.-Ing. Rudolf Lödige sowie dem Geschäftsführer der *Lödige Industries GmbH*, Herrn Philippe De Backer, für die Gelegenheit zu einer Industriekooperation und für die Möglichkeit, mein Wissen in verschiedenen praxisnahen Projekten einbringen und erweitern zu können. Für die vielen konstruktiven Gespräche und Einblicke in die Welt der Luftfracht(-Terminals) möchte ich mich besonders bei Herrn Knud Segatz und Herrn Rainer Leichtweiss bedanken.

Meinen Kollegen am Lehrstuhl möchte ich für die motivierende und humorvolle Arbeitsatmosphäre sowie die interessanten fachlichen, aber auch nicht-fachlichen Gespräche danken. Es war eine schöne Zeit, die mir noch lange in Erinnerung bleiben wird. Mein besonderer Dank gilt meinen Bürokollegen und Mitstreitern bei *Lödige Industries GmbH*, Sebastian Lauck und Lukas Kopecki.

Mein tiefer Dank gilt meinen Eltern Verena und Helmuth, die mich während meiner wissenschaftlichen Ausbildung fortwährend unterstützt und in meinem Weg stets bestärkt sowie mir den notwendigen Rückhalt gegeben haben.

Mein innigster Dank gilt meiner Lebensgefährtin Sarah. Du warst stets mein größter Rückhalt und hast viel Verständnis für die wenige gemeinsame Zeit aufgebracht. Ich danke dir für deine Liebe und Unterstützung in den vergangenen Jahren, mit der du zur Entstehung dieses Buches sehr viel beigetragen hast.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Liste der Algorithmen	ix
Abkürzungsverzeichnis	xi
Symbolverzeichnis	xiii
1 Einleitung	1
1.1 Zielsetzung der Arbeit	3
1.2 Gliederung der Arbeit	4
2 Problemstellung	7
2.1 Systembeschreibung	7
2.1.1 Luftfrachttransportkette und involvierte Akteure	7
2.1.2 Prozessabläufe im Luftfrachtterminal	11
2.2 Problemfelder der Ablaufplanung und -steuerung	15
2.2.1 Datenanalyse der Abfertigungsprozesse	17
2.2.1.1 Vorhandene Datenbasis	17
2.2.1.2 Benötigter Informationsstand	19
2.2.2 Multikriterielle Ablaufplanung der Abfertigungsprozesse	20
2.2.2.1 Charakterisierung der Planungsumgebung	20
2.2.2.2 Darstellung des Ablaufplanungsproblems	21
2.2.2.3 Berücksichtigung der Personalkapazitäten	28
2.2.2.4 Berücksichtigung der Förderkapazitäten	30
2.2.3 Ablaufsteuerung der Abfertigungsprozesse	30
2.3 Anforderungen an die Verfahren zur Problemlösung	32
2.3.1 Anforderungen an ein Verfahren zur Datenanalyse der Abfertigungsprozesse	32
2.3.2 Anforderungen an ein Verfahren zur multikriteriellen Ablaufplanung der Abfertigungsprozesse	33
2.3.3 Anforderungen an ein Verfahren zur Ablaufsteuerung der Abfertigungsprozesse	34

3	Stand der Forschung	35
3.1	Datenanalyse und Wissensextraktion aus Datenquellen	35
3.1.1	Verfahren für den Aufgabenbereich der Klassenbildung	36
3.1.1.1	Hierarchische Verfahren	40
3.1.1.2	Partitionierende Verfahren	42
3.1.2	Verfahren für den Aufgabenbereich der Klassifizierung	44
3.1.2.1	Generative Modelle zur Klassifizierung	45
3.1.2.2	Diskriminative Modelle zur Klassifizierung	47
3.1.3	Literaturüberblick	49
3.2	Multikriterielle Ablaufplanung unter Unsicherheit	51
3.2.1	Exakte Verfahren	52
3.2.1.1	Robuste Optimierung	53
3.2.1.2	Stochastische Optimierung	54
3.2.2	Heuristische Verfahren	56
3.2.2.1	Multikriterielle populationsbasierte Metaheuristiken	57
3.2.2.2	Multikriterielle evolutionäre Metaheuristiken unter Unsicherheit	59
3.2.3	Ablaufplanung unter multiplen Ressourcenbeschränkungen	61
3.2.4	Literaturüberblick	62
3.3	Ablaufsteuerung und Revision von Ablaufplänen	69
3.3.1	Vorgehensweisen zur Revision von Ablaufplänen	69
3.3.2	Literaturüberblick	70
3.4	Fazit zum Stand der Forschung	71
4	Zu leistende Arbeit	75
4.1	Entwicklung eines Verfahrens zur Datenanalyse der Abfertigungsprozesse	75
4.2	Entwicklung einer multikriteriellen Ablaufplanung der Abfertigungsprozesse	76
4.3	Entwicklung eines Verfahrens zur Ablaufsteuerung der Abfertigungsprozesse	77
5	Konzeption	79
5.1	Komposition des Gesamtsystems in einer Architektur	79
5.1.1	Aufbau der Architektur	79
5.1.2	Ablaufprozess innerhalb der Architektur	81
5.2	Erstellung des benötigten Informationsstands aus historischen Betriebsdaten	82
5.2.1	Clusterbildung der Abfertigungsaufträge	82
5.2.1.1	Auftragsbeschreibende Eigenschaften	83
5.2.1.2	Distanzfunktion für Aufträge	85
5.2.1.3	Verfahren zur Clusteranalyse von Aufträgen	87
5.2.1.4	Intervallgrenzen der Bearbeitungszeitabweichungen	91
5.2.1.5	Ermittlung von Häufigkeitsverteilungen der Bearbeitungszeitabweichungen	94
5.2.2	Klassifikatorentwicklung für unbekannte Aufträge	97
5.2.2.1	Erstellung von Trainingsdaten	97
5.2.2.2	Erstellung des Klassifikators	99

5.3	Multikriterielle Ablaufplanung der Abfertigungsprozesse	105
5.3.1	Formalisierung des Ablaufplanungsproblems	105
5.3.1.1	Klassierung der zu bearbeitenden Aufträge	105
5.3.1.2	Zielsystem	107
5.3.1.3	Modellformulierung	110
5.3.2	Aufbau einer rollierenden Ablaufplanung	112
5.3.2.1	Ablauf der rollierenden Ablaufplanung	113
5.3.2.2	Zielkriterium „Stabilität“	114
5.3.2.3	Erweiterung der Modellformulierung	117
5.3.3	Evolutionäre Heuristik für die Ablaufplanung	120
5.3.3.1	Kodierung und Dekodierung der Ablaufpläne	121
5.3.3.2	Initialisierung	123
5.3.3.3	Evaluation der Ablaufpläne auf Basis des Zielsystems	127
5.3.3.4	Selektion und Rekombination von Ablaufplänen	130
5.3.3.5	Mutationsoperator für Ablaufpläne	133
5.3.3.6	Ersetzungsstrategie für die Ablaufpläne	134
5.3.3.7	Terminierung und Auswahl eines Ablaufplans durch den Entscheidungsträger	135
5.3.4	Integration der Personaleinsatzplanung	135
5.3.4.1	Erweiterung der Modellformulierung	136
5.3.4.2	Heuristik für die Personaleinsatzplanung	138
5.3.5	Integration der Transportsysteme	139
5.3.5.1	Erweiterung der Modellformulierung	139
5.3.5.2	Heuristik für die Materialflussteuerung	142
5.4	Ablaufsteuerung und Revision von Ablaufplänen	144
5.4.1	Ausführungsüberwachung des aktiven Ablaufplans	144
5.4.2	Ereignisorientierte Revision des aktiven Ablaufplans	145
6	Validierung	149
6.1	Validierung der Erstellung des benötigten Informationsstands	149
6.1.1	Testinstanzen für die Validierung des Clustering- und Klassifikationsverfahrens	149
6.1.2	Analyse des Clusteringverfahrens	151
6.1.2.1	Auswahl der Initialisierungsmethode	152
6.1.2.2	Parametrisierung der Anzahl Cluster	155
6.1.3	Analyse des Klassifikationsverfahrens	155
6.1.3.1	Effektivität und Effizienz des Klassifikationsverfahrens	155
6.1.3.2	Ermittlung von Kontingenztafeln	159
6.2	Validierung der multikriteriellen Ablaufplanung der Abfertigungsprozesse	161
6.2.1	Testinstanzen für die Validierung der multikriteriellen Ablaufplanung	161
6.2.2	Parametrisierungsexperiment für den genetischen Algorithmus	162
6.2.3	Effektivität und Effizienz der proaktiven Ablaufplanung	170
6.2.4	Betrachtung der rollierenden Ablaufplanung	173

7 Zusammenfassung und Ausblick	177
7.1 Zusammenfassung	177
7.2 Weitere Forschungsmöglichkeiten	179
Literaturverzeichnis	181

Abbildungsverzeichnis

2.1	Luftfrachttransportkette und involvierte Akteure	8
2.2	Frachtflüsse im Luftfrachtterminal	14
2.3	Dynamische Ankunft neuer Aufträge und ihre Berücksichtigung in partiell ausgeführten Ablaufplänen	25
2.4	Auswirkungen stochastischer Bearbeitungszeiten der Aufträge auf den Ab- laufplan	26
2.5	Auswirkungen der Ressource „Personal“ auf den Ablaufplan	29
5.1	Übersicht über die Architektur	80
5.2	Kumulative Verteilungs- und Quantilfunktion der Bearbeitungszeitabwe- ichungen im Intervall I_C	94
5.3	Aufbau einer Häufigkeitsverteilung, dargestellt als Histogramm	96
5.4	Rollierende Ablaufplanung	113
5.5	Visualisierung der Berechnung für die Zielfunktion „Stabilität“ unter Be- rücksichtigung eines beliebigen, aber festen Szenarios	116
5.6	Ablauf des genetischen Algorithmus	121
5.7	Dekodierung eines Individuums zu einem Ablaufplan	122
5.8	Visualisierung eines Worst-Case-Szenarios bei Anwendung des absoluten Regret-Kriteriums, das kein Extrempunktszenario ist	128
5.9	Rekombinationsoperator des genetischen Algorithmus	133
5.10	Mutationsoperatoren des genetischen Algorithmus	134
6.1	Verläufe des Silhouettenkoeffizienten für verschiedene Parametervariationen der Anzahl Cluster für die untersuchten Testinstanzen	156
6.2	Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Populationsgröße	165
6.3	Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Rekombinationswahrscheinlichkeit	166
6.4	Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation des Verhältnisses der ersten zur zweiten Mutationsart	167
6.5	Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Mutationswahrscheinlichkeit der ersten Mutationsart	168
6.6	Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Mutationswahrscheinlichkeit der zweiten Mutationsart	169
6.7	Pareto-Fronten der pareto-optimalen Ablaufpläne für drei Testinstanzen	174

Tabellenverzeichnis

2.1	Datenbasis über Frachtabfertigungsaufträge	18
6.1	Anzahl der historischen Aufträge in den realen und synthetischen Testinstanzen unterteilt nach den nicht (quasi)stetigen Eigenschaften	150
6.2	Arithmetisches Mittel und Populations-Standardabweichung der Bearbeitungszeitabweichung sowie der (quasi)stetigen Eigenschaften für die realen Testinstanzen	151
6.3	Silhouettenkoeffizient, Rand-Maß und Jaccard-Koeffizient der verschiedenen Testinstanzen in Abhängigkeit der gewählten Initialisierungsmethode	154
6.4	Metriken zur Bestimmung der Effektivität und Effizienz des Klassifikationsverfahrens für die verschiedenen Testinstanzen	159
6.5	Kontingenztafel für Testinstanz R2 nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung	160
6.6	Durch den Klassifikator ermittelte Zugehörigkeitswahrscheinlichkeiten für vier ausgewählte Zellen der Kontingenztafel von Testinstanz R2	160
6.7	Anzahl der einzuplanenden Aufträge in den realen und synthetischen Testinstanzen, unterteilt nach den nicht (quasi)stetigen Eigenschaften	162
6.8	Für die proaktive Ablaufplanung zur Verfügung stehende Ressourcen	162
6.9	Für die proaktive Ablaufplanung zu ermittelnde Parameterwerte und festgelegte Parameter	163
6.10	Arithmetisches Mittel der benötigten Zeiten zur Erstellung von 54 Szenarien und einer Initialpopulation \mathcal{P}_0 von 30 Individuen für drei Testinstanzen	164
6.11	Vergleich der Ablaufpläne RA, PA, EA und CA anhand von vier Metriken für die verschiedenen Testinstanzen	171
6.12	MIP-Statistiken des Modells zur Berechnung des Worst-Case-Szenarios nach dem Pre-Solve-Schritt für die verschiedenen Testinstanzen	173
6.13	Anzahl der Aufträge mit unterschiedlichem Ausführungsstatus für die verschiedenen Testinstanzen zum Zeitpunkt $t_1 = 8:30$ Uhr	173
6.14	Vergleich der Ablaufpläne RA, PA und SA anhand von sechs Metriken für die verschiedenen Testinstanzen	175

Liste der Algorithmen

5.1 Clusteranalyse von Aufträgen	92
5.2 Ermittlung der verkleinerten Clustermenge \mathcal{C}^*	95
5.3 Ermittlung der Häufigkeitsverteilungen der Bearbeitungszeitabweichungen .	98
5.4 Erstellung von Trainingsdaten für den Klassifikator	99
5.5 Klassifikator für Frachtabfertigungsaufträge	103
5.6 Klassierung der einzuplanenden Frachtabfertigungsaufträge und Ermittlung der Bearbeitungszeitintervalle	106
5.7 Erstellung der Initialpopulation für den genetischen Algorithmus	126
5.8 Ablauf der Evaluation einer Population	131

Abkürzungsverzeichnis

4PL	Fourth-Party-Logistics-Provider
A.-Plan	Ablaufplan
AGV	Automated Guided Vehicle
Anz.	Anzahl
AWB	Air Waybill
BD	Break-down
BU	Build-up
CA	Optimaler Ablaufplan für die geschätzte Bearbeitungszeit
CBT	Calibrated Boosted Trees
CLARA	Clustering Large Applications
CLARANS	Clustering Large Applications based upon Randomized Search
CPT	Conditional Probability Table
DM	Data-Mining
DRC	Dual-resource Constrained
DRCS	Dual-resource Constrained Systems/Scheduling
EA	Erwartungswertoptimaler Ablaufplan
EDF	Earliest Deadline First
ETV	Elevated Transfer Vehicle
GA	Genetischer Algorithmus
HAWB	House Air Waybill
IATA	International Air Transport Association
Inst.	Instanz
KDD	Knowledge Discovery in Databases
KDDM	Knowledge Discovery and Data Mining
KNN	Künstliche neuronale Netze
LO	Linear Optimization
LP	Linear Program
MAWB	Master Air Waybill

MCFP	Multi-commodity Flow Problem
MCNF	Multi-commodity Network Flow
MHS	Material Handling System
MILP	Mixed-integer Linear Program
MIP	Mixed-integer Program
MOEA	Multi-objective Evolutionary Algorithm
MOP	Multi-objective Optimization Problem
MRCS	Multiple-resource Constrained Systems/Scheduling
NLMIP	Non-linear Mixed-integer Program
NOTOC	Notification to Captain
NSGA-II	Nondominated Sorting Genetic Algorithm II
PA	Ablaufplan mit der geringsten Anzahl säumiger Aufträge
PAM	Partitioning Around Medoids
RA	Ablaufplan mit der höchsten Robustheit
RC	Robust Counterpart
RFS	Road Feeder Service
RGV	Rail-guided Vehicle
RTK	Revenue tonne kilometre
s. ed.	lat. sine editore (ohne Herausgeber)
s. l.	lat. sine loco (ohne Ort)
SA	Ablaufplan mit der höchsten Stabilität
SHC	Special Handling Code
u. B. d. N.	unter Beachtung der Nebenbedingungen
ULD	Unit Load Device

Symbolverzeichnis

\mathcal{A}	Menge von Aufträgen a
$\mathcal{A}_a^{\text{prec}}$	Menge aller Aufträge, die vor Auftrag a eingeplant werden können
$\mathcal{A}_a^{\text{succ}}$	Menge aller Aufträge, die nach Auftrag a eingeplant werden können
\mathcal{A}^{H}	Menge historischer Frachtabfertigungsaufträge ha
\mathcal{A}_{t_i}	Menge von Aufträgen a , die zum Zeitpunkt t_i bekannt sind. Es gilt: $\mathcal{A}_{t_i} \subset \mathcal{A}$ und $\mathcal{A}_{t_i} = \mathcal{A}_{t_{i+1}}^{\text{C}} \cup \mathcal{A}_{t_{i+1}}^{\text{R}} \cup \mathcal{A}_{t_{i+1}}^{\text{F}} \cup \mathcal{A}_{t_{i+1}}^{\text{P}}$
$\mathcal{A}_{t_i}^{\text{close}}$	Menge von Aufträgen a , die zum Zeitpunkt t_i bereits in einem Individuum η kodiert sind
$\mathcal{A}_{t_i}^{\text{open}}$	Menge von Aufträgen a , die zum Zeitpunkt t_i in einem Individuum η kodiert werden müssen
$\mathcal{A}_{t_{i+1}}^{\text{C}}$	Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, die bereits fertiggestellt wurden
$\mathcal{A}_{t_{i+1}}^{\text{F}}$	Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, für die an den Arbeitsstationen bereits eine Vorbereitung stattgefunden hat (z. B. Vorpufferung von Fracht). Aufträge dieser Menge sind somit fixiert
$\mathcal{A}_{t_{i+1}}^{\text{N}}$	Menge von Aufträgen a , die im Zeitraum von t_i bis t_{i+1} neu hinzugekommen sind
$\mathcal{A}_{t_{i+1}}^{\text{P}}$	Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, die lediglich auf eine Arbeitsstation eingeplant sind
$\mathcal{A}_{t_{i+1}}^{\text{Pb}}$	Menge der erstmalig einzuplanenden sowie neu einzuplanenden Aufträge zum Zeitpunkt t_{i+1} . Es gilt: $\mathcal{A}_{t_{i+1}}^{\text{Pb}} = \mathcal{A}_{t_{i+1}}^{\text{N}} \cup \mathcal{A}_{t_{i+1}}^{\text{P}}$
$\mathcal{A}_{t_{i+1}}^{\text{R}}$	Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, die bereits auf den Arbeitsstationen gestartet wurden
\mathcal{AP}	Menge aller zulässigen Ablaufpläne ap
\mathcal{AP}_i	Menge aller zulässigen Ablaufpläne ap zum Zeitpunkt t_i
α	Glättungsparameter für den Klassifikator
a	Auftrag $a \in \mathcal{A}$
ap	Ablaufplan $ap \in \mathcal{AP}$
ap_i^η	Ablaufplan, der sich aus dem Individuum η und einem evtl. bereits in der Ausführung befindlichen Ablaufplan ap_{i-1} ergibt
ap^{P}	Geplanter Ablaufplan
ap^{r}	Realisierter Ablaufplan
ap_{sz}^*	Optimaler Ablaufplan für Szenario $sz \in \mathcal{SZ}$, der zu der geringsten Anzahl säumiger Aufträge führt
$ap_{sz^{\text{co}}}^*$	Optimaler Ablaufplan für Szenario sz^{co} , der zu der geringsten Anzahl säumiger Aufträge führt

$ap_{sz^{ex}}^*$	Optimaler Ablaufplan für Szenario sz^{ex} , der zu der geringsten Anzahl säumiger Aufträge führt
$ap_{sz_{\eta}^{dis}}^*$	Optimaler Ablaufplan für das Ersatzszenario $sz_{\eta}^{dis} \in \mathcal{SZ}$ des Individuums η , der zu der geringsten Anzahl säumiger Aufträge führt
$ap_{sz^{lb}}^*$	Optimaler Ablaufplan für Szenario sz^{lb} , der zu der geringsten Anzahl säumiger Aufträge führt
$ap_{sz^{ub}}^*$	Optimaler Ablaufplan für Szenario sz^{ub} , der zu der geringsten Anzahl säumiger Aufträge führt
Bk	Basisklassifikator
\mathcal{C}	Menge von Clustern \mathcal{C}
\mathcal{C}^*	Menge von verkleinerten Clustern \mathcal{C}^*
\mathcal{C}	Cluster $\mathcal{C} \in \mathcal{C}$
\mathcal{C}^*	Verkleinertes Cluster $\mathcal{C}^* \in \mathcal{C}^*$, das nur Werte enthält, die kleiner als ein bestimmtes Quantil sind
\mathcal{C}_{td}^*	Cluster, das einem Trainingsdatum $td \in \mathcal{T}$ als Ausgabe zugeordnet ist
\mathcal{C}_z	Cluster, dessen Clusterzentrum $z \in \mathcal{Z}$ ist
\mathcal{C}_a	Fertigstellungszeitpunkt von Auftrag $a \in \mathcal{A}$
$C_a^{sz}(ap)$	Funktion zur Bestimmung des Fertigstellungszeitpunkts von Auftrag $a \in \mathcal{A}$ bei Verwendung des Ablaufplans $ap \in \mathcal{AP}$ in Szenario $sz \in \mathcal{SZ}$
$c^{sched}(ap)$	Funktion zur Berechnung der erwarteten gewichteten Anzahl säumiger Aufträge für einen gegebenen Ablaufplan $ap \in \mathcal{AP}$
$classify(a)$	Klassierungsfunktion für Auftrag $a \in \mathcal{A}$
$cost(\cdot)$	Einstellige Kostenfunktion
$\Delta\tau_{\mathcal{C}^*}$	Eine auf Basis der Häufigkeitsverteilung $\mathcal{H}_{\mathcal{C}^*}$ des Clusters $\mathcal{C}^* \in \mathcal{C}^*$ ermittelte Bearbeitungszeitabweichung
$\Delta\tau_{\mathcal{C}^*}^{\max}$	Maximaler Wert der Bearbeitungszeitabweichungen aller in einem Cluster $\mathcal{C}^* \in \mathcal{C}^*$ enthaltenen Aufträge
$\Delta\tau_{\mathcal{C}^*}^{\min}$	Minimaler Wert der Bearbeitungszeitabweichungen aller in einem Cluster $\mathcal{C}^* \in \mathcal{C}^*$ enthaltenen Aufträge
$\Delta\tau_{\mathcal{C}}^{\max}$	Maximaler Wert der Bearbeitungszeitabweichungen aller in einem Cluster $\mathcal{C} \in \mathcal{C}$ enthaltenen Aufträge
$\Delta\tau_{\mathcal{C}}^{\min}$	Minimaler Wert der Bearbeitungszeitabweichungen aller in einem Cluster $\mathcal{C} \in \mathcal{C}$ enthaltenen Aufträge
$\Delta\tau_{\mathcal{C}}^{pc}$	$p_{\mathcal{C}}$ -Quantil: Bearbeitungszeitabweichungen, für die gilt, dass $p_{\mathcal{C}} \cdot 100\%$ aller Bearbeitungszeitabweichungen des Clusters $\mathcal{C} \in \mathcal{C}$ kleiner sind
$\Delta c_{z,ha}^{medoid}$	Kostendifferenz durch die Vertauschung des Clusterzentrums $z \in \mathcal{Z}$ und des Auftrags $ha \in \mathcal{A}^H \setminus \mathcal{Z}$
$\Delta\tau_{ha}$	Bearbeitungszeitabweichung des historischen Frachtabfertigungsauftrags $ha \in \mathcal{A}^H$
d_a	Fälligkeitstermin von Auftrag $a \in \mathcal{A}$
$dist(\cdot, \cdot)$	Zweistellige Distanzfunktion
DM	Distanzmatrix

dm_{ha_i, ha_j}	Eintrag in der Distanzmatrix, der die Distanz zwischen den historischen Aufträgen $ha_i \in \mathcal{A}^H$ und $ha_j \in \mathcal{A}^H$ bezeichnet
\mathcal{E}	Menge von Vorrangbeziehungen. Wenn $(a_i, a_j) \in \mathcal{E}$, dann muss Auftrag $a_i \in \mathcal{A}$ vor dem Beginn von Auftrag $a_j \in \mathcal{A}$ fertiggestellt sein. Es gilt: $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$
\mathcal{E}^{Br}	Menge der Transportstrecken im Break-Netzwerk des Transportsystems
\mathcal{E}^{Bu}	Menge der Transportstrecken im Build-Netzwerk des Transportsystems
\mathcal{E}_η	Menge von Vorrangbeziehungen für den Ablaufplan, der durch das Individuum η kodiert wird. Entspricht einer Erweiterung der Menge \mathcal{E} um Vorrangbeziehungen für die Aufträge, die auf der gleichen Maschine eingeplant sind
\mathcal{E}_η^+	Transitive Hülle der Menge \mathcal{E}_η
\mathcal{EU}_p	Menge der Eigenschaftswerte der Eigenschaft p aller historischen Aufträge $ha \in \mathcal{A}^H$
\vec{e}_{ha}	Eigenschaftsvektor des historischen Auftrags $ha \in \mathcal{A}^H$
$e_{ha,p}$	Wert der Eigenschaft p des historischen Auftrags $ha \in \mathcal{A}^H$
e_p	Der normalisierte Wert von Eigenschaft p
e_p^*	Der nicht normalisierte Wert von Eigenschaft p
$e_{td,p}$	Wert der Eigenschaft p , die als Eingabe des Trainingsdatums $td \in \mathcal{T}$ dient
ev	Ein Ereignis ev , das zum Zeitpunkt t_{ev} eingetreten ist
\mathcal{F}	Liste (hier auch als Front bezeichnet) nicht dominierter bzw. pareto-optimaler Individuen
\mathcal{F}_{ha}	Menge der Fracht f des historischen Auftrags $ha \in \mathcal{A}^H$
\mathcal{F}_{ha}^{mh}	Menge der Fracht f des historischen Auftrags $ha \in \mathcal{A}^H$, die mit Hilfsmitteln bearbeitet werden muss
\mathcal{F}_{ha}^{SHC}	Menge der Fracht f des historischen Auftrags $ha \in \mathcal{A}^H$, die einen Special Handling Code (SHC) besitzt
$\widehat{F}_{\mathcal{C}}(x)$	Empirische kumulative Verteilungsfunktion des Clusters $\mathcal{C} \in \mathcal{C}$
φ	Schwellenwertparameter für den absoluten Regret, der in möglichst vielen Szenarien eingehalten werden sollte
f_a	Zeitpunkt, zu dem Vorbereitungen für Auftrag a an der Arbeitsstation begonnen haben
f_{ha}	Flugzeugladeeinheit, die dem historischen Auftrag ha zugeordnet ist
\mathcal{G}	Vorranggraph, der aus einer Menge von Aufträgen \mathcal{A} als Knoten und einer Menge von Vorrangbeziehungen \mathcal{E} als Kanten besteht
\mathcal{G}^{Br}	Graph (Break-Netzwerk) zur Repräsentation des Transportsystems für Abbauaufträge, der aus einer Menge von Knoten \mathcal{V}^{Br} und einer Menge von Kanten \mathcal{E}^{Br} besteht
\mathcal{G}^{Bu}	Graph (Build-Netzwerk) zur Repräsentation des Transportsystems für Aufbauaufträge, der aus einer Menge von Knoten \mathcal{V}^{Bu} und einer Menge von Kanten \mathcal{E}^{Bu} besteht
\vec{g}_η	Vektor der Gene (hier auch als Chromosom bezeichnet) des Individuums η

g	Ein Gen eines Individuums η
g_{\max}^{mh}	Grenzwert für die einzelnen Frachtstückgewichte, ab dem diese mit einem Hilfsmittel bearbeitet werden müssen
$\text{gew}(\cdot)$	Einstellige Gewichtsfunktion
$\mathcal{H}_{\mathcal{C}^*}$	Relative Häufigkeitsverteilung der Bearbeitungszeitabweichungen für Cluster $\mathcal{C}^* \in \mathcal{C}^*$
$h_{\Delta\tau_{ha}}$	Absolute Häufigkeit einer bestimmten Bearbeitungszeitabweichung
ha	Historischer Frachtabfertigungsauftrag $ha \in \mathcal{A}^{\text{H}}$
I_a	Intervall der Bearbeitungszeiten eines Auftrags $a \in \mathcal{A}$. Es gilt: $I_a = [\tau_a^{\min}, \tau_a^{\max}]$
I_a^{Rest}	Restintervall der Bearbeitungszeiten eines Auftrags $a \in \mathcal{A}$
$I_{\mathcal{C}}$	Intervall der Bearbeitungszeitabweichungen der historischen Aufträge des Clusters $\mathcal{C} \in \mathcal{C}$. Es gilt: $I_{\mathcal{C}} = [\Delta\tau_{\mathcal{C}}^{\min}, \Delta\tau_{\mathcal{C}}^{\max}]$
$I_{\mathcal{C}^*}$	Verkleinertes Intervall der Bearbeitungszeitabweichungen der historischen Aufträge des Clusters $\mathcal{C}^* \in \mathcal{C}^*$. Es gilt: $I_{\mathcal{C}^*} = [\Delta\tau_{\mathcal{C}^*}^{\min}, \Delta\tau_{\mathcal{C}^*}^{\text{pc}}]$
η	Individuum $\eta \in \mathcal{P}$
\mathcal{K}	Menge aller Typen von Ladeeinheiten
\mathcal{K}_a^-	Menge an Typen von Ladeeinheiten, die von Auftrag $a \in \mathcal{A}$ benötigt werden
\mathcal{K}_a^+	Menge an Typen von Ladeeinheiten, die von Auftrag $a \in \mathcal{A}$ angeboten werden
κ	Cohens Kappa. Statistisches Maß für die Interrater-Reliabilität der Klassierungen von zwei Ratern
$kbr_{\mathcal{C}^*}$	Klassenbreite der Klassen des Clusters $\mathcal{C}^* \in \mathcal{C}^*$
\mathcal{L}_m	Liste von Aufträgen a , die auf Maschine m eingeplant werden sollen. Die Liste wird zur Erstellung eines Individuums verwendet.
\mathcal{L}	Menge der Knoten im Netzwerk, an denen Ladeeinheiten gelagert werden können. Es gilt: $\mathcal{L} \subset \mathcal{V}^{\text{Br}}$ bzw. $\mathcal{L} \subset \mathcal{V}^{\text{Bu}}$
$\mathcal{L}\mathcal{S}_{ha}$	Bei Aufbauaufträgen Menge von Ladeeinheiten im Sendungslager, in denen sich die Fracht für den Aufbauauftrag $ha \in \mathcal{A}^{\text{H}}$ befindet. Bei Abbauaufträgen Menge der Ladehilfsmittel des Sendungslagers, die verwendet werden, um die Fracht der Flugzeugladeeinheit zu vereinzeln
$lb_{TI_i^{\mathcal{C}^*}}$	Untere Grenze des Teilintervalls bzw. der Klasse TI_i des Clusters $\mathcal{C}^* \in \mathcal{C}^*$
\mathcal{M}	Menge von Maschinen (hier synonym: Arbeitsstationen) m
\mathcal{M}_a	Menge von Maschinen m , auf denen Auftrag $a \in \mathcal{A}$ ausgeführt werden kann. Es gilt: $\mathcal{M}_a \subseteq \mathcal{M}$
$\mu_{p,i}$	Erwartungswert für den Wert der Eigenschaft p in Cluster $\mathcal{C}_i^* \in \mathcal{C}^*$
$\hat{\mu}_{p,i}$	Geschätzter Erwartungswert für den Wert der Eigenschaft p in Cluster $\mathcal{C}_i^* \in \mathcal{C}^*$
m	Maschine (hier synonym: Arbeitsstation) $m \in \mathcal{M}$

\mathcal{N}	Normalverteilung
\mathcal{NE}	Menge von Knoten, die Neuronen in einem künstlichen neuronalen Netz simulieren
\mathcal{NE}^H	Menge von verdeckten Knoten bzw. Neuronen, die die verdeckte Schicht bzw. die verdeckten Schichten eines künstlichen neuronalen Netzes bilden
\mathcal{NE}^{In}	Menge von Eingangsknoten bzw. Neuronen, die die Eingabeschicht eines künstlichen neuronalen Netzes bilden
$\mathcal{NE}^{\text{Out}}$	Menge von Ausgangsknoten bzw. Neuronen, die die Ausgabeschicht eines künstlichen neuronalen Netzes bilden
\mathcal{NF}	Menge von Normalisierungsfaktoren nf_p , für jede Eigenschaft p
n_{c^*}	Parameter für die Anzahl Teilintervalle bzw. Klassen in die das Intervall I_{c^*} aufgeteilt wird
nf_p	Normalisierungsfaktor $nf_p \in \mathcal{NF}$ der Eigenschaft p
ω	Schwellenwertparameter für den absoluten Regret, der in allen Szenarien eingehalten werden muss
\mathcal{P}	Liste (hier auch als Population bezeichnet) von Individuen η
\mathcal{P}^*	Menge pareto-optimaler Lösungen
\mathcal{PF}^*	Pareto-Front der Menge pareto-optimaler Lösungen \mathcal{P}^*
$P(\cdot)$	Wahrscheinlichkeit
$P(\cdot \cdot)$	Bedingte Wahrscheinlichkeit
$\hat{P}(\cdot \cdot)$	Geschätzte bedingte Wahrscheinlichkeit
p_c	Parameter für den Prozentsatz des Quantils für Cluster \mathcal{C}
$\hat{p}_{c^*}(ha)$	Empirische Wahrscheinlichkeitsfunktion der Bearbeitungszeitabweichungen des Clusters $\mathcal{C}^* \in \mathcal{C}^*$ für einen historischen Auftrag $ha \in \mathcal{A}^H$
p^{cx}	Parameter zur Festlegung der Rekombinationswahrscheinlichkeit
p^{mut}	Parameter zur Festlegung des Verhältnisses von der ersten zur zweiten Mutationsart
pa^{mut}	Parameter zur Festlegung der Mutationswahrscheinlichkeit der ersten Mutationsart
ph	Planungshorizont
PM	Mustermatrix
pm^{mut}	Parameter zur Festlegung der Mutationswahrscheinlichkeit der zweiten Mutationsart
$popsize$	Parameter zur Festlegung der Populationsgröße
$pos(a, \eta)$	Funktion zur Bestimmung der Position eines Auftrags a auf einer Maschine in dem durch das Individuum η kodierten Ablaufplan
$pos(ps_i)$	Funktion zur Bestimmung der Position eines Partitionierungssymbols ps_i im Chromosom eines Individuums
ps	Partitionierungssymbol innerhalb des Chromosoms eines Individuums η
pz	Planungszyklus
\mathcal{Q}	Liste (auch: Nachfolgepopulation) von Individuen η , die durch Rekombination und Mutation aus der Population \mathcal{P} gewonnen werden

$Q(p)$	Quantilfunktion
$q_{TI_i^{C^*}}$	Relative Häufigkeit einer Klasse $TI_i^{C^*}$ des Clusters $C^* \in \mathcal{C}^*$
\mathcal{R}	Liste (hier auch als Population bezeichnet) von Individuen η , die durch die Vereinigung der Populationen \mathcal{P} und \mathcal{Q} erzeugt wird
$R(ap, sz)$	Funktion zur Berechnung des absoluten Regrets für den Ablaufplan $ap \in \mathcal{AP}$ in Szenario $sz \in \mathcal{SZ}$
$R^{\max}(ap)$	Funktion zur Berechnung des maximalen absoluten Regrets für den Ablaufplan $ap \in \mathcal{AP}$
ρ	Pseudoauftrag, dessen Bearbeitungszeit, Bereitstellungszeit sowie Gewicht 0 ist. Es gilt: $\rho \in \mathcal{A}$
r_a	Bereitstellungs- bzw. Einlastzeit von Auftrag $a \in \mathcal{A}$
$r_\varphi^{\text{sched}}(ap)$	Funktion zur Ermittlung der Robustheit eines Ablaufplans $ap \in \mathcal{AP}$ bei Verwendung des Schwellenwertparameters φ
\mathcal{SZ}	Menge von Szenarien sz
$\sigma_{p,i}^2$	Varianz des Wertes der Eigenschaft p in Cluster $C_i^* \in \mathcal{C}^*$
$\hat{\sigma}_{p,i}^2$	Korrigierte Stichprobenvarianz des Wertes der Eigenschaft p in Cluster $C_i^* \in \mathcal{C}^*$
s_a	Tatsächlicher Startzeitpunkt des Auftrags $a \in \mathcal{A}$
$s^{\text{sched}}(ap_{i-1}, ap_i)$	Funktion zur Berechnung der erwarteten Stabilität für einen gegebenen Ablaufplan ap_i unter Berücksichtigung des bereits gestarteten Ablaufplans ap_{i-1}
sk_η	Strafkosten für den durch das Individuum η kodierten Ablaufplan aufgrund der Überschreitung des Parameters ω
sz	Szenario $sz \in \mathcal{SZ}$
sz_{ap}^{worst}	Das Worst-Case-Szenario für Ablaufplan $ap \in \mathcal{AP}$
sz^{co}	Szenario, bei dem jeder Auftrag $a \in \mathcal{A}$ die über eine Formel geschätzte Bearbeitungszeit annimmt
sz^{ex}	Szenario, bei dem jeder Auftrag $a \in \mathcal{A}$ den Erwartungswert $E(\tau_a)$ bzw. $E(\tau_a^{\text{Rest}})$ annimmt
sz_η^{dis}	Ersatzszenario für sz_η^{worst} , das möglichst ungünstig für den Ablaufplan $ap_i^\eta \in \mathcal{AP}$ ist
sz_η^{worst}	Das Worst-Case-Szenario für den durch das Individuum η kodierten Ablaufplan $ap_i^\eta \in \mathcal{AP}$
sz^{lb}	Szenario, bei dem jeder Auftrag $a \in \mathcal{A}$ seine kürzeste Bearbeitungszeit aus dem Intervall I_a bzw. Restbearbeitungszeit aus dem Intervall I_a^{Rest} annimmt
sz^{ub}	Szenario, bei dem jeder Auftrag $a \in \mathcal{A}$ seine längste Bearbeitungszeit aus dem Intervall I_a bzw. Restbearbeitungszeit aus dem Intervall I_a^{Rest} annimmt
\mathcal{T}	Menge von Trainingsdaten td
\mathcal{T}_{C^*}	Menge von Trainingsdaten für das Cluster C^*

$TI_i^{C^*}$	Das i -te Teilintervall bzw. die i -te Klasse des Clusters C^* . Es gilt: $TI_i^{C^*} = [lb_{TI_i^{C^*}}, ub_{TI_i^{C^*}})$
θ	Schwellenwert bei künstlichen neuronalen Netzwerken
τ_a	Bearbeitungszeit eines Auftrags $a \in \mathcal{A}$. Diese wird auf Basis der Häufigkeitsverteilung \mathcal{H}_{C^*} ermittelt, die in dem Cluster C^* hinterlegt ist, zu dem der Auftrag klassiert wurde
$\widehat{\tau}_a$	Geschätzte Bearbeitungszeit eines Auftrags $a \in \mathcal{A}$
τ_a^{\max}	Maximale Bearbeitungszeit eines Auftrags $a \in \mathcal{A}$
τ_a^{\min}	Minimale Bearbeitungszeit eines Auftrags $a \in \mathcal{A}$
τ_a^{Rest}	Restbearbeitungszeit eines gestarteten Auftrags $a \in \mathcal{A}$. Diese wird auf Basis der Häufigkeitsverteilung \mathcal{H}_{C^*} ermittelt, die in dem Cluster C^* hinterlegt ist, zu dem der Auftrag klassiert wurde
τ_a^{sz}	Bearbeitungszeit eines Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
τ_{ha}	Tatsächliche bzw. realisierte Bearbeitungszeit eines historischen Auftrags $ha \in \mathcal{A}^H$
$\widehat{\tau}_{ha}$	Geschätzte Bearbeitungszeit eines historischen Auftrags $ha \in \mathcal{A}^H$, bevor dieser bearbeitet wurde
td	Trainingsdatum $td \in \mathcal{T}$. Es gilt: $td = (\vec{e}_{ha}, C^*)$ mit $ha \in C^*$ und $C^* \in \mathcal{C}^*$
\mathcal{U}	Unsicherheitsmenge in der robusten Optimierung
$U(ap, sz)$	Funktion zur Berechnung der gewichteten Anzahl säumiger Aufträge, wenn der Ablaufplan $ap \in \mathcal{AP}$ unter Szenario $sz \in \mathcal{SZ}$ ausgeführt wird
U_a	Straf- oder Säumniskosten von Auftrag $a \in \mathcal{A}$
$U_a^{sz}(ap)$	Funktion zur Bestimmung der Straf- oder Säumniskosten von Auftrag $a \in \mathcal{A}$ bei Verwendung des Ablaufplans $ap \in \mathcal{AP}$ in Szenario $sz \in \mathcal{SZ}$
U_{sz}^*	Minimale Straf- oder Säumniskosten in Szenario $sz \in \mathcal{SZ}$
$ub_{TI_i^{C^*}}$	Obere Grenze des Teilintervalls bzw. der Klasse TI_i des Clusters C^*
\mathcal{V}^{Br}	Menge der Knoten im Break-Netzwerk des Transportsystems
\mathcal{V}^{Bu}	Menge der Knoten im Build-Netzwerk des Transportsystems
$\text{vol}(\cdot)$	Einstellige Volumenfunktion
\mathcal{W}	Menge von Gewichtungsfaktoren w_p für jede Eigenschaft p
\mathcal{WO}	Menge von Mitarbeitern wo
\mathcal{WO}^{Br}	Menge von Mitarbeitern, die ausschließlich Abbauaufträge bearbeiten können. Es gilt: $\mathcal{WO}^{\text{Br}} \subset \mathcal{WO} \setminus \mathcal{WO}^{\text{Bu}}$
\mathcal{WO}^{Bu}	Menge von Mitarbeitern, die sowohl für Abbau- als auch Aufbauaufträge eingesetzt werden können. Es gilt: $\mathcal{WO}^{\text{Bu}} \subset \mathcal{WO} \setminus \mathcal{WO}^{\text{Br}}$
w_a	Gewichtungsfaktor des Auftrags $a \in \mathcal{A}$ zur Beschreibung der Priorität
w_p	Gewichtungsfaktor $w_p \in \mathcal{W}$ der Eigenschaft p
wo	Mitarbeiter $wo \in \mathcal{WO}$
\mathcal{Z}	Menge von Clusterzentren. Es gilt: $\mathcal{Z} \subset \mathcal{A}^H$
z	Clusterzentrum $z \in \mathcal{Z}$

1 Einleitung

Gegenüber der Fähigkeit, die Arbeit eines einzigen Tages sinnvoll zu ordnen, ist alles andere im Leben ein Kinderspiel.

Johann Wolfgang von Goethe
(1749 – 1832)

Luftfracht ist ein essentieller Bestandteil globaler Supply-Chains, wenn es um die Geschwindigkeit bei der Bereitstellung von wertvollen oder zeitkritischen Industrie- und Konsumgütern über mittlere bis weite Entfernungen geht. Die anhaltende Globalisierung, eine weltweit verteilte Produktion und verkürzte Produktlebenszyklen mit den einhergehenden stark reduzierten Lagerbeständen verstärken den Bedarf nach einer zeitnahen Bereitstellung von Gütern aller Art zusätzlich. Mit jährlichen Wachstumsraten von 2,7 % seit 2003 und einem prognostizierten jährlichen Wachstum zwischen 4,1 % und 5,6 % bis 2033 wird sich das weltweite Luftfrachtaufkommen von 200 Milliarden RTK¹ im Jahr 2013 auf voraussichtlich 512,3 Milliarden RTK im Jahr 2033 erhöhen.² Dabei wird mit weniger als 3 % gemessen am Volumen des weltweiten Frachtaufkommens zwar nur ein geringer Teil mittels Luftfracht transportiert,³ aber mit einem Wert von 30–35 % am Welthandel ist dies ein bedeutender Teil.⁴

Dieses zukünftige Wachstum stellt die Luftfrachtindustrie vor besondere Herausforderungen. Zum einen belasten steigende Kerosinpreise und neue Sicherheitsanforderungen die Gewinnspannen der Akteure in der Luftfrachtindustrie.⁵ Zum anderen besteht die Notwendigkeit, das gesteigerte Frachtaufkommen mit der für den Kunden gewohnten Servicequalität,⁶ in einem klassischerweise hinsichtlich der Kapazität stark beschränkten System,⁷ zu bearbeiten. Dabei entfällt auf die reine Flugzeit mit ca. 7–10 % der Gesamttransportzeit der Luftfracht ein eher geringer Anteil.⁸ Die Prozesse am Boden im Rahmen des Vor- und Nachlaufs sowie der Umschlag und die Zollabfertigung im Luftfrachtterminal machen hingegen mit 80–90 % den Großteil der gesamten Transportzeit aus.⁹

¹RTK (engl. *revenue tonne kilometre*) berechnet sich aus dem Produkt des Frachtgewichts (gemessen in Tonnen) und der zurückgelegten Strecke (gemessen in km).

²Siehe [Boe14], S. 13.

³Siehe [Sal13], S. 1.

⁴Siehe [Sal13], S. 1; [Air13], S. 5.

⁵Siehe [PKM11], S. 221.

⁶Hier sind insbesondere Zuverlässigkeit, Qualität und Geschwindigkeit bei der Frachtauslieferung sowie die Beförderungskosten für die Fracht zu nennen (vgl. [PKM11], S. 223; [Boe14], S. 11).

⁷Siehe [Pet07], S. 44.

⁸Siehe [JS00], S. 301.

⁹Siehe [BG12], S. 136; [Koe04], S. 321.

Die Luftfrachtterminals nehmen hierbei eine besondere Stellung innerhalb der Supply-Chain ein. Diese Terminals dienen als Schnittstelle zwischen dem Land- und Luftverkehr. Sie erfüllen typischerweise Umschlags- und Konsolidierungsleistungen (Wechsel von einem Ladehilfsmittel zu einem anderen bzw. Konsolidierung der Fracht auf bzw. in einem größeren Ladehilfsmittel), Sortierung der Fracht, temporäre Lagerung zur Entkopplung der unterschiedlichen Lastsituationen auf der Land- und Luftseite des Terminals sowie die Unterstützung von Aufgaben im Rahmen der Dokumentations- und Zollvorschriften.¹⁰ Diese Leistungen sind im Hinblick auf ihren direkten Einfluss auf die vom Kunden empfundene Servicequalität möglichst effizient auszuführen.

Den dargestellten Entwicklungen wird in den letzten Jahren mit einer zunehmenden Automatisierung der bedeutenden Luftfrachtterminals begegnet. Die Automatisierung erstreckt sich hierbei sowohl auf die Lager- und Transport- als auch Kommissioniersysteme. Dies hat zum einen den Vorteil, dass die zur Verfügung stehende Gebäudegrundfläche des Terminals effizienter ausgenutzt werden kann. So kann beispielsweise durch die Verwendung von Transferfahrzeugen (engl. *Transfer Vehicles*) in Kombination mit Aufzügen bzw. durch die Verwendung von Regalbediengeräten (engl. *Elevated Transfer Vehicles* (ETVs)) eine Lagerung von Fracht in mehreren Gebäudeebenen erfolgen.¹¹ Zum anderen ermöglicht die direkte fördertechnische Anbindung der manuell bedienten Arbeitsstationen zum Auf- und Abbau der Luftfrachtcontainer und -paletten an das Lager eine zuverlässige Frachtbearbeitung. Die Verwendung von Informationssystemen und insbesondere die mobile Datenerfassung und -übertragung unterstützen hierbei die Prozesse zur physischen und dokumentarischen Frachtabfertigung.¹²

Vorhandene Überkapazitäten der Luftfrachtterminals konnten dabei in der Vergangenheit ineffiziente Prozesse sowohl *verbergen* als auch *kompensieren*. Mit der Zunahme des Frachtaufkommens und dem begrenzten Platzangebot der Terminals an großen Flughäfen werden ineffiziente Prozesse mit den sinkenden Überkapazitäten jedoch evident. Diese Prozesse sind daher dahingehend zu optimieren, dass für den Kunden ein besseres Leistungsangebot zu geringeren Kosten angeboten bzw. das gewohnte Niveau gehalten wird.¹³ Innerhalb des Terminals lassen sich Umschlagprozesse für den Export, Import sowie den Transfer unterscheiden, in denen Frachtabfertigungsaufträge bearbeitet werden.¹⁴

Die Effizienz, mit der diese Prozesse ausgeführt werden, hängt maßgeblich von der Ablaufplanung und -steuerung ab, die die räumliche und zeitliche Koordination der Fracht-abfertigungsaufträge unter Berücksichtigung der zur Verfügung stehenden Ressourcen¹⁵ vornimmt und dessen Ergebnis ein Ablaufplan ist. Bei der Planung sind vielfältige Restriktionen zu beachten, die sich insbesondere aus dem festgelegten Flugplan und der

¹⁰Siehe [AMW11], S. 457 f.

¹¹Siehe ebd., S. 459.

¹²Siehe [AIK⁺08], S. 776.

¹³Siehe [RG09], S. 725.

¹⁴Siehe [Fry11], S. 42; [Men13], S. 708.

¹⁵Die zur Bearbeitung eines Frachtabfertigungsauftrags notwendigen Ressourcen sind zum einen eine Arbeitsstation sowie das notwendige Personal für den Auf- und Abbau der Luftfrachtcontainer und -paletten, zum anderen ein automatisches Lager- und Materialflusssystem zur Versorgung der Arbeitsstationen mit der zu bearbeitenden Fracht.

dem Kunden zugesicherten Frachtverfügbarkeit innerhalb eines bestimmten Zeithorizontes nach der Landung des Flugzeugs ergeben und die somit die spätesten Endzeitpunkte für die Frachtbearbeitung determinieren. Des Weiteren sind bei der Ablaufplanung das dynamische und stochastische (auch: nicht-deterministische) Planungsumfeld zu berücksichtigen.

Es ist ersichtlich, dass es sich bei der Ablaufplanung und -steuerung um eine komplexe Aufgabe handelt und die vielfältigen Aspekte durch eine manuelle, d. h. durch einen Menschen durchgeführte Planung, nur unzureichend berücksichtigt werden können. Eine automatische bzw. den Menschen unterstützende Planung auf Basis von Modellen und unter Verwendung exakter mathematischer Verfahren oder Heuristiken findet hingegen eine geringe Akzeptanz in der Praxis, solange die zuvor adressierten Anforderungen nicht in akzeptabler Weise abgebildet und berücksichtigt werden können. So ist die aufwendige Bestimmung eines optimalen Ablaufplans nicht hilfreich, wenn durch die fehlende Beachtung von z. B. stochastischen Einflussgrößen der optimale Plan nicht realisiert werden kann. Aus dem zuvor Geschriebenen begründet sich die Zielsetzung der vorliegenden Arbeit.

1.1 Zielsetzung der Arbeit

In dieser Arbeit soll ein Ablaufplanungs- und -steuerungssystem für die im Rahmen der Umschlagprozesse in Luftfrachtterminals durchzuführenden Frachtabfertigungsaufträge unter Berücksichtigung des dynamischen und stochastischen Planungsumfeldes konzipiert werden. Mit Hilfe der zu entwickelnden Verfahren sollen Ablaufpläne automatisiert erstellt und durchgesetzt werden können, die trotz der stochastischen Einflüsse des Planungsumfeldes gute bzw. für den Entscheidungsträger eine im Vorhinein,¹⁶ d. h. vor Bekanntwerden der tatsächlich eintretenden Umweltsituation, mit Sicherheit bestimmbare Qualität besitzen. Des Weiteren sind, aufgrund des dynamischen Planungsumfeldes, einmal getroffene Entscheidungen in dem Ablaufplan ggf. mehrfach zu revidieren. Dies soll unter Berücksichtigung der Planstabilität erfolgen. Ablaufpläne, die diese Kriterien erfüllen, werden im Folgenden als *robust* und *stabil* bezeichnet.

Die nicht-deterministischen Einflüsse des Planungsumfeldes können sich in vielfältiger Weise auf einen erstellten Ablaufplan auswirken, z. B. in Form von unsicheren Bearbeitungszeiten der Frachtabfertigungsaufträge. Aus dem angestrebten Ziel, durch eine Ablaufplanung robuste Ablaufpläne zu erstellen, folgt daher die Notwendigkeit, die Unsicherheiten, die bei der Planung in Form von Parametern beachtet werden sollen, in adäquater Weise zu modellieren. Ausgangspunkt für Informationen über diese Unsicherheiten stellen dabei die zuvor beschriebenen Informationssysteme dar, durch die eine umfangreiche Datenbasis über vergangene Frachtabfertigungsaufträge zur Verfügung steht. Aus dieser Datenbasis ist ein Informationsstand zu generieren, der eine Quantifizierung unsicherer Parameter ermöglicht. Im Hinblick auf sich evtl. ändernde

¹⁶Die Qualität eines durch klassische Optimierungsverfahren ermittelten optimalen Ablaufplans ist zwar auch im Vorhinein bekannt, kann sich aber bereits bei einer kleinen Änderung der verwendeten Parameter signifikant ändern.

Prozessabläufe sollte der Informationsstand adaptierbar sein, um stets die tatsächlich bestehenden Unsicherheiten adäquat abbilden zu können.

Zur Erstellung robuster Ablaufpläne ist ein Ablaufplanungsverfahren notwendig, das den ermittelten Informationsstand berücksichtigen kann. Hierbei ist zu einem gewissen Grad ein Zielkonflikt zwischen einem hinsichtlich eines Qualitätsziels optimalen, einem robusten und einem stabilen Ablaufplan zu erwarten, da die Absicherung gegen Unsicherheiten evtl. mit einer Verminderung des Qualitätsziels einhergeht und ein stabiler Ablaufplan ggf. den Handlungsspielraum für Umplanungen einschränkt. Daher ist von einem multikriteriellen und konfliktären Zielsystem auszugehen, das bei der Ablaufplanung entsprechend beachtet werden muss. Des Weiteren bedingt die dynamische Umgebung die regelmäßige Adaption eines erstellten Ablaufplans und damit ein Ablaufsteuerungsverfahren, das in Situationen, in denen die Unsicherheit nicht durch den robusten Ablaufplan kompensiert werden kann, die notwendigen Änderungen vornimmt.

1.2 Gliederung der Arbeit

Die vorliegende Arbeit ist in sieben Kapitel gegliedert. In Kapitel 2 wird das System „Luftfrachtterminal“ beschrieben und die Schnittstellenfunktion der Luftfrachtterminals für den Umschlag von Fracht zwischen dem Land- und Luftverkehr herausgestellt. Wichtige Punkte sind hierbei die detaillierte Beschreibung der Luftfrachttransportkette und des Luftfrachtterminals als zentralem Bindeglied sowie die innerhalb des Terminals durchgeführten Prozesse. Anschließend werden die spezifischen Herausforderungen, die im Kontext der Planung und Steuerung der Abfertigungsprozesse in einem Luftfrachtterminal bestehen, erläutert sowie drei Problemfelder identifiziert und diskutiert, die es für die Realisierung eines Ablaufplanungs- und -steuerungssystems zu adressieren gilt. Abschließend werden für jedes Problemfeld die Anforderungen an eine Problemlösung definiert.

Anschließend wird in Kapitel 3 der relevante Stand der Forschung im Hinblick auf geeignete Verfahren zur Lösung der diskutierten Problemfelder untersucht. Die Verfahren werden anhand ihrer Eignung zur Lösung der gestellten Anforderungen und ihrer Verwendung in einer Problemlösung je Problemfeld analysiert.

Darauf aufbauend wird in Kapitel 4 eine Präzisierung der zu leistenden Arbeit vorgenommen. Hierfür werden Arbeitspakete definiert, die existierende Ansätze erweitern bzw. neue Konzepte entwickeln, um das dargestellte Problem zu lösen. Die Arbeitspakete werden dabei den identifizierten Problemfeldern zugeordnet.

In Kapitel 5 wird das Konzept zur Realisierung einer Ablaufplanung und -steuerung im Kontext von Luftfrachtterminals vorgestellt und anhand der dargestellten Problemfelder untergliedert. Zunächst werden die für die einzelnen Problemfelder zu entwickelnden Verfahren in Form von Modulen in eine Architektur eingeordnet und die sich ergebenden Interdependenzen erläutert. Anschließend erfolgt die detaillierte Beschreibung der Module zur Erstellung des benötigten Informationsstands. Hierauf aufbauend wird die Ablaufplanung formalisiert und in einer evolutionären Metaheuristik umgesetzt. Den Abschluss bildet die Konzeption der Ablaufsteuerung.

Das erarbeitete Konzept eines Ablaufplanungs- und -steuerungssystems wird anschließend in Kapitel 6 validiert. Hierzu werden reale und synthetische Testinstanzen herangezogen und die vorgestellten Verfahren in verschiedenen Experimenten anhand von Indikatoren hinsichtlich ihrer Effektivität und Effizienz untersucht.

Abschließend erfolgt in Kapitel 7 eine Zusammenfassung sowie kritische Würdigung der in der Arbeit entwickelten Konzepte und der erzielten Ergebnisse. Ein Ausblick liefert Ansätze für zukünftigen Forschungsbedarf.

2 Problemstellung

Der Untersuchungsgegenstand dieser Arbeit ist die Ablaufplanung und -steuerung von Frachtabfertigungsprozessen in Luftfrachtterminals unter Berücksichtigung der dynamischen und stochastischen Umgebung, in der die Umschlagprozesse ausgeführt werden. In Abschnitt 2.1 erfolgt zunächst eine detaillierte Beschreibung des Systems „Luftfrachtterminal“ und die Definition wichtiger Begriffe, um die terminologische Basis für diese Arbeit zu bestimmen. Dabei werden die Prozessabläufe und verschiedenen Frachtflüsse innerhalb des Terminals näher betrachtet. Anschließend erfolgt in Abschnitt 2.2 die Beschreibung des Ablaufplanungs- und -steuerungsproblems sowie die Unterteilung in zu betrachtende Problemfelder. Die Anforderungen an Verfahren, die die identifizierten Problemfelder adressieren, werden in Abschnitt 2.3 definiert.

2.1 Systembeschreibung

Für ein grundlegendes Verständnis der im Rahmen der Ablaufplanung und -steuerung in Luftfrachtterminals bestehenden Herausforderungen, die es in dieser Arbeit zu adressieren gilt, bedarf es im Folgenden zunächst einer Einordnung des Luftfrachtterminals in die *Luftfrachttransportkette*.¹ Diese Terminals dienen als zentrales Bindeglied für den Umschlag von Fracht² zwischen dem Land- und Luftverkehr. Die Vor- und Nachläufe, die meist als bodengebundene Transporte erfolgen, sowie der Hauptlauf, der mit dem Flugzeug erfolgt, determinieren hierbei wichtige Faktoren, die es bei der Planung der Prozessabläufe innerhalb des Terminals zu berücksichtigen gilt.

2.1.1 Luftfrachttransportkette und involvierte Akteure

Ausgangspunkt des Frachttransports und damit der Beginn der Luftfrachttransportkette ist der *Versender*³ und das Ziel ist der *Empfänger*.⁴ Die involvierten Akteure und die Prozesse der Luftfrachttransportkette werden im Folgenden näher erläutert. Eine Übersicht zeigt Abbildung 2.1.

¹„Die Transportkette ist die Folge von technisch und organisatorisch miteinander verknüpften Vorgängen, bei denen Personen oder Güter von einer Quelle zu einem Ziel bewegt werden.“ [DIN 30781], S. 3.

²Unter Luftfracht wird in dieser Arbeit ausschließlich „die Fracht verstanden, die nach IATA-Beförderungsbestimmungen für Frachtgut abgefertigt und transportiert wird (IATA International Air Transport Association).“ [AIK⁺08], S. 757. Dies entspricht einer engeren Definition für Luftfracht (siehe [AIK⁺08], S. 757). Weiter gefasst können zu Luftfracht „alle Güter, die auf Linien- oder Charterflügen als Fracht, Express oder Post transportiert werden“ [AIK⁺08], S. 757, gezählt werden.

³International üblich ist der Begriff *Shipper* oder *Consignor*.

⁴Auch als *Consignee* bezeichnet.

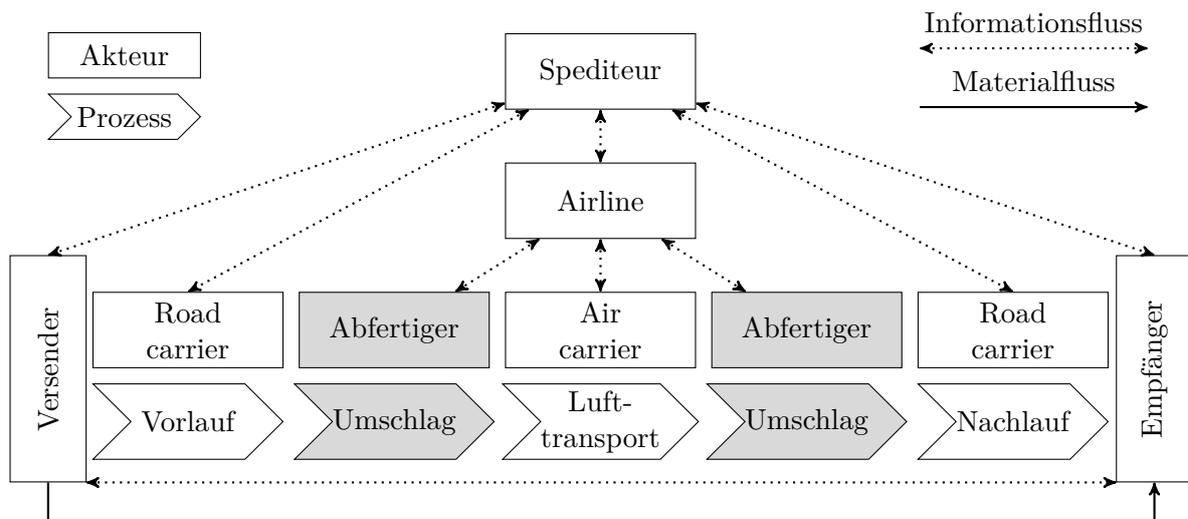


Abbildung 2.1: Luftfrachttransportkette und involvierte Akteure (in Anlehnung an: [Fry03], S. 12, und [Fry13], S. 230)

Versender ist das Unternehmen oder die Person, die für den Versand von Waren verantwortlich ist. Zwischen Versender und Empfänger besteht in der Regel ein Vertrag, der unter anderem die Lieferbedingungen sowie die Verantwortlichkeiten der beteiligten Parteien für den Transport regelt.⁵ Die für den Transport verantwortliche Partei, typischerweise⁶ der Verkäufer der Waren (z. B. produzierende Industrie, Handel etc.), beauftragt in der Regel⁷ einen *Luftfrachtspediteur*⁸ mit der Besorgung der Versendung⁹ der Waren an den Empfänger, typischerweise den Käufer. Die Vertragsbeziehungen zwischen Versender und Spediteur sind dabei im Speditionsvertrag¹⁰ geregelt.

Der Luftfrachtspediteur¹¹ ist für die Organisation der Beförderung der Fracht vom Versender zum Empfänger zuständig. Hierfür organisiert und koordiniert der Spediteur in der Regel sowohl die bodengebundenen Transporte vom Versender zum Abflughafen und vom Zielflughafen zum Empfänger als auch den Lufttransport durch eine Fluggesellschaft. Greift der Spediteur für den physischen Transport ausschließlich auf fremde Dienstleistungen zurück, handelt es sich bei dem Spediteur um einen *Fourth-Party-Logistics-Provider* (4PL).¹² In diesem Fall wird der Spediteur sowohl mit

⁵Siehe [Fry13], S. 230.

⁶Der Empfänger kann z. B. auch die Partei sein, die als Auftraggeber für die Spedition auftritt.

⁷Der Fall, in dem der Versender seine Fracht direkt zum Abflughafen oder zur Airline liefert, ist die Ausnahme (vgl. [GRR07], S. 128; [Pet07], S. 13).

⁸„Der Luftfrachtspediteur übernimmt es, gewerbsmäßig Güterversendungen durch Luftverkehrsgesellschaften für [sic] Rechnung des Versenders oder Empfängers im eigenen Namen zu organisieren.“ [KKK12], S. 414.

⁹Siehe § 454 HGB.

¹⁰Siehe § 453 HGB.

¹¹International üblich ist der Begriff *Forwarder*.

¹²4PL-Logistikdienstleister besitzen in der Regel keine eigenen Assets für den Frachttransport (vgl. [PKM11], S. 213).

Lkw-Transportunternehmen als auch der Fluggesellschaft Frachtverträge¹³ abschließen, die den physischen Transport der Waren übernehmen. Die meisten Spediteure übernehmen jedoch den bodengebunden Transport selbst und greifen lediglich für den Lufttransport auf die Fluggesellschaften zurück.¹⁴

Die Sendungen sind mit einem *Luftfrachtbrief*¹⁵ (engl. *Air Waybill* (AWB)) zu versehen. Hierbei sind verschiedene Arten von Luftfrachtbriefen zu unterscheiden:¹⁶ Sendungen mit einem Absender und einem Empfänger, die direkt zum Ziel befördert werden, erhalten eine Direkt-AWB. Speditionsintern werden die Sendungen jedoch häufig zunächst in eigenen Lagern gesammelt und einzelne Sendungen zu größeren Einheiten (Sammelgutsendungen oder Consol) zusammengefasst, bevor sie zum Abflughafen gebracht werden. Die einzelnen Luftfrachtsendungen der Versender erhalten jeweils einen eigenen Haus-Luftfrachtbrief, die sogenannte *House Air Waybill* (HAWB),¹⁷ der den Frachtvertrag zwischen Versender und Spediteur dokumentiert. Zur Ausgabe der HAWB sind ausschließlich als IATA-Agenten zertifizierte Spediteure berechtigt.¹⁸ Eine konsolidierte Sendung wird anschließend mit einer *Master Air Waybill* (MAWB) versehen, unter der die HAWBs zusammengefasst werden. Die MAWB dokumentiert den Frachtvertrag zwischen Spediteur und Fluggesellschaft.¹⁹

Der Spediteur übernimmt zudem die notwendigen Aufgaben für die versandfertige Bereitstellung der Sendung (engl. *ready for carriage*) für den Lufttransport. Hierzu gehört die korrekte Verpackung und Beschriftung, die Zollabfertigung für den Export sowie die Bereitstellung von Begleitdokumenten und Einhaltung von gesetzlichen Vorschriften für Sonderfracht (Gefahrgut, verderbliche Ware, Wertfracht und lebende Tiere).²⁰ Zur Deklaration werden sogenannte *Special Handling Codes* (SHCs) verwendet.

Am Abflughafen übergibt der Spediteur die Sendung inklusive Luftfrachtbrief und evtl. weiterer Dokumente an den *Frachtabfertiger*,²¹ der im Auftrag der Fluggesellschaft für die Flugvorbereitung der Fracht zuständig ist.²² Abhängig von der Art des Frachtgutes, dem Ziel, der Priorität und weiteren Faktoren müssen die Sendungen durch den Spediteur bis zu einer bestimmten Zeit vor dem Abflug (engl. *latest acceptance time* oder *time before departure*) am Abflughafen eintreffen.²³ Früher eintreffende Sendungen werden in den Luftfrachtterminals temporär zwischengelagert und später für die Anforderungen

¹³Siehe § 407 HGB.

¹⁴Vgl. [Pet07], S. 18.

¹⁵„Urkunde über den Abschluss und Inhalt des Frachtvertrages im Luftverkehr. Das Dokument dient als Empfangsbestätigung durch den Carrier (Luftfrachtführer), als Warenbegleit- und Sperrpapier, als Auslieferungsquittung für den Empfänger und ist damit Grundlage für die Verzollung und die Erstellung von Versicherungspolicen.“ [KKK12], S. 9.

¹⁶Vgl. ebd., S. 413.

¹⁷Siehe [RCB10], S. 372.

¹⁸Siehe ebd., S. 368. Für eine ausführliche Erläuterung, insbesondere der Voraussetzungen für eine Zertifizierung, vgl. [GRR07], S. 138–141.

¹⁹Siehe [RCB10], S. 372.

²⁰Vgl. [GRR07], S. 129.

²¹International üblich ist der Begriff *Ground Handler*.

²²Siehe [Fry13], S. 230, 243.

²³Vgl. [GRR07], S. 129.

der Beförderung im Flugzeug durch den Frachtabfertiger vorbereitet.²⁴ Die fertigen Luftfrachtcontainer oder -paletten werden anschließend durch *Bodenverkehrsdienstleister* (präziser: Vorfeldabfertiger) vom Luftfrachtterminal zum Vorfeld transportiert. Von hier aus beginnt zu einem bestimmten Zeitpunkt vor Abflug die Beladung des Flugzeugs anhand des Ladeplans (engl. *Loadsheet*) durch die *Rampabfertiger*.²⁵ Des Weiteren werden neben der Übergabe der Fracht die Ladeliste (engl. *Cargo Manifest*) und die Hinweise für den Piloten (engl. *Notification to Captain* (NOTOC)) erstellt. Die Ladeliste enthält dabei Informationen über die verladene Fracht (Anzahl, Gewicht, Luftfrachtbriefnummern etc.) und die NOTOC informiert über mögliche Gefahren durch Sonderfracht, wie z. B. Gefahrgut.²⁶

Neben der Anlieferung von Sendungen durch den Spediteur erfolgt zusätzlich die Anlieferung von Luftfrachtsendungen mit Luftfrachtbrief von anderen Flughäfen als sogenannter *Luftfrachtersatzverkehr* (engl. *Road Feeder Service* (RFS)). Dieser überwiegend mit speziellen Lkws ausgeführte Transport von Flugzeugladeeinheiten erfolgt im Auftrag der Fluggesellschaften.²⁷

Für den eigentlichen Lufttransport vom Ab- zum Zielflughafen bzw. über Zwischenflughäfen sind die *Fluggesellschaften* (engl. *Airline*) verantwortlich. Diese lassen sich in drei Hauptklassen unterteilen:²⁸ *Reine Frachtfluggesellschaften*, die spezielle Frachtflugzeuge oder umgebaute Passagiermaschinen verwenden und ausschließlich Fracht transportieren. Diese machen ca. 10 % bis 15 % am weltweiten Frachtaufkommen aus.²⁹ *Kombinierte Fluggesellschaften* nutzen hauptsächlich Passagiermaschinen und transportieren zusätzlich Fracht in Unterflurladeräumen. Je nach Fluggesellschaft (insbesondere europäische Fluggesellschaften) nimmt der zusätzliche Frachttransport einen wichtigen Faktor im Geschäftsmodell ein. Mehr als 50 % des weltweiten Frachtaufkommens werden als Beiladefracht transportiert.³⁰ Als dritte Klasse sind die *Integratoren* zu nennen. Im Kontrast zu den anderen beiden Typen, die ihre Dienstleistung den Luftfrachtspediteuren anbieten, stellen Integratoren ihre Dienstleistung direkt dem Versender zur Verfügung. Hierfür besitzen diese Unternehmen³¹ eigene Fuhrparks und Flugzeuge und bieten die gesamten Leistungen der Luftfrachttransportkette aus einer Hand an.³²

Am Zielflughafen wird die Transportkette in umgekehrter Reihenfolge durchlaufen. Die Fracht wird vom Flugzeug zum Luftfrachtterminal gebracht, wo sie und die zugehörigen Dokumente überprüft werden. Der Spediteur erhält eine Vorbenachrichtigung über die Ankunft der Sendung.³³ Fracht in Luftfrachtcontainern oder -paletten wird vereinzelt und bis zur Abholung durch den Spediteur zwischengelagert. Der Spediteur bereitet – falls nötig – die Einfuhrdokumente vor und kümmert sich um die Zollabfertigung für den

²⁴Vgl. Abschnitt 2.1.2.

²⁵Vgl. [GRR07], S. 129, 143.

²⁶Vgl. ebd., S. 154.

²⁷Vgl. ebd., S. 86 ff.

²⁸Siehe [Pet07], S. 22; [PKM11], S. 214–218.

²⁹Siehe [PKM11], S. 214.

³⁰Siehe [PKM11], S. 215; [Men13], S. 22.

³¹FedEx, UPS, TNT und DHL sind die vier größten Integratoren weltweit.

³²Siehe [PKM11], S. 216.

³³Vgl. [Pet07], S. 17.

Import. Die Sendung wird anschließend vom Spediteur am Luftfrachtterminal abgeholt, falls nötig im eigenen Lager erneut vereinzelt sowie zwischengelagert und schließlich zum Empfänger befördert.

2.1.2 Prozessabläufe im Luftfrachtterminal

Bei der im vorangegangenen Abschnitt beschriebenen Luftfrachttransportkette handelt es sich um eine mehrgliedrige Transportkette, bei der neben dem Wechsel des Verkehrsmittels und -trägers³⁴ häufig auch das Transportbehältnis gewechselt wird und es sich somit um gebrochenen oder multimodalen Verkehr handelt.³⁵

Diese Wechsel erfolgen innerhalb der Luftfrachtterminals, die als Umschlagsysteme von Jünemann & Schmidt technischen Systemen zugeordnet werden, „die einen Transformationsprozeß hinsichtlich Ort, Zeit, Menge, Zusammensetzung und Qualität durchführen.“³⁶ Das Luftfrachtterminal nimmt somit innerhalb der Transportkette eine zentrale Stellung an der Schnittstelle zwischen Straßen- und Luftverkehrsmitteln ein, in dem der *Umschlag*³⁷ zwischen diesen Verkehrsmitteln und die Abfertigung der Fracht realisiert werden. Nach Stein dient der Umschlag dem Ziel, „die Auslastung der Transportfahrzeuge zu erhöhen und damit die Transportkosten im Gesamtsystem zu senken.“³⁸ Dies wird innerhalb des Luftfrachtterminals zum einen durch die Bildung von *Ladeeinheiten*³⁹ und zum anderen durch die temporäre Lagerung der Fracht in geeigneten Lagern erreicht.⁴⁰

Die Ladeeinheitenbildung richtet sich nach den spezifischen Anforderungen der eingesetzten Flugzeugtypen und den durch die Laderäume vorgegebenen Restriktionen.⁴¹ Durch eine Transformation der Fracht hinsichtlich Menge und Zusammensetzung wird das Ziel verfolgt, die Laderäume hinsichtlich Volumen und Gewicht möglichst optimal auszulasten. Des Weiteren ermöglichen es die Ladeeinheiten, die hohe Anzahl kleiner Stückgüter zu wenigen größeren Einheiten zusammenzufassen und somit die Be- und Entladevorgänge des Flugzeugs sowie die Transportprozesse effizienter auszuführen. Für den Aufbau der Ladeeinheiten, die häufig als *Unit Load Device* (ULD) bezeichnet werden, kommen als *Ladehilfsmittel*⁴² (hier synonym: Ladungsträger) Luftfrachtcontainer und -paletten zum Einsatz, die abhängig von ihrer Form und Größe in verschiedenen Flugzeugtypen eingesetzt werden können. Zur Charakterisierung der Ladehilfsmittel wird von

³⁴Es erfolgt ein Wechsel des Verkehrsmittels vom Straßenfahrzeug zum Flugzeug und des Verkehrsträgers von der Straße in die Luft.

³⁵Vgl. [JS00], S. 327.

³⁶Ebd., S. 3.

³⁷„Der Umschlag ist die Gesamtheit der Förder- und Lagervorgänge beim Übergang der Güter auf ein Transportmittel, beim Abgang der Güter von einem Transportmittel und wenn Güter das Transportmittel wechseln.“ [DIN 30781], S. 3.

³⁸[Ste12], S. 602.

³⁹„Als Ladeinheit bezeichnet man Güter, die zum Zwecke des Umschlags durch einen Ladungsträger zusammengefaßt sind.“ [DIN 30781], S. 2.

⁴⁰Vgl. [Fry11], S. 27 f.

⁴¹Vgl. [BG12], S. 115; [Fry11], S. 15 f.

⁴²Ladehilfsmittel „dienen der Aufnahme ungleichförmiger Artikel, um daraus gleichartige Ladeeinheiten zwecks Mengenbündelung zu bilden.“ [KKK12], S. 297.

der IATA eine Typbezeichnung in Form einer zehnstelligen ULD-Nummer⁴³ verwendet, aus der sich die Verwendbarkeit⁴⁴ der ULD für die verschiedenen Flugzeugtypen und Einsatzbereiche ergibt.⁴⁵

Der Transformationsprozess hinsichtlich der Zeit erfolgt durch das Lagern der Fracht. Das Lagern unterstützt die Ladeeinheitenbildung, indem die innerhalb eines bestimmten Zeitfensters vor Abflug von Straßenfahrzeugen angelieferte Fracht solange zwischengelagert wird, bis der entsprechende Zeitpunkt für den Aufbau der Ladeeinheit gekommen ist. Dies gilt in gleicher Weise für bereits abgebaute und somit vereinzelt Fracht, die ebenfalls bis zur Abholung durch den Spediteur zwischengelagert wird. Die hohen Bereitschaftskosten des Flugzeugs determinieren zudem „die zeitlichen Anforderungen für die Luftfrachtanfertigung am Boden.“⁴⁶ Durch das Lager kann die Ladeeinheitenbildung über einen längeren Zeitraum vor dem Abflug verteilt und bereits erstellte Ladeeinheiten können zwischengelagert werden. Die Lagerung dient dazu, die Auslastung der Flugzeuge zu optimieren, da diese nicht auf die Fertigstellung der Ladeeinheitenbildung warten können.

Das Luftfrachtterminal, in dem die zuvor beschriebenen Transformationsprozesse durchgeführt werden, untergliedert sich nach Arnold et al.⁴⁷ in:

- „die Landseite [...] [mit] Flächen zur Anlieferung und Abholung [von Fracht durch Straßenfahrzeuge] [...],
- die Luft- bzw. Vorfeldseite [...] [mit] Flächen für die Bereitstellung und den Transport zur Be- und Entladung der Flugzeuge [...]
- [sowie] das eigentliche Luftfrachtgebäude.“

Innerhalb des Luftfrachtterminals lassen sich grob die Bereiche der für die Lagerung der Fracht verwendeten Lager und der Abfertigungsbereiche mit Arbeitsstationen für den Auf- bzw. Abbau von Ladeeinheiten unterscheiden.⁴⁸

Die für die Lagerung der Fracht eingesetzten Lager können grundsätzlich in Sendungslager und ULD-Lager unterschieden werden. In den Sendungslagern wird das lose Stückgut z. B. in Gitterboxen und in den ULD-Lagern werden vollständige Flugzeugladeeinheiten gelagert.⁴⁹ Des Weiteren gibt es spezielle Lager für die Aufbewahrung

⁴³Die Nummer besteht aus einem dreistelligen Präfix zur Identifizierung der Art des Ladehilfsmittels, der Grundfläche sowie der Kontur und Kompatibilität, gefolgt von einer fünfstelligen Seriennummer und einer zweistelligen Buchstabenkombination für den Eigentüternachweis (siehe [BG12], S. 119 f.). Bei AKE12345 LH handelt es sich beispielsweise „um einen zertifizierten Container (A) mit der Grundfläche 60,4 × 61,5 Zoll (K) und halber Breite sowie abgeschrägter Kontur zum Einsatz auf dem Lower Deck (E), der die laufende Nummer 12345 trägt und im Eigentum der Lufthansa (LH) steht.“ [BG12], S. 120.

⁴⁴Siehe IATA ULD Regulations (ULDR), vormals IATA ULD Technical Manual (UTM), für eine vollständige Auflistung [IATA2014].

⁴⁵Vgl. [BG12], S. 118 f.

⁴⁶[Fry11], S. 18.

⁴⁷Siehe [AIK⁺08], S. 770.

⁴⁸Weitere Bereiche, wie z. B. Büroräume, gehören häufig ebenfalls zum Terminal, sind aber nicht Gegenstand der Betrachtung in dieser Arbeit.

⁴⁹Siehe [AIK⁺08], S. 772.

von Sonderfracht, z. B. temperaturgeführte Räume für verderbliche Waren.⁵⁰ Je nach Automatisierungsgrad⁵¹ des Luftfrachtterminals können die Lager sowohl manuell bedient als auch vollautomatisiert sein.⁵²

Angebunden an die Lager sind die Arbeitsstationen in den Abfertigungsbereichen. Hier erfolgt der Aufbau (engl. *Build-up* (BU)), bei dem einzelne Sendungen aus dem Sendungslager zu Luftfrachtcontainern und -paletten konsolidiert werden sowie der Abbau (engl. *Break-down* (BD)), bei dem die Flugzeugladeeinheiten dekonsolidiert und die einzelnen Sendungen auf die Lager verteilt werden.⁵³ Die an den Arbeitsstationen verrichtete Arbeit erfolgt zu einem hohen Grad als manuelle Tätigkeit unter Einsatz von Arbeitspersonal, die je nach Gewicht der zu bearbeitenden Sendungen durch Gabelstapler unterstützt wird.

Die innerhalb des Luftfrachtterminals ausgeführten Umschlagprozesse und die sich hieraus ergebenden Frachtflüsse sind in Abbildung 2.2 dargestellt. Nach Frye⁵⁴ können die folgenden Umschlagprozesse unterschieden werden:

- Export: Annahme von loser bzw. palettierter Fracht auf der Landseite des Terminals. Bei Bedarf Aufbau der Ladeeinheiten für den Flug. Ausgang über die Luftseite des Terminals zum Flugzeug.
- Import: Eingang von Ladeeinheiten auf der Luftseite des Terminals. Bei Bedarf Abbau der Ladeeinheiten und Bereitstellung der Fracht auf der Landseite des Terminals.
- Transfer: Eingang von Ladeeinheiten auf der Luftseite des Terminals. Erstellung zielreiner Ladeeinheiten und Ausgang über die Luftseite des Terminals zum Flugzeug.

Der für die Umschlagprozesse nötige Aufwand bei der Frachtabfertigung richtet sich danach, ob die Fracht als Ladeeinheit oder als lose⁵⁵ Fracht, d. h. einzelne Sendungen und Frachtstücke, umgeschlagen wird.

Fracht für den Export wird auf der Landseite des Terminals durch Straßenfahrzeuge angeliefert. Vollständige Ladeeinheiten können je nach Zeitpunkt ihres Eintreffens entweder direkt zur Luftseite transportiert oder zum zeitlichen Ausgleich zunächst im ULD-Lager zwischengelagert werden. Lose Fracht, die zu einer Ladeeinheit aufgebaut werden soll, muss hingegen, nach eventueller Zwischenlagerung im Sendungslager, zu einer der Arbeitsstationen transportiert und dort zunächst zu einer Ladeeinheit konsolidiert werden, bevor diese dann mit eventuell erneuter Zwischenlagerung im ULD-Lager zum Flugzeug transportiert wird. Lose Fracht wird bei Bedarf im Sendungslager zwischengelagert und zum entsprechenden Zeitpunkt mit Hilfe von Frachtwagen zum Flugzeug transportiert.⁵⁶

⁵⁰Siehe [AIK⁺08], S. 774.

⁵¹In dieser Arbeit wird von vollautomatisierten Lagern ausgegangen, die zudem fördertechnisch direkt an die entsprechenden Arbeitsstationen für die manuelle Bearbeitung der Fracht angebunden sind.

⁵²Vgl. [JS00], S. 302; [AMW11], S. 460 ff.

⁵³Vgl. [BG12], S. 102.

⁵⁴Siehe [Fry11], S. 42.

⁵⁵International üblich ist hier auch der Begriff *loose*.

⁵⁶Vgl. [RG09], S. 728.

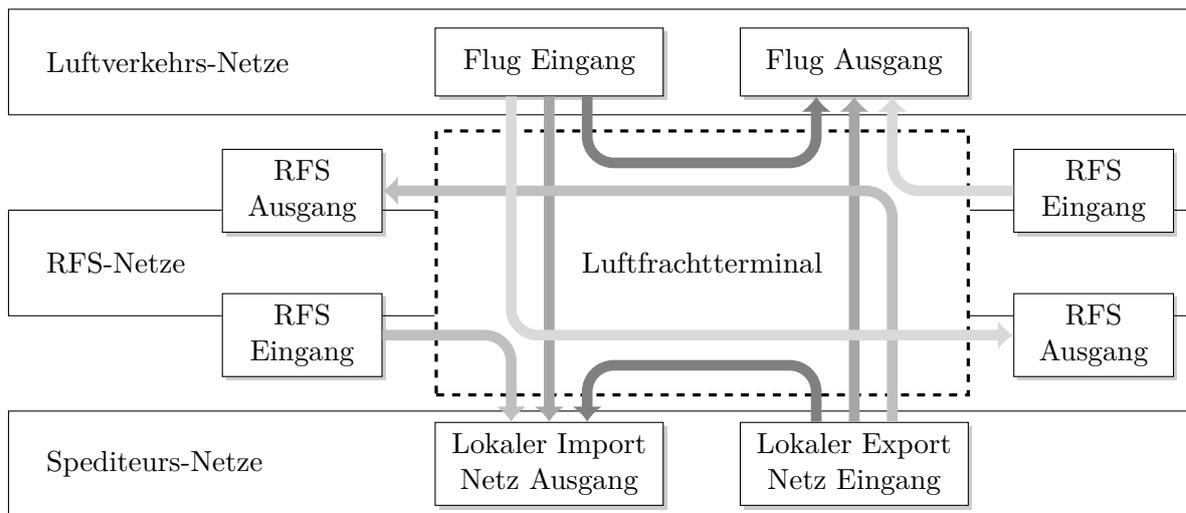


Abbildung 2.2: Frachtflüsse im Luftfrachtterminal (in Anlehnung an: [Fry03], S. 13)

Importfracht, die das Terminal auf der Luftseite erreicht, kann ebenfalls als Ladeeinheit oder als lose Fracht angeliefert werden. Die lose Fracht kann je nach Zeitpunkt ihres Eintreffens entweder zu den Sendungslagern oder direkt zur Landseite transportiert werden. Die ankommenden Ladeeinheiten, die abgebaut werden sollen, werden nach eventuell erfolgter Zwischenlagerung im ULD-Lager an den Arbeitsstationen dekonsolidiert und die Fracht wird zu den Sendungslagern transportiert. Fracht, die ihren Zielflughafen erreicht hat, wird entweder als ULD oder lose Fracht auf der Landseite des Terminals durch die Spediteure abgeholt.⁵⁷

Transferfracht kann direkt auf den anschließenden Flug umgeschlagen werden, wenn die in den Ladeeinheiten enthaltene Fracht auf die gleiche Route gebucht ist und die Ladeeinheit durch den Anschlussflug transportiert werden darf. Andernfalls muss die Ladeeinheit zunächst dekonsolidiert und anschließend mit anderer Fracht konsolidiert werden.⁵⁸

Unterstützt werden die physischen und dokumentarischen Frachtabfertigungsprozesse durch lokale Informationssysteme bzw. Frachtabfertigungssysteme. Arnold et al.⁵⁹ beschreiben die wichtigsten Funktionen dieser Systeme wie folgt:

- „Datenerfassung und -abgleich bei Anlieferung,
- Datenabgleich (Check-in) bei der Eingangsabfertigung,
- Lagerverwaltung,
- Ladevorbereitung und Manifestierung (Erstellung Ladeliste) bei der Ausgangsabfertigung,
- Zollabwicklung bei Auslieferung.“

⁵⁷Vgl. [RG09], S. 728.

⁵⁸Vgl. ebd.

⁵⁹Siehe [AIK⁺08], S. 776.

Die manuelle Frachtbearbeitung an den Arbeitsstationen wird durch mobile Datenerfassung und -übertragung durch Handgeräte unterstützt. Dies ermöglicht eine Überwachung des Bearbeitungsfortschritts in Echtzeit.

2.2 Problemfelder der Ablaufplanung und -steuerung

Die im vorangegangenen Abschnitt beschriebenen Prozesse zur Abfertigung der Fracht bedürfen einer genauen Planung und Steuerung, die an den für die einzelnen Sendungen zugesicherten Lieferterminen sowie den durch den Flugplan vorgegebenen Zeiten auszurichten sind. Des Weiteren sind bei dieser Planung die betrieblichen Kapazitäten des Luftfrachtterminals zu berücksichtigen.

Die betrieblichen Kapazitäten (Kapazitätsangebot) des Luftfrachtterminals sind definiert durch das zur Verfügung stehende Personal, die Arbeitsstationen mit ihren Pufferflächen sowie die Lager- und Förderkapazitäten.⁶⁰ Die Belastung (Kapazitätsnachfrage) ist durch die zu bearbeitenden Sendungen sowie die Strukturierung dieser Nachfrage über die Zeit gegeben. Der Vorgang der Abstimmung zwischen Kapazitätsangebot und -nachfrage wird als Kapazitätsabstimmung bezeichnet.⁶¹ Das Kapazitätsangebot kann einerseits an die Kapazitätsnachfrage angepasst werden, dann liegt eine Kapazitätsanpassung vor. Im anderen Fall liegt ein Kapazitätsabgleich vor, bei dem die Kapazitätsnachfrage an das -angebot angepasst wird. Eine Kapazitätsanpassung ist kurzfristig (auf Tagesbasis) nur begrenzt möglich. Bis auf das Personal ist das Kapazitätsangebot mit der Fertigstellung des Terminals festgelegt. Die Personalkapazität wird für gewöhnlich auf Wochenbasis geplant⁶² und kann sowohl Schichten für Vollzeit- als auch Teilzeitbeschäftigte beinhalten. Eine kurzfristige Anpassung (auf Tages- oder Stundenbasis) der Personalkapazität ist nur durch flexible Arbeitszeitregelungen (z. B. Abrufarbeit) durchführbar. Ein Kapazitätsabgleich ist ebenfalls nur begrenzt möglich, da die Menge der zu bearbeiten Sendungen sowie deren Ankunfts- und Abflugzeiten durch die Buchung zu bestimmten Flügen bereits fest definiert wurde und somit das Belastungsprofil für das Luftfrachtterminal vorgibt. Die Kapazitätsnachfrage kann jedoch unter Berücksichtigung der Sendungs- und Flugvorgaben zeitlich strukturiert und damit dem Kapazitätsangebot angeglichen werden, indem eine Ablaufplanung durchgeführt wird.

Mit der Ablaufplanung (engl. *Scheduling*) wird die zeitliche und mengenmäßige Allokation limitierter Ressourcen zu einer Menge von Aufgaben vorgenommen. Ziel hierbei ist es, eine oder mehrere Zielfunktionen unter Berücksichtigung von verschiedenen Nebenbedingungen bzw. Restriktionen zu optimieren.⁶³ Ergebnis der Ablaufplanung ist ein Ablaufplan (engl. *Schedule*), der für jede Aufgabe angibt, wann diese unter Verwendung welcher Ressourcen bearbeitet werden soll.

Übertragen auf das Luftfrachtterminal entsprechen die Arbeitsstationen und das Personal den Ressourcen und die durchzuführenden Auf- und Abbau-Tätigkeiten den

⁶⁰Vgl. [Fry11], S. 50.

⁶¹Vgl. [Bul86], S. 38.

⁶²Vgl. [RG09].

⁶³Vgl. [JP14], S. 3 f.; [Pin12], S. 1; [BEP⁺07], S. 1.

Aufgaben, die im Folgenden als Frachtabfertigungsaufträge oder kurz Aufträge (engl. *Jobs*) bezeichnet werden. Die Aufgabe der Ablaufplanung besteht in der Zuordnung einer Arbeitsstation zu einem Auftrag sowie der Festlegung der zeitlichen Reihenfolge, mit der die Aufträge bearbeitet werden, wenn einer Ressource mehr als ein Auftrag zugewiesen wurde. Für jeden Auftrag ist durch vorausgegangene Planungen der späteste Zeitpunkt, zu dem der Umschlag der Fracht, die zu diesem Auftrag gehört, abgeschlossen sein muss, bereits festgelegt.⁶⁴ Dieser Zeitpunkt wird im Folgenden als Fälligkeitstermin bezeichnet. Für Exportfracht ist dies der geplante Abflugzeitpunkt des Flugzeugs, auf den die Fracht gebucht ist, abzüglich der Zeiten, die für die verschiedenen Vorfeldprozesse benötigt werden. Für Importfracht ist dies ein mit dem Spediteur vertraglich festgelegter Zeitpunkt, der in Stunden nach der Ankunft des Flugzeugs definiert wird. Der Zeitraum, in dem ein Auftrag bearbeitet werden muss, ergibt sich somit aus dem Zeitpunkt, ab dem die Fracht physisch in dem Terminal vorhanden ist und dem Zeitpunkt des Fälligkeitstermins. Ziel der Ablaufplanung ist es, den Anteil der Aufträge, die ihren Fälligkeitstermin einhalten, unter Berücksichtigung der betrieblichen Kapazitäten und der Zeiträume, in denen die Fracht bearbeitet werden muss, zu maximieren.

Neben der Zuordnung von Arbeitsstationen zu Aufträgen ist als weitere Ressource die für einen Auftrag notwendige Personalstärke zu berücksichtigen. Der Aufbau kleiner Ladeeinheiten benötigt bspw. weniger Personal als der großer Ladeeinheiten. Des Weiteren kann von einer Qualifikationshierarchie des Personals bezüglich der Auf- und Abbautätigkeiten ausgegangen werden.⁶⁵ Für den Aufbau einer Ladeeinheit sind bspw. umfangreichere Kenntnisse notwendig als für den Abbau. Die Ablaufplanung muss daher bei der Erstellung der Ablaufpläne sowohl die benötigte Personalstärke für den Auf- bzw. Abbau einer Ladeeinheit als auch die Qualifikation des Personals berücksichtigen.

Die durch die Ablaufplanung generierten Ablaufpläne induzieren eine gewisse Belastung der Lager und der innerbetrieblichen Transportsysteme, die Lager und Arbeitsstationen miteinander verbinden. Die maximalen Förderkapazitäten für den durch den Ablaufplan abgedeckten Zeitraum sind daher bei der Ablaufplanung zu berücksichtigen. Der Plan ist ggf. so anzupassen, dass einige Bereiche des Lagers und damit des Transportsystems weniger und andere stärker belastet werden, um somit einen Ausgleich zu schaffen.

Der Aufwand für die Auf- und Abbautätigkeiten an den Ladeeinheiten variiert mit der Struktur der umzuschlagenden Objekte. Aufgrund der Heterogenität⁶⁶ der zu bearbeitenden Fracht ist eine Abschätzung der benötigten Dauer für die Abfertigung eines Auftrags nur auf Basis von Erfahrung und Schätzwerten möglich. Schwere Sendungen, die mit einem Gabelstapler bearbeitet werden müssen, benötigen bspw. mehr Zeit als Sendungen, die ohne Hilfsmittel bearbeitet werden können. Des Weiteren erfolgen diese Umschlagstätigkeiten als manuelle Tätigkeiten,⁶⁷ die je nach eingesetztem Personal und schwankenden Leistungsgraden ebenfalls zeitlichen Störungen unterliegen. Diese Tatsache erschwert die Ablaufplanung, da die Bearbeitungszeiten der verschiedenen Aufträge als unsicher anzunehmen sind. Eine Ablaufplanung, die auf Basis von Schätzwerten bezüglich

⁶⁴Siehe [Fry11], S. 49 f.

⁶⁵Vgl. [RG09], S. 727.

⁶⁶Vgl. [Fri10], S. 12.

⁶⁷Vgl. Abschnitt 2.1.2.

der Bearbeitungszeiten beruht, erzeugt somit unter Umständen einen Ablaufplan, dessen geplanter Zielfunktionswert stark von dem tatsächlich realisierten Zielfunktionswert abweicht. Informationen über diese Bearbeitungszeitunsicherheiten sollten daher bereits bei der initialen Ablaufplanung berücksichtigt werden.

Als Informationsquelle können z. B. historische Betriebsdaten über vergangene Aufträge herangezogen werden. In diesen ist gespeichert, wie lang die Bearbeitungszeiten der Aufträge mit bestimmten Auftragseigenschaften waren. Auf Basis der häufigen Ausführung gleicher oder sehr ähnlicher Aufträge sind somit umfangreiche Informationen über die Bearbeitungszeitunsicherheiten vorhanden, die es in geeigneter Weise in einer für die Ablaufplanung verwendbaren Form aufzubereiten gilt.

Der erstellte Ablaufplan wird schließlich ausgeführt und die eingeplanten Aufträge werden sukzessiv an den eingeplanten Arbeitsstationen abgearbeitet. Hierbei ist der Fortschritt laufend zu überwachen und auf Basis eines Soll-/Ist-Vergleichs der geplanten und realisierten Zeiten sind Abweichungen zu identifizieren. Führen diese Abweichungen zu Auswirkungen, die durch die initiale Berücksichtigung der Informationen über Unsicherheiten nicht kompensiert werden können, ist der Ablaufplan entsprechend anzupassen.

Zusammenfassend sind die folgenden drei Problemfelder bei der Planung und Steuerung der Abfertigungsprozesse in einem Luftfrachtterminal zu adressieren, welche in den Abschnitten 2.2.1 bis 2.2.3 detailliert beschrieben werden:

- Datenanalyse der Abfertigungsprozesse
- Multikriterielle Ablaufplanung der Abfertigungsprozesse
- Ablaufsteuerung und Revision von Ablaufplänen

2.2.1 Datenanalyse der Abfertigungsprozesse

Für die Ablaufplanung ist ein Informationsstand für die neu einzuplanenden Aufträge nötig. Insbesondere Informationen über die Bearbeitungszeiten dieser Aufträge sind wichtig, um eine zuverlässige Planung der Anfangs- und Endzeiten der Abfertigungsaufträge durchführen zu können. Wie im vorherigen Abschnitt beschrieben, sind die Bearbeitungszeiten als unsicher anzunehmen und Informationen, die eine Quantifizierung dieser unsicheren Bearbeitungszeiten ermöglichen, lassen sich aus historischen Betriebsdaten extrahieren. Im Folgenden wird daher zunächst die vorhandene Datenbasis beschrieben und anschließend aufgezeigt, welcher Informationsstand für die Ablaufplanung benötigt wird.

2.2.1.1 Vorhandene Datenbasis

Daten über einzuplanende sowie bereits bearbeitete Frachtabfertigungsaufträge werden typischerweise in einer Datenbank vorgehalten. Hier werden pro Zeile n verschiedene Aufträge und in den Spalten d Eigenschaften über diese Aufträge gespeichert. Ein Beispiel

Tabelle 2.1: Datenbasis über Frachtabfertigungsaufträge

Auftrag	Eigenschaft 1	...	Eigenschaft d
1	$e_{1,1}$...	$e_{1,d}$
\vdots	\vdots	\ddots	\vdots
n	$e_{n,1}$...	$e_{n,d}$

über die grundlegende Struktur ist in Tabelle 2.1 aufgeführt.⁶⁸ Für einzuplanende Aufträge werden z. B. folgende Eigenschaften gespeichert:

- Schätzwert über die voraussichtliche Bearbeitungszeit
- die innerhalb des Auftrags zu bearbeitenden Sendungen
- das für den Auftrag zu verwendende Ladehilfsmittel, wenn es sich um einen Aufbauauftrag handelt
- Fertigstellungszeit, zu dem der Auftrag bearbeitet sein muss

Für bereits bearbeitete Aufträge (historische Aufträge) werden zusätzlich statistische Daten erhoben und gespeichert, z. B.:

- die tatsächliche Start- und Endzeit sowie die Bearbeitungszeit des Auftrags
- die Arbeitsstation, die den Auftrag ausgeführt hat
- die Mitarbeiter, die an dem Auftrag beteiligt waren

Für die Schätzung der voraussichtlichen Bearbeitungszeit eines neu einzuplanenden Abfertigungsauftrags wird jedoch selten auf die Datenbasis über vergangene Aufträge zurückgegriffen. Stattdessen wird vereinfachend angenommen, dass für bestimmte Arten von Sendungen bestimmte Bearbeitungszeiten nötig sind. Diese Zeiten können beispielsweise im Rahmen einer Zeitstudie ermittelt werden. Aus den gemessenen Ist-Zeiten wird bspw. durch Mittelwertbildung eine Soll-Zeit abgeleitet, die dann für zukünftige Aufträge Verwendung findet.

Als Beispiel sei ein Auftrag a gegeben, der zwei Arten von Sendungen enthält. Die Menge \mathcal{F}_i bezeichne hierbei die Sendungen der Art i . Die erste Art von Sendung kann ohne Hilfsmittel bearbeitet werden und es wurde eine Sollbearbeitungszeit von τ_1^{soll} festgelegt. Für die zweite Art wird aufgrund des Sendungsgewichts ein Gabelstapler benötigt und es wurde eine Sollbearbeitungszeit von τ_2^{soll} festgesetzt. Die geschätzte Bearbeitungszeit $\hat{\tau}_a$ des Auftrags lässt sich dann über Gleichung (2.1) ermitteln.

$$\hat{\tau}_a = \sum_{i=1}^2 |\mathcal{F}_i| \cdot \tau_i^{\text{soll}} \quad (2.1)$$

⁶⁸Jain & Dubes sprechen in diesem Zusammenhang von einer „ $n \times d$ pattern matrix“ [JD88], S. 8.

Dieses Vorgehen führt jedoch zu einigen Problemen. Zum einen ist die Zeitstudie unter realistischen Bedingungen durchzuführen und ggf. häufiger zu wiederholen, um einen aktuellen Bezug zu den realen Bedingungen bspw. der Frachtstruktur zu behalten. Zum anderen wird über Gleichung (2.1) nur eine auf Mittelwerten basierende Punktschätzung der benötigten Bearbeitungszeit vorgenommen. Es kann jedoch nicht davon ausgegangen werden, dass die Bearbeitungszeit für eine bestimmte Art von Sendung immer dem verwendeten Mittelwert entspricht. Dies kann in der Folge zu den in Abschnitt 2.2 beschriebenen Problemen mit Ablaufplänen, die auf Basis von Schätzwerten erstellt werden, führen.⁶⁹

Daher soll hier auf Grundlage der vorhandenen Datenbasis ein für die Ablaufplanung benötigter Informationsstand erarbeitet werden, der – statt einer Punktschätzung für die Bearbeitungszeit eines Auftrags – eine Verteilung ermittelt. Auf die Beschreibung des benötigten Informationsstands wird im folgenden Abschnitt näher eingegangen.

2.2.1.2 Benötigter Informationsstand

Für die Ablaufplanung unter Unsicherheit wird ein Informationsstand benötigt, der über die im vorherigen Abschnitt beschriebene Punktschätzung der Bearbeitungszeit eines Auftrags hinausgeht und stattdessen Informationen über die Verteilung sowie die minimal und maximal mögliche Bearbeitungszeit enthält.

Die Bearbeitungszeit τ_a eines Auftrags a soll daher als ein Intervall $[\tau_a^{\min}, \tau_a^{\max}]$ modelliert werden, das die minimal und maximal mögliche Bearbeitungszeit dieses Auftrags beschreibt. Für das Intervall ist je nach Umfang der vorhandenen Datenbasis eine Verteilung der Bearbeitungszeiten innerhalb des Intervalls in Form einer Häufigkeitsverteilung zu bestimmen. Bei geringem Umfang der Datenbasis soll jedoch auch von einer Gleichverteilung ausgegangen werden können.

Ein Problem, das sich bei der Ermittlung eines Intervalls und einer Häufigkeitsverteilung für einen neu einzuplanenden Auftrag ergibt, ist, dass in der Datenbasis nur sehr selten historische Aufträge mit genau den gleichen Ausprägungen der Eigenschaften wie diejenigen des einzuplanenden Auftrags vorhanden sind. Für den Aufbau des benötigten Informationsstands sind jedoch Wiederholungen in der Datenbasis notwendig, um aus den verschiedenen Bearbeitungszeiten der historischen Aufträge eine Verteilung von Bearbeitungszeitabweichungen⁷⁰ generieren zu können, mit der die Bearbeitungszeit des neu einzuplanenden Auftrags geschätzt werden kann.

Wird die Forderung nach genau den gleichen Ausprägungen der Eigenschaften abgeschwächt und werden auch solche historischen Aufträge zur Prognose der Bearbeitungszeit betrachtet, die ähnliche Ausprägungen der Eigenschaften wie der neu einzuplanende Auftrag besitzen, so kann davon ausgegangen werden, dass sich Wiederholungen häufiger einstellen. Diesem Vorgehen liegt die Annahme zugrunde, dass ähnliche Ausprägungen der Eigenschaften zu ähnlichen Bearbeitungszeiten führen. Historische Aufträge, die sich ähnlich sind, können somit zu Clustern zusammengefasst werden. Für die Cluster werden anschließend Häufigkeitsverteilungen und Intervalle der Bearbeitungszeitabweichungen

⁶⁹Siehe auch Abbildung 2.4 auf Seite 26 für ein Beispiel.

⁷⁰Siehe Abschnitt 5.2.1.4 auf Seite 91 für eine Erläuterung der Bearbeitungszeitabweichung.

der enthaltenen historischen Aufträge ermittelt. Diese können für neu einzuplanende Aufträge als Schätzer für die Bearbeitungszeit dienen.

Aus dem zuvor Geschriebenen ergibt sich die Notwendigkeit aus den zur Verfügung stehenden d Eigenschaften eines Auftrags diejenigen Eigenschaften auszuwählen bzw. neue zu erzeugen, die für die Charakterisierung eines Auftrags verwendet werden sollen. Des Weiteren ist eine Ähnlichkeits- bzw. Distanzfunktion zu verwenden, mit dem die Ähnlichkeit zweier Aufträge auf Basis der verwendeten Eigenschaften quantifiziert werden kann, um somit Cluster ähnlicher historischer Aufträge bilden zu können. Hierbei sind unterschiedliche Skalenniveaus sowie Wertebereiche der Eigenschaften zu beachten. Schließlich ist ein neu einzuplanender Auftrag auf Basis seiner Eigenschaften demjenigen Cluster zuzuordnen (d. h., es ist eine Klassierung vorzunehmen), in dem die enthaltene Häufigkeitsverteilung und das Bearbeitungszeitintervall die beste Schätzung für diesen Auftrag liefert.

2.2.2 Multikriterielle Ablaufplanung der Abfertigungsprozesse

Im Folgenden soll das Problem der Ablaufplanung der Abfertigungsprozesse im Luftfrachtterminal näher erläutert werden. Hierfür wird zunächst die Planungsumgebung als sowohl dynamisch als auch stochastisch charakterisiert. Im Anschluss wird das in Abschnitt 2.2 skizzierte Problem auf ein Mehr-Maschinen-Ablaufplanungsproblem übertragen und die sich aus der Umgebung ergebenden Anforderungen beschrieben. Abschließend werden die Restriktionen, die sich aus der Berücksichtigung des Personals und der Transportsysteme als einschränkende Ressourcen ergeben, näher betrachtet.

2.2.2.1 Charakterisierung der Planungsumgebung

Bei der Ablaufplanung wird häufig davon ausgegangen, dass die Umgebung, in der der erstellte Ablaufplan ausgeführt wird, statischer Natur ist und alle Informationen über die zukünftig zu bearbeitenden Aufträge sowie weitere planungsrelevante Informationen zum Zeitpunkt $t = 0$ der Erstellung des Ablaufplans vollständig bekannt sind und sich im Zeitverlauf nicht ändern.⁷¹ In diesem Fall handelt es sich um eine prädiktive Ablaufplanung.⁷² Der erstellte und hinsichtlich einer Zielfunktion optimierte sowie die Restriktionen einhaltende Ablaufplan wird anschließend ausgeführt und dient als Grundlage zur Koordination der Abfertigungsprozesse.

In der Praxis kann jedoch nicht von einer statischen Umgebung ausgegangen werden, da das System „Luftfrachtterminal“ dynamischen Veränderungen im Zeitverlauf unterliegt. Aufgrund des zeitlich offenen Entscheidungsfeldes⁷³ sind zum Zeitpunkt der Ablaufplanung nicht alle Informationen über die zukünftige Entwicklung vorhanden. Die Informationen bezüglich der zu bearbeitenden Aufträge sind bspw. auf eine zeitliche Reichweite von wenigen Stunden begrenzt. Für spätere Zeitpunkte sind die Auftragsinformationen noch ungenau bzw. sie sind noch gar nicht vorhanden. Daher

⁷¹Vgl. [Pin12], S. 432.

⁷²Vgl. [Sau02], S. 12.

⁷³Siehe z. B. [Sch96], Kapitel 1.1 sowie Kapitel 1.2.3.

ist die Ablaufplanung regelmäßig zu wiederholen und neu eingetroffene Aufträge sind in den bereits vorhandenen und häufig auch schon partiell ausgeführten Ablaufplan zu integrieren.

Im Gegensatz zu den im Zeitverlauf neu eintreffenden Aufträgen, die als regelmäßige Ereignisse betrachtet werden können, treten während der Ausführung des Ablaufplans ebenso Ereignisse bzw. Störungen ein, die zum Zeitpunkt der Planerstellung nicht bekannt waren und auch nicht regelmäßig auftreten. Diese unregelmäßigen Ereignisse erfordern unter Berücksichtigung ihrer Dringlichkeit eine evtl. sofortige Reaktion, die von der periodisch ausgeführten Ablaufplanung unabhängig zu erfolgen hat. In Abschnitt 2.2.3 wird auf diesen Umstand näher eingegangen.

Eine weitere häufig getroffene Annahme ist die, dass alle Parameter des Ablaufplanungsproblems mit Sicherheit bekannt bzw. hinreichend genau prognostiziert werden können und daher als deterministisch angenommen werden.⁷⁴ Diese Annahme trifft in der Praxis jedoch ebenfalls selten zu. So sind bspw. die Bearbeitungszeiten der Aufträge aufgrund der manuellen Tätigkeiten als unsicher anzunehmen.⁷⁵ Ein Ablaufplan, der von deterministischen Werten ausgeht und somit die Unsicherheit ignoriert, ist in der Umsetzung jedoch ein wenig zuverlässiger Plan. Der ermittelte Zielfunktionswert kann bspw. signifikant von dem realisierten⁷⁶ Zielfunktionswert abweichen, wenn die zur Planung verwendeten Parameter von den tatsächlich eingetretenen Daten abweichen. Aufgrund der Abweichungen zwischen den Plan- und Ist-Werten kann es zudem vorkommen, dass dem Ablaufplan nur eingeschränkt oder gar nicht gefolgt werden kann.

Werden diese Abweichungen, wie oben beschrieben, als Ereignisse betrachtet, die eine Neu- bzw. Umplanung des vorhandenen Ablaufplans anstoßen, so ist aufgrund der Vielzahl von Ereignissen sehr häufig bzw. permanent eine Ablaufplanung durchzuführen,⁷⁷ die unter Umständen umfangreiche Änderungen an dem bereits vorhandenen Plan vornimmt und damit den Zielfunktionswert laufend anpasst. Der resultierende Ablaufplan besitzt daher einen geringen Informationswert und insbesondere eine beschränkte Aussagekraft über die zu erwartende Performance des Ablaufplans.

Zusammenfassend kann die Umgebung, für die ein Ablaufplan erstellt werden soll, sowohl als dynamisch als auch stochastisch charakterisiert werden. Die Ablaufplanung hat die sich hieraus ergebenden Herausforderungen entsprechend zu berücksichtigen. Auf die Beschreibung des Ablaufplanungsproblems wird im nächsten Abschnitt näher eingegangen.

2.2.2.2 Darstellung des Ablaufplanungsproblems

Wie bereits in Abschnitt 2.2 erläutert, erfolgt durch die Ablaufplanung eine Zuordnung von Ressourcen zu Aufträgen. Die Zuordnung einer Arbeitsstation stellt dabei

⁷⁴Vgl. [ALM⁺05], S. 87.

⁷⁵Vgl. Abschnitt 2.2.

⁷⁶Der realisierte Zielfunktionswert ist die Performance (z. B. gemessen an der Anzahl säumiger Aufträge) des Ablaufplans nach der Ausführung der Aufträge und nach Bekanntwerden der tatsächlich eingetretenen Parameterwerte.

⁷⁷Vgl. [CU92], S. 153.

die wichtigste Entscheidung dar, da hierdurch auch die Belastung für die Lager und innerbetrieblichen Transportsysteme festgelegt wird. Betrachtet man die zur Verfügung stehenden Arbeitsstationen als Maschinen zur Durchführung der Aufträge, so sind die folgenden Entscheidungen zu treffen:

- Zuordnung von Aufträgen zu Maschinen, die für die Ausführung eines solchen Auftrags geeignet sind
- Bestimmung der Reihenfolge der Aufträge auf einer Maschine unter Beachtung von evtl. vorhandenen Vorrangbeziehungen zwischen den Aufträgen

Für eine statische und deterministische Umgebung lässt sich das hier betrachtete Problem der Ablaufplanung wie folgt formalisieren:⁷⁸ Gegeben ist eine Menge von n Aufträgen (Jobs) $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$, die auf einer Menge von l parallelen Maschinen $\mathcal{M} = \{m_1, m_2, \dots, m_l\}$ zu bearbeiten sind. Nicht jeder Auftrag kann auf jeder Maschine bearbeitet werden. So sind z. B. Transferaufträge, bei denen Fracht zwischen zwei Ladeeinheiten verladen wird, auf Arbeitsstationen auszuführen, die zwei Luftfrachtcontainer bzw. -paletten aufnehmen können. Die Menge an Maschinen, auf denen ein Auftrag a ausgeführt werden kann, wird mit \mathcal{M}_a beschrieben. Die Maschinen innerhalb dieser Menge sind identisch, d. h., ein Auftrag benötigt unabhängig von der gewählten Maschine die gleiche Bearbeitungszeit. Diese Bearbeitungszeit wird für jeden Auftrag als τ_a mit $a \in \mathcal{A}$ angegeben. Weiterhin besitzt jeder Auftrag eine Bereitstellungs- bzw. Einlastzeit r_a , die angibt, wann der entsprechende Auftrag frühestens auf einer Maschine bearbeitet werden kann, sowie einen Fälligkeitstermin d_a , zu dem der Auftrag spätestens fertiggestellt sein muss. Schließlich besitzt jeder Auftrag eine durch einen Gewichtungsfaktor w_a beschriebene Priorität, die die Wichtigkeit des Auftrags angibt. Eilaufträgen wird bspw. eine höhere Priorität zugewiesen, da sie bei Verzug zu höheren Verzugskosten als normale Aufträge führen und sie daher bei der Planung stärker berücksichtigt werden sollten.

Zwischen den Aufträgen können zudem Vorrangbeziehungen (*prec*) bestehen, die angeben, dass ein Auftrag erst starten kann, wenn andere Aufträge fertiggestellt wurden. Dies ist z. B. für Aufbauaufträge der Fall, bei denen Fracht konsolidiert werden soll, die zuvor durch Abbauaufträge dekonsolidiert und hierdurch verfügbar gemacht werden muss. Diese Vorrangbeziehungen können durch einen gerichteten azyklischen Graphen $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ repräsentiert werden. Dabei entsprechen die Knoten des Graphen den Aufträgen und die gerichteten Kanten $\mathcal{E} \subset \mathcal{A} \times \mathcal{A}$ entsprechen den Vorrangbeziehungen. Eine Vorrangbeziehung $(a_i, a_j) \in \mathcal{E}$ beschreibt, dass der Auftrag $a_i \in \mathcal{A}$ vor dem Beginn von Auftrag $a_j \in \mathcal{A}$ fertiggestellt sein muss.⁷⁹

Als Zielfunktion bzw. Performancemaß soll die gewichtete Anzahl säumiger Aufträge minimiert werden. Hierfür werden zunächst Strafkosten U_a für jeden Auftrag $a \in \mathcal{A}$ in

⁷⁸Diese Einschränkung wird nach der Formalisierung aufgehoben und die Umgebung, wie sie in Abschnitt 2.2.2.1 beschrieben wurde, berücksichtigt.

⁷⁹Vgl. [Bru07], S. 3.

Abhängigkeit seines Fertigstellungszeitpunkts C_a sowie seines Fälligkeitstermins d_a wie folgt definiert:

$$U_a = \begin{cases} 1, & \text{wenn } C_a > d_a \\ 0 & \text{in allen anderen Fällen} \end{cases} \quad \forall a \in \mathcal{A} \quad (2.2)$$

Die zu minimierende Zielfunktion wird dann durch $\sum_{a \in \mathcal{A}} w_a U_a$ angegeben. Ergebnis der Ablaufplanung ist ein Ablaufplan ap , der jeden Auftrag genau einer Maschine zuweist. Ist zwischen dem geplanten und realisierten Ablaufplan zu unterscheiden, so wird im Folgenden ersterer als ap^p und letzterer als ap^r bezeichnet.

Zusammenfassend kann das hier betrachtete Ablaufplanungsproblem für eine statische und deterministische Umgebung als $P|prec, r_a, d_a, \mathcal{M}_a|\sum w_a U_a$ in der Drei-Felder-Notation⁸⁰ nach Graham et al. beschrieben werden.

Des Weiteren wird für das hier zu betrachtende Ablaufplanungsproblem von folgenden Einschränkungen ausgegangen:

- Jede Maschine $m \in \mathcal{M}$ kann zur gleichen Zeit höchstens einen Auftrag $a \in \mathcal{A}$ bearbeiten.
- Jeder Auftrag $a \in \mathcal{A}$ kann zur gleichen Zeit auf höchstens einer Maschine $m \in \mathcal{M}_a$ bearbeitet werden.
- Eine Unterbrechung eines einmal gestarteten Auftrags $a \in \mathcal{A}$ auf einer Maschine $m \in \mathcal{M}_a$ durch z. B. einen höher priorisierten Auftrag ist nicht erlaubt (ausgenommen Maschinenausfälle), d. h., der Auftrag wird bis zu seiner Fertigstellung auf der Maschine ausgeführt.
- Rüstzeiten und -kosten sind zu vernachlässigen, da ein Auftrag $a \in \mathcal{A}$ grundsätzlich auf jeder Maschine der Menge $\mathcal{M}_a \subseteq \mathcal{M}$ ohne vorheriges Rüsten bearbeitet werden kann.

Betrachtet man nun die in Abschnitt 2.2.2.1 beschriebene dynamische Umgebung, so ist die Menge \mathcal{A} nicht mehr auf n Aufträge begrenzt, sondern aufgrund des offenen Entscheidungsfeldes unbegrenzt $\mathcal{A} = \{a_1, a_2, \dots, a_n, \dots\}$. Bei der i -ten Erstellung eines Ablaufplans zum Zeitpunkt t_i ist von dieser Menge nur eine Teilmenge $\mathcal{A}_{t_i} = \{a_1, a_2, \dots, a_n\} \subset \mathcal{A}$ von n Aufträgen bekannt, wobei a_n dem letzten Auftrag entspricht über den Informationen vorhanden sind. Der durch die Ablaufplanung erstellte Plan ap_i wird anschließend ausgeführt.

Zu einem späteren Zeitpunkt $t_{i+1} > t_i$ wird die Ablaufplanung wiederholt und die bis dahin neu hinzugekommenen Aufträge $\mathcal{A}_{t_{i+1}}^N = \{a_{n+1}, a_{n+2}, \dots, a_{n+n'}\}$ werden in

⁸⁰Die Drei-Felder-Notation $\alpha|\beta|\gamma$ wurde von Graham et al. (vgl. [GLLR79]) zur Klassifikation von Scheduling-Problemen eingeführt. Das α -Feld beschreibt die Maschinen-Umgebung. Charakteristika der Aufträge und Scheduling-Restriktion werden durch das β -Feld beschrieben. Schließlich beschreibt das γ -Feld die zu optimierende Zielfunktion des Problems (vgl. u. a. [Leu04], Kapitel 1, S. 6 ff.; [Bru07], S. 3–7). Siehe auch [Pin12], S. 17, für die Erweiterung der Notation des β -Feldes um \mathcal{M}_a , wenn ein Auftrag a nicht auf allen m Maschinen bearbeitet werden kann.

den bereits in der Ausführung befindlichen Ablaufplan ap_i integriert. Für einige der Aufträge aus \mathcal{A}_{t_i} können die in ap_i getroffenen Entscheidungen revidiert werden. Hierzu bezeichne s_a die Startzeit des Auftrags a und f_a den Zeitpunkt, zu dem Vorbereitungen für den Auftrag a an der Arbeitsstation begonnen haben. Es gilt $f_a < s_a < C_a$. Die Menge \mathcal{A}_{t_i} kann somit zum Zeitpunkt t_{i+1} in die folgenden vier Mengen unterteilt werden:

- $\mathcal{A}_{t_{i+1}}^C = \{a \mid a \in \mathcal{A}_{t_i} \wedge f_a < t_{i+1} \wedge s_a < t_{i+1} \wedge C_a < t_{i+1}\}$: Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, die bereits fertiggestellt wurden.
- $\mathcal{A}_{t_{i+1}}^R = \{a \mid a \in \mathcal{A}_{t_i} \wedge f_a < t_{i+1} \wedge s_a < t_{i+1} \wedge C_a = \text{null}\}$: Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, die bereits auf den Arbeitsstationen gestartet wurden.
- $\mathcal{A}_{t_{i+1}}^F = \{a \mid a \in \mathcal{A}_{t_i} \wedge f_a < t_{i+1} \wedge s_a = \text{null} \wedge C_a = \text{null}\}$: Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, für die an den Arbeitsstationen bereits eine Vorbereitung stattgefunden hat (z. B. Vorpufferung von Fracht). Die Aufträge dieser Menge sind somit fixiert.
- $\mathcal{A}_{t_{i+1}}^P = \{a \mid a \in \mathcal{A}_{t_i} \wedge f_a = \text{null} \wedge s_a = \text{null} \wedge C_a = \text{null}\}$: Menge der zum Zeitpunkt t_{i+1} bekannten Aufträge, die lediglich auf eine Arbeitsstation eingeplant sind.

Hierbei gilt $\mathcal{A}_{t_i} = \mathcal{A}_{t_{i+1}}^C \cup \mathcal{A}_{t_{i+1}}^R \cup \mathcal{A}_{t_{i+1}}^F \cup \mathcal{A}_{t_{i+1}}^P$ und jeder Auftrag ist genau einer der vier Mengen zugeordnet. Von der Gesamtmenge $\mathcal{A}_{t_{i+1}} = \mathcal{A}_{t_i} \cup \mathcal{A}_{t_{i+1}}^N$ der zum Zeitpunkt t_{i+1} bekannten Aufträge können lediglich Aufträge aus der Menge $\mathcal{A}_{t_{i+1}}^P$ umgeplant werden, da die Zuordnungen in ap_i noch vorläufig sind. Die Menge der erstmalig einzuplanenden sowie neu einzuplanenden Aufträge zum Zeitpunkt t_{i+1} ergibt sich somit aus $\mathcal{A}_{t_{i+1}}^{\text{Pb}} = \mathcal{A}_{t_{i+1}}^N \cup \mathcal{A}_{t_{i+1}}^P$. Das Ergebnis der Ablaufplanung ist der neue Ablaufplan ap_{i+1} , der bis zum nächsten Planungszeitpunkt t_{i+2} ausgeführt wird.

Für Aufträge aus $\mathcal{A}_{t_{i+1}}^P$ ist zu beachten, dass bereits Sekundärplanungen erfolgt sein können, die bspw. auf den berechneten Endterminen der Aufträge beruhen. Daher sollten Änderungen dieser Aufträge unter Berücksichtigung der in Ablaufplan ap_i getroffenen Entscheidungen erfolgen, bspw. über die Definition eines Stabilitätsmaßes. Dieses Maß quantifiziert die Änderungen an diesen Aufträgen zwischen dem alten Ablaufplan ap_i und dem neuen Plan ap_{i+1} . Zwischen dem Stabilitätsmaß und dem zuvor definierten Performancemaß besteht ein Trade-off. Auf der einen Seite schränkt eine hohe Stabilität die Möglichkeiten ein, die Anzahl säumiger Aufträge durch Umplanungen zu vermindern. Auf der anderen Seite kann eine Optimierung des Performancemaßes dazu führen, dass viele Aufträge aus Menge $\mathcal{A}_{t_{i+1}}^P$ umgeplant werden müssen, was zu einer geringen Stabilität führt.

Abbildung 2.3 zeigt ein Beispiel mit sieben Aufträgen und zwei Maschinen in einem Gantt-Diagramm.⁸¹ Zum Zeitpunkt $t_0 = 0$ wurde für die Aufträge eine Ablaufplanung

⁸¹Gantt-Diagramme, benannt nach Henry L. Gantt (1861 – 1919), eignen sich zur Darstellung von Maschinenbelegungsplänen. In den Diagrammen sind die Zuordnungen sowie Belegungsdauern der Maschinen (abgebildet an der Ordinate) durch die Aufträge (abgebildet als Balken auf der Zeitachse [Abszisse]) dargestellt (vgl. [ZB05], S. 7).

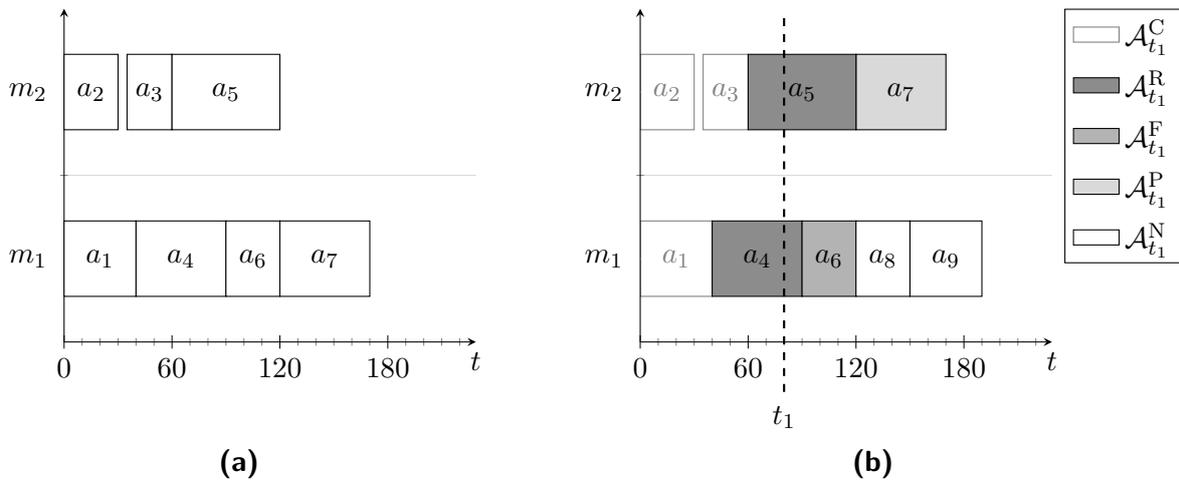


Abbildung 2.3: Dynamische Ankunft neuer Aufträge und ihre Berücksichtigung in partiell ausgeführten Ablaufplänen: (a) Initial erstellter Ablaufplan ap_0 (b) Ablaufplan ap_1 , erstellt zum Zeitpunkt $t_1 = 80$

durchgeführt und der ermittelte Ablaufplan ap_0 ist in Abbildung 2.3(a) dargestellt. Zum Zeitpunkt $t_1 = 80$ wird die Ablaufplanung erneut durchgeführt. Die neu hinzugekommenen Aufträge $\mathcal{A}_{t_1}^N = \{a_8, a_9\}$ sind dabei in den bereits in der Ausführung befindlichen Plan ap_0 zu integrieren, wobei für beide Aufträge angenommen wird, dass sie nur auf Maschine m_1 ausgeführt werden können. Es wird die Menge $\mathcal{A}_{t_1}^{pb} = \mathcal{A}_{t_1}^N \cup \mathcal{A}_{t_1}^P$ der erstmalig sowie der neu einzuplanenden Aufträge gebildet und ein neuer Ablaufplan ap_1 erzeugt, der in Abbildung 2.3(b) dargestellt ist. Auftrag a_7 wurde dabei umgeplant und von Maschine m_1 zu m_2 verschoben.

Wird neben der dynamischen Umgebung nun auch die stochastische Umgebung betrachtet, so sind die Bearbeitungszeiten der Aufträge keine deterministischen Werte mehr, sondern unsicher und werden als diskrete Zufallsvariablen X_a modelliert. Die realisierte Bearbeitungszeit τ_a eines Auftrags a wird erst nach dessen Fertigstellung bekannt. Informationen über die Verteilung⁸² der Zufallsvariablen sind durch die Analyse der historischen Informationen über vergangene Aufträge vorhanden (siehe Abschnitt 2.2.1). Des Weiteren gilt für jede Bearbeitungszeit τ_a , dass diese innerhalb eines Intervalls $\tau_a \in [\tau_a^{\min}, \tau_a^{\max}]$ liegt.

Eine häufig durchgeführte Vereinfachung besteht darin, die Unsicherheit aus der Ablaufplanung auszuschließen, indem für jeden stochastischen Parameter ein deterministischer Ersatzwert verwendet wird. Beispielsweise könnte für die Bearbeitungszeiten τ_a der Aufträge der Erwartungswert $E[X_a]$ der Zufallsvariablen der jeweiligen Bearbeitungszeit verwendet werden.⁸³ Dieses Vorgehen ignoriert mögliche Ausprägungen der Bearbeitungszeiten, die nicht denen des verwendeten Erwartungswertes entsprechen.

⁸²Es wird davon ausgegangen, dass die Bearbeitungszeiten der Aufträge stochastisch unabhängig sind, sowohl untereinander als auch von dem Startzeitpunkt des Auftrags.

⁸³Diese Art der Modellierung wird von Scholl als „Deterministisches Erwartungswertmodell“ ([Sch01], S. 187) bezeichnet.

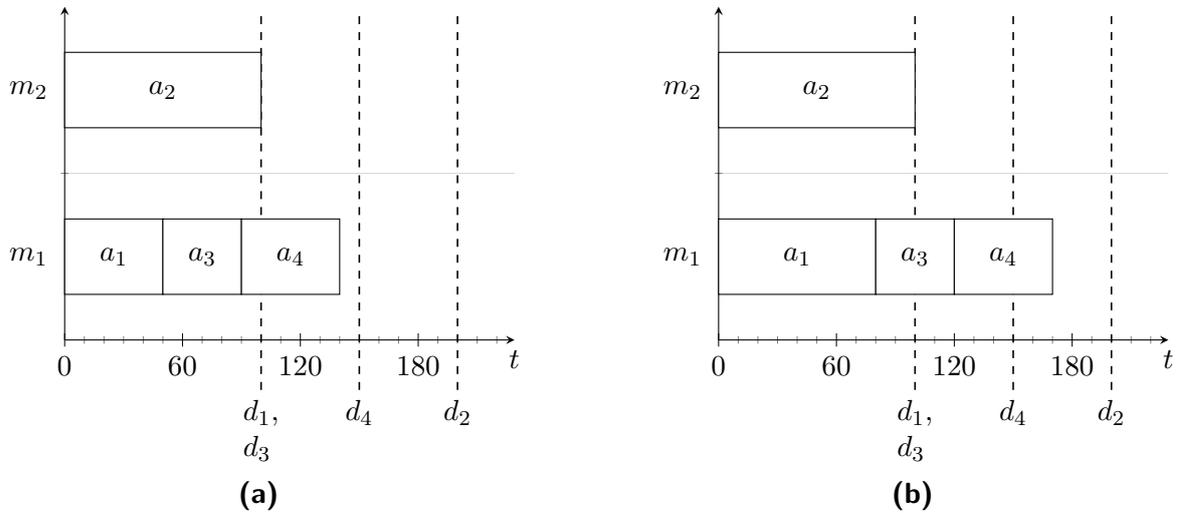


Abbildung 2.4: Auswirkungen stochastischer Bearbeitungszeiten der Aufträge auf den Ablaufplan: (a) Initial erstellter Ablaufplan ap^p (b) Realisierter Ablaufplan ap^r

Der Zielfunktionswert des realisierten Ablaufplans kann bei einer Abweichung von diesen Erwartungswerten substanziell von dem berechneten optimalen Wert abweichen.⁸⁴

Abbildung 2.4 zeigt ein Beispiel mit vier Aufträgen und ihren entsprechenden Fälligkeitsterminen sowie zwei Maschinen. Zusätzlich gilt für a_2 , dass dieser Auftrag nur auf m_2 eingeplant werden darf. Für Auftrag a_1 wird von einer stochastischen und gleichverteilten Bearbeitungszeit im Intervall $\tau_1 \in [20, 80]$ und für die anderen Aufträge von deterministischen Bearbeitungszeiten ausgegangen. Nach dem oben beschriebenen Vorgehen würde auf Auftrag a_1 eine erwartete Bearbeitungszeit von 50 Zeiteinheiten verwendet. Der Ablaufplan könnte dann z. B. mit der Prioritätsregel *Earliest Deadline First* (EDF)⁸⁵ erstellt werden. Der so erstellte Ablaufplan ist in Abbildung 2.4(a) dargestellt und weist keine säumigen Aufträge auf. Wenn der Auftrag a_1 statt der erwarteten Zeit jedoch 80 Zeiteinheiten benötigt, so sind die Aufträge a_3 und a_4 säumig, da sie einen späteren Fertigstellungszeitpunkt als ihre Fälligkeitstermine besitzen (siehe Abbildung 2.4(b)).

Ein weiteres Vorgehen besteht darin, den Erwartungswert der Zielfunktion,⁸⁶ in diesem Fall $E[\sum w_a U_a]$, zu optimieren, wobei der Erwartungswert über eine als bekannt⁸⁷ vorausgesetzte Verteilungsfunktion gebildet wird.⁸⁸ Der so ermittelte Ablaufplan ist stochastisch optimal. Interpretiert werden kann der ermittelte Zielfunktionswert als das

⁸⁴Vgl. [KY97], S. 3.

⁸⁵Bei der EDF-Regel werden die Aufträge aufsteigend nach ihrem Fälligkeitstermin sortiert und anschließend eingeplant.

⁸⁶Scholl spricht hier von dem „Erwartungswert-Kriterium“ [Sch01], S. 124.

⁸⁷Diese Annahme ist entscheidend für die Anwendung der stochastischen Optimierung (vgl. [Thi08], S. 290). In der Praxis ist die Ermittlung einer solchen Verteilung jedoch eine schwierige Aufgabe und die über die Verteilung getroffenen Annahmen sind häufig ungeeignet für Entscheidungsumgebungen, bei denen einige Faktoren die Unsicherheit vieler Elemente bestimmen (siehe [KY97], S. 5).

⁸⁸Vgl. [KY97], S. 3f.

Ergebnis, „das bei hinreichend häufiger Wiederholung des zugrundeliegenden Zufallsexperiments durchschnittlich zu erwarten ist“⁸⁹. Die konkrete Entscheidungssituation, für die der stochastisch optimale Ablaufplan ermittelt wird, tritt in genau dieser Ausprägung jedoch nicht erneut ein und ist damit einmalig bzw. nicht wiederholbar. Der realisierte Zielfunktionswert kann auch in diesem Fall erheblich von dem ermittelten Erwartungswert abweichen. Ist diese Abweichung negativ (wenn mehr säumige Aufträge auftreten als durch den Erwartungswert angegeben), so fehlt aufgrund der Einmaligkeit der Entscheidung die Möglichkeit, das Ergebnis durch eine positive Abweichung zu kompensieren.⁹⁰

Der Zielfunktionswert eines generierten Ablaufplans hängt von der eintretenden Umweltsituation (der realisierten Bearbeitungszeiten der Aufträge) ab und besitzt somit eine Verteilung möglicher Ausprägungen. Die beiden oben beschriebenen Vorgehensweisen berücksichtigen bei der Erstellung der Pläne jedoch entweder nur eine mögliche zukünftige Situation (Erwartungswert der Bearbeitungszeiten) oder den Erwartungswert des Zielfunktionswertes, der sich bei hinreichend häufiger Wiederholung einstellen würde. Damit ist keines dieser Vorgehen geeignet, das Risiko bzw. die Abweichung vom berechneten Zielfunktionswert adäquat zu berücksichtigen und der Zielfunktionswert hat, wie in Abschnitt 2.2.2.1 beschrieben, eine geringe Aussagekraft. Risikoaverse Entscheidungsträger, die für die Planerstellung verantwortlich sind und die ex post anhand der realisierten Performance des Ablaufplans unter der Annahme evaluiert werden, dass die eingetretene Umweltsituation bereits zum Zeitpunkt der Entscheidung für einen Ablaufplan bekannt gewesen wäre, sind hingegen an einem Ablaufplan interessiert, der für möglichst alle⁹¹ zukünftigen Umweltsituationen gute Resultate liefert und dabei die Möglichkeit bietet, den Ablaufplan gegen die ungünstigste Umweltsituation (Worst-Case-Szenario) abzusichern.⁹²

Eine Absicherung gegen diese ungünstigste Umweltsituation ist jedoch häufig mit sehr konservativen bzw. hinsichtlich der Performance des Ablaufplans schlechten Ergebnissen verbunden. Vor diesem Hintergrund sollte die Möglichkeit bestehen, ein Anspruchsniveau für die Performance des Ablaufplans vorzugeben, das in jedem Fall eingehalten werden muss, und somit ein gewisses Maß an Risiko zu tolerieren, wenn hierdurch eine wesentlich bessere Performance erzielt werden kann. Ablaufpläne, die dieses Anspruchsniveau erfüllen, sollen im Folgenden als *robust* bezeichnet werden. Zur Quantifizierung der Robustheit ist ein Maß zu erstellen, das bei der Ablaufplanung verwendet werden kann. Für dieses Robustheitsmaß soll erneut der Trade-off zu den beiden zuvor definierten Maßen betrachtet werden.

Werden die drei zuvor angesprochenen Maße (Performance-, Robustheits- und Stabilitätsmaß) als gleichrangige Zielfunktionen in der Ablaufplanung berücksichtigt, so liegt ein

⁸⁹[Sch01], S. 124. Vgl. auch Kouvelis & Yu, die dies als „*Long Run Optimality*“ ([KY97], S. 3) bezeichnen.

⁹⁰Vgl. [Sch01], S. 124 f.; [KY97], S. 5 f.

⁹¹Hier soll von normalen Umweltsituationen ausgegangen werden, d. h. Situationen, in denen die Bearbeitungszeiten der Aufträge innerhalb ihrer berechneten Verteilungen liegen. Gravierende Veränderungen, die in Abschnitt 2.2.2.1 als unregelmäßige Ereignisse (z. B. Maschinenausfälle) beschrieben wurden, sind hiervon ausgenommen.

⁹²Vgl. [KY97], S. 5 f.; [Sch01], S. 90–93.

multikriterielles Ablaufplanungsproblem vor. Ein solches Mehrzieloptimierungsproblem (engl. *Multi-objective Optimization Problem* (MOP)) kann nach Talbi folgendermaßen definiert werden:⁹³

$$MOP = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{u. B. d. N. } x \in \mathcal{S} \end{cases}$$

Hierbei bezeichne $f_i(x)$ eine der n mit $n \geq 2$ Zielfunktionen, $F(x)$ den Vektor der zu optimierenden Zielfunktionen, x den Vektor der Entscheidungsvariablen und \mathcal{S} die Menge der zulässigen Lösungen. Aufgrund des zuvor beschriebenen Trade-offs zwischen den Zielfunktionen ist regelmäßig nicht von einer Lösung x^* auszugehen, die für jede Zielfunktion optimal ist, d. h., $f_i(x^*) < f_i(x), \forall x \in \mathcal{S}, i = 1, 2, \dots, n$ ist unwahrscheinlich. Daher wird im Folgenden auf das Konzept der Pareto-Dominanz zurückgegriffen. Hierbei dominiert ein Zielfunktionsvektor \vec{u} einen Vektor \vec{v} , bezeichnet als $\vec{u} \prec \vec{v}$, wenn jede Komponente von \vec{u} kleiner oder gleich der entsprechenden Komponente von \vec{v} ist und in einer Komponente strikt kleiner:

$$\forall i \in \{1, 2, \dots, n\} : u_i \leq v_i \wedge \exists i \in \{1, 2, \dots, n\} : u_i < v_i$$

Auf Basis der Pareto-Dominanz kann anschließend die Pareto-Optimalität definiert werden. Eine Lösung $x^* \in \mathcal{S}$ ist dabei pareto-optimal, wenn Folgendes gilt:

$$\forall x \in \mathcal{S} : F(x) \not\prec F(x^*) \quad (2.3)$$

Die pareto-optimalen Lösungen bilden zusammen die pareto-optimale Menge des MOPs:

$$\mathcal{P}^* = \left\{ x \in \mathcal{S} \mid \nexists x' \in \mathcal{S}, F(x') \prec F(x) \right\}$$

Für ein MOP und die Menge \mathcal{P}^* kann schließlich die Pareto-Front bestimmt werden:

$$\mathcal{PF}^* = \left\{ F(x) \mid x \in \mathcal{P}^* \right\}$$

Neben den zu berücksichtigenden Anforderungen, die sich aus der stochastischen und dynamischen Umgebung ergeben, sind bei der Ablaufplanung weitere Restriktionen zu beachten. Diese ergeben sich aus den für die Abfertigungsprozesse benötigten personellen Ressourcen und den zur Beförderung der Fracht eingesetzten Transportsystemen. Auf diese Aspekte wird in den nächsten beiden Abschnitten näher eingegangen.

2.2.2.3 Berücksichtigung der Personalkapazitäten

Im vorherigen Abschnitt wurde die Zuordnung von Aufträgen zu Maschinen und ihre Reihenfolge auf diesen Maschinen betrachtet. Die Maschinen stellten dabei die einzige knappe Ressource dar. Wenn das Personal als weitere Ressource berücksichtigt wird,

⁹³Siehe [Tal09], S. 310 f.

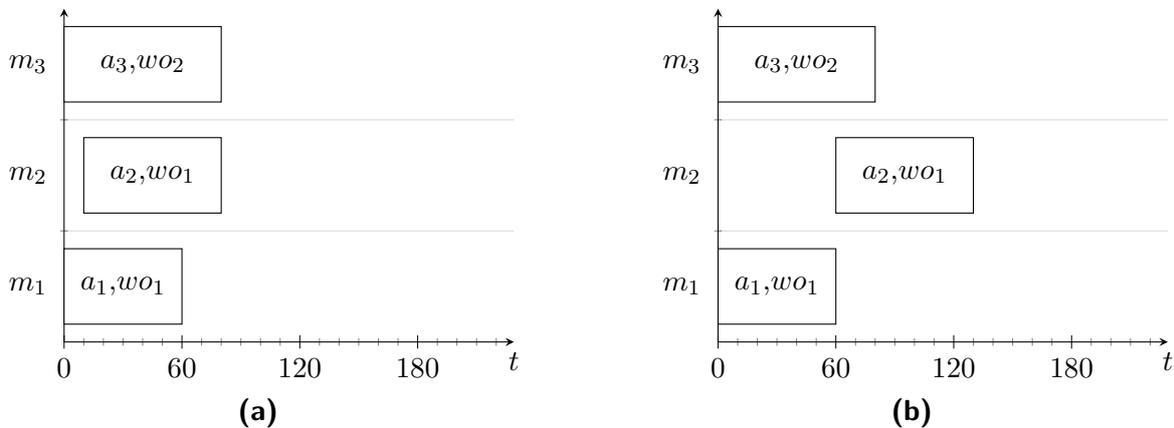


Abbildung 2.5: Auswirkungen der Ressource „Personal“ auf den Ablaufplan: (a) Ablaufplan ohne Berücksichtigung des Personals (b) Ablaufplan unter Berücksichtigung des Personals

ergibt sich eine weitere Einschränkung.⁹⁴ Aufgrund der manuellen Tätigkeit der Frachtabfertigung bei den Auf- und Abbauprozessen können nur solche Aufträge aktiv bearbeitet werden, bei denen Personal in der notwendigen Stärke und Qualifikation verfügbar ist. Aufträge, bei denen zwar die Fracht an der Arbeitsstation bereitgestellt wurde, aber das benötigte Personal z. B. noch durch einen anderen Auftrag gebunden ist, blockieren die entsprechende Station. Durch die Personaleinsatzplanung, die hier als Teil der Ablaufplanung betrachtet wird, sind daher die folgenden Entscheidungen zu treffen:

- Zuordnung von Personal in der benötigten Stärke und Qualifikation zu Aufträgen
- Bestimmung der Reihenfolge, in der eine Arbeitskraft die ihr zugeordneten Aufträge bearbeitet

Das Personal wird hierzu als Menge $\mathcal{WO} = \{wo_1, wo_2, \dots, wo_n\}$ modelliert. Ein Teil des Personals $\mathcal{WO}^{\text{Br}} \subset \mathcal{WO}$ kann hierbei nur für die Abbauaufträge eingesetzt werden. Der andere Teil $\mathcal{WO}^{\text{Bu}} \subset \mathcal{WO} \setminus \mathcal{WO}^{\text{Br}}$ kann aufgrund einer höheren Qualifikation sowohl für Abbau- als auch Aufbauaufträge eingeplant werden. Hinsichtlich der Bedienung der Arbeitsstationen wird keine Unterscheidung der Qualifikation vorgenommen.

Ein Beispiel mit drei Aufträgen und drei Maschinen sowie zwei Mitarbeitern ist in Abbildung 2.5 dargestellt. Bleibt das Personal unberücksichtigt, so könnte durch eine Ablaufplanung der Ablaufplan in Abbildung 2.5(a) realisiert werden. Wird das Personal jedoch als beschränkende Ressource berücksichtigt, so ergibt sich der Ablaufplan in Abbildung 2.5(b), bei dem Mitarbeiter wo_1 Auftrag a_2 erst nach der Abarbeitung von a_1 durchführen kann.

⁹⁴Diese Einschränkung ergibt sich dann, wenn weniger Personal als Arbeitsstationen zur Verfügung steht bzw. wenn durch größere Aufträge so viel Personal gebunden ist, dass nicht jede Arbeitsstation zur gleichen Zeit bedient werden kann. Ein Grund dafür, weniger Personal als Arbeitsstationen vorzuhalten, kann bspw. damit motiviert sein, mit der Ressource „Personal“ effizienter umzugehen bzw. Personalkosten einzusparen.

2.2.2.4 Berücksichtigung der Förderkapazitäten

Wie in Abschnitt 2.2 bereits erwähnt, erzeugen die durch die Ablaufplanung erstellten Ablaufpläne Belastungen im Lager (sowohl in den Sendungslagern als auch in dem ULD-Lager) und für die innerbetrieblichen Transportsysteme, die Lager und Arbeitsstationen miteinander verbinden. Übersteigt der sich aus einem Ablaufplan ergebene Materialfluss zu bzw. von einer Arbeitsstation die maximalen Förderkapazitäten der beteiligten Transportsysteme, so sind diese als weitere einschränkende Ressource zu betrachten.

Aufbauaufträge können nur aktiv durch das eingeplante Personal bearbeitet werden, wenn ein leeres Ladehilfsmittel für das Flugzeug und mindestens eine Ladeeinheit aus dem Sendungslager (hier: Gitterbox) mit der benötigten Fracht für den Auftrag an der Arbeitsstation vorhanden ist. Ein überlastetes Transportsystem kann hier zu Wartezeiten an der Arbeitsstation führen. Bei Aufbauaufträgen ist der Bedarf, d. h. die Anzahl der durch das Transportsystem zu bewegenden Gitterboxen aus dem Sendungslager, bereits zum Zeitpunkt der Ablaufplanung bekannt. Für Abbauaufträge gelten die gleichen Einschränkungen, allerdings ist hier der tatsächliche Bedarf an benötigten Ladehilfsmitteln aus dem Sendungslager zur Vereinzelnung der Fracht aus der Flugzeugladeeinheit nicht im Vorhinein exakt bekannt. Daher kann im Laufe eines Abbauauftrags die Notwendigkeit entstehen, weitere leere Gitterboxen aus dem Sendungslager anzufordern.

Es wird angenommen, dass die Route, die eine Ladeeinheit im Lager und auf dem Transportsystem verwendet, eindeutig⁹⁵ durch den Ausgangs- und Zielpunkt der Ladeeinheit bestimmt ist. Daher sind die Transportwege sowie die Belastungen der einzelnen Routen durch einen Ablaufplan determiniert. Somit besteht ein Entscheidungsproblem. Auf der einen Seite sollte ein Auftrag der Arbeitsstation zugewiesen werden, für den die dem Auftrag zugehörige Fracht aus dem Lager eine möglichst kurze Route zurückzulegen hat und dadurch möglichst wenig Transportzeit benötigt wird. Auf der anderen Seite kann dieses Vorgehen dazu führen, dass einige Routen überlastet werden und die Transportzeit hierdurch steigt. In diesem Fall ist die Zuweisung einer weiter entfernten Arbeitsstation sinnvoll, sofern die hieraus resultierende längere Transportzeit nicht zu einer Überschreitung des für den Auftrag geltenden Fälligkeitstermins führt. Ziel der Berücksichtigung der Förderkapazitäten bei der Ablaufplanung ist es daher, diesen Trade-off in Hinblick auf die für die Aufträge geltenden Fälligkeitstermine zu lösen.

2.2.3 Ablaufsteuerung der Abfertigungsprozesse

Der für einen bestimmten in die Zukunft reichenden Zeitraum erstellte Ablaufplan ist durch ein Ablaufsteuerungssystem auszuführen und zu überwachen. Die einer Arbeitsstation zugeordneten Aufträge werden dabei sukzessiv von dem für den Auftrag eingeplantem

⁹⁵Die bei einigen komplexen Systemen (auf Basis von Echtzeitinformationen über die aktuelle Belastungssituation) während der Ausführung dynamisch erzeugten Alternativrouten (vgl. z. B. [XZMC14]; [SCXL10]; [MCLL09]; [CLM06]) werden im Rahmen der hier durchgeführten Ablaufplanung nicht berücksichtigt. Der Fokus liegt hier auf der Erstellung von Ablaufplänen, die zu keinen Belastungssituationen führen, die eine Verwendung von Alternativrouten notwendig machen.

Personal abgearbeitet und durch die mobile Datenerfassung wird der aktuelle Fortschritt an das Ablaufsteuerungssystem gemeldet. Dieses führt einen Soll-/Ist-Vergleich der realisierten und geplanten Zeiten durch und kann so Abweichungen identifizieren. Des Weiteren werden durch die Ablaufsteuerung unregelmäßige, d. h. nicht vorhersehbare, Ereignisse (wie z. B. der Ausfall einer Arbeitsstation) erkannt, die die Ausführung des Ablaufplans beeinträchtigen.

Nicht jede Abweichung von den geplanten Endzeitpunkten der Aufträge führt zu einer Störung im Sinne einer Änderung der Anzahl säumiger Aufträge. Hierfür wurden bei der Ablaufplanung Informationen über Unsicherheiten bei den Bearbeitungszeiten berücksichtigt. Erst die Überschreitung des für einen Auftrag angenommenen Intervalls der Bearbeitungszeit führt zu einer Störung, dessen Auswirkungen betrachtet werden müssen. Die Auswirkungen sind entweder durch lokale Änderungen an dem Ablaufplan zu kompensieren, sodass der Gesamtplan nicht beeinträchtigt wird (bspw. zusätzliches Personal) oder die Ablaufplanung muss erneut durchgeführt werden. In diesem Fall sind bereits getroffene Entscheidungen des derzeit aktiven Ablaufplans bei der Umplanung durch die Ablaufplanung zu berücksichtigen und möglichst wenige Änderungen vorzunehmen.

Für die erkannten Ereignisse können nach Ouelhadj & Petrovic⁹⁶ ressourcenbezogene sowie auftragsbezogene Ereignisse unterschieden werden. Erstere beziehen sich auf die Ressourcen, die für die Ausführung eines Auftrags benötigt werden. Beispiele für Ereignisse im Kontext des Luftfrachtterminals sind z. B.:

- der Ausfall einer Arbeitsstation oder Teilen des Transportsystems zur Ver- und Entsorgung mit der benötigten Fracht,
- die Nichtverfügbarkeit von Personal
- sowie die Nichtverfügbarkeit von Fracht durch verzögerte Abbauaufträge.

Auftragsbezogene Ereignisse hingegen beziehen sich direkt auf die Aufträge, wie z. B.:

- eine Bearbeitungszeit, die nicht innerhalb des Bearbeitungszeitintervalls des Auftrags liegt,
- das Eintreffen von Eilaufträgen,
- der Abbruch von Aufträgen oder die Änderung der Auftragspriorität.

Auch diese Ereignisse erfordern lokale Änderungen am Ablaufplan bzw. eine Neu- oder Umplanung unter Verwendung der Ablaufplanung, die von der periodisch durchgeführten Ablaufplanung abweicht.

⁹⁶Vgl. [OP09], S. 418.

2.3 Anforderungen an die Verfahren zur Problemlösung

Im Folgenden werden basierend auf dem beschriebenen Problem der Planung und Steuerung der Abfertigungsprozesse in einem Luftfrachtterminal die Anforderungen an die Verfahren der einzelnen Problemfelder definiert.

2.3.1 Anforderungen an ein Verfahren zur Datenanalyse der Abfertigungsprozesse

Die folgenden Anforderungen ergeben sich für ein Verfahren, das auf Basis historischer Betriebsdaten über vergangene Auf-, Abbau- und Transferaufträge Informationen über die Bearbeitungszeitunsicherheiten der Abfertigungsprozesse ermittelt:

- **Extraktion von Clustern ähnlicher Aufträge auf Basis historischer Betriebsdaten**
Historische Betriebsdaten über vergangene Abfertigungsprozesse wurden als wichtige Quelle für Informationen über Unsicherheiten der Bearbeitungszeiten von Aufträgen identifiziert. Da die Aufträge in ihren Eigenschaften selten vollständig identisch sind, sollte das Verfahren daher in der Lage sein, Cluster ähnlicher Aufträge zu bilden. Hierbei sind unterschiedliche Skalenniveaus und Wertebereiche der Eigenschaften zu berücksichtigen.
- **Ermittlung von Häufigkeitsverteilungen für die Cluster**
Informationen über die Bearbeitungszeiten von Aufträgen eines Clusters sind durch das Verfahren geeignet, zu Häufigkeitsverteilungen zu aggregieren, um Informationen für die Prognose von Bearbeitungszeiten neuer Aufträge bereitzustellen.
- **Ermittlung von Bearbeitungszeitintervallen für die Cluster**
Auf Basis der Häufigkeitsverteilungen soll es möglich sein, für die Cluster Bearbeitungszeitintervalle zu definieren, die bei Erstellung robuster Ablaufpläne und der Absicherung gegen die ungünstigste Umweltsituation Verwendung finden.
- **Klassierung neuer Aufträge anhand von Eigenschaften**
Für neu eintreffende Aufträge ist eine genaue Abschätzung der Bearbeitungszeit essenziell. Anhand auftragsspezifischer Eigenschaften sollte das Verfahren mit Hilfe einer Klassierung in der Lage sein, Verteilungsinformationen über die Bearbeitungszeit des Auftrags zu gewinnen.
- **Ausgabe der Zugehörigkeitswahrscheinlichkeit**
Das Verfahren zur Klassierung sollte neben der Ermittlung eines Clusters Informationen über die Konfidenz der Clusterzugehörigkeit in Form von Zugehörigkeitswahrscheinlichkeiten ausgeben.

2.3.2 Anforderungen an ein Verfahren zur multikriteriellen Ablaufplanung der Abfertigungsprozesse

Die folgenden Anforderungen ergeben sich für ein Verfahren zur multikriteriellen Ablaufplanung unter Unsicherheit für die Abfertigungsprozesse (Auf-, Abbau und Transferaufträge) in einem Luftfrachtterminal:

- **Verwendung des Klassifikators und der verfügbaren Informationen über die Bearbeitungszeitunsicherheiten**
Das Ablaufplanungsverfahren soll den Klassifikator nutzen, um für neu einzuplanende Aufträge Informationen über die Verteilung ihrer Bearbeitungszeiten bzw. die Bearbeitungszeitintervalle zu ermitteln, und diese entsprechend innerhalb des Verfahrens verwenden, um robuste Ablaufpläne zu erstellen.
- **Keine explizite Konzentration auf den Worst Case**
Das Verfahren sollte parametrierbar sein, sodass ein bestimmtes Anspruchsniveau durch den Entscheidungsträger vorgegeben werden kann, das von einem Ablaufplan unter jeder möglichen Realisierung der Bearbeitungszeiten der Aufträge eingehalten wird.
- **Berücksichtigung der dynamischen und stochastischen Umgebung**
Die sich aus der dynamischen und stochastischen Umgebung ergebenden Zielfunktionen der Stabilität und Robustheit sind durch das Verfahren in Kombination mit dem Performancemaß (Anzahl säumiger Aufträge) als multikriterielles Ablaufplanungsproblem zu betrachten und als Mehrzieloptimierungsproblem zu berücksichtigen.
- **Betrachtung des Personals als einschränkende Ressource**
Die beschränkt zur Verfügung stehende Ressource „Personal“ ist durch das Verfahren bei der Generierung von Ablaufplänen zu berücksichtigen. Hierbei sind zudem weitere Restriktionen, wie die unterschiedliche Qualifikation der Mitarbeiter und die benötigte Personalstärke, für einen Auftrag zu beachten.
- **Betrachtung der Transportsysteme als einschränkende Ressource**
Das Verfahren soll die durch den Ablaufplan erzeugte Belastung der Transportsysteme bereits bei der Planerstellung berücksichtigen, um somit einer Überlastsituation vorzubeugen.
- **Beherrschung der Komplexität**
Aufgrund des Einsatzes des Verfahrens für reale Problemgrößen und der Berücksichtigung der Unsicherheit ist die Ablaufplanung mit einer erheblichen Komplexität verbunden. Das Verfahren sollte daher innerhalb der zur Verfügung stehenden Zeit eine Lösung liefern können bzw. jeder Zeit unterbrechbar sein und den bis dahin besten Ablaufplan zurückliefern.

2.3.3 Anforderungen an ein Verfahren zur Ablaufsteuerung der Abfertigungsprozesse

Folgende Anforderungen können für ein Verfahren zur Revision von Ablaufplänen in einer dynamischen Umgebung genannt werden:

- **Berücksichtigung eines bereits aktiven Ablaufplans**
Das Verfahren zur Ablaufsteuerung hat beim Eintreten eines Ereignisses und der Revision des aktuell ausgeführten Ablaufplans zu berücksichtigen, dass bereits getroffene Entscheidungen dieses Plans möglichst wenig geändert werden sollten.
- **Schnelle Entscheidungsfindung**
Beim Eintreten eines Ereignisses sollte das Verfahren möglichst schnell einen neuen zulässigen Ablaufplan generieren.

3 Stand der Forschung

In den folgenden Abschnitten wird ein Überblick über den relevanten Stand der Forschung hinsichtlich der im vorherigen Kapitel identifizierten Problemfelder gegeben. Es werden bereits bekannte Arbeiten und Lösungsverfahren betrachtet und auf ihre Eignung hinsichtlich der definierten Anforderungen an eine Ablaufplanung und -steuerung für Luftfrachtterminals untersucht. Der Aufbau dieses Kapitels orientiert sich an den drei Problemfeldern. In Abschnitt 3.1 werden zunächst Verfahren vorgestellt, die den beschriebenen benötigten Informationsstand auf Basis historischer Daten ermitteln. Anschließend werden in Abschnitt 3.2 bestehende Ansätze zur Ablaufplanung unter Unsicherheit und in Abschnitt 3.3 zur Ablaufsteuerung und Revision von Ablaufplänen untersucht. Das Kapitel schließt in Abschnitt 3.4 mit einer Bewertung der vorgestellten Verfahren.

3.1 Datenanalyse und Wissensextraktion aus Datenquellen

Ausgehend von dem in Abschnitt 2.2.1 beschriebenen Problem der Datenanalyse der Abfertigungsprozesse wurden historische Betriebsdaten über Frachtabfertigungsaufträge als wichtige Informationsquelle identifiziert. Diese Informationen liegen jedoch häufig als Rohdaten in Datenbanken vor, die zur Prognose von Bearbeitungszeiten nicht direkt verwendet werden können. Daher ist die Datenbasis entsprechend aufzubereiten, um den benötigten Informationsstand über die unsicheren Bearbeitungszeiten der Frachtabfertigungsaufträge zu ermitteln. Dieser Informationsstand unterscheidet sich von der Datenbasis dahingehend, dass die Daten in einer aggregierten, abstrakten und für die spätere Ablaufplanung besser verwertbaren Form vorliegen.¹ Die zuvor genannten Eigenschaften werden in der Literatur im Bereich des *Knowledge Discovery in Databases* (KDD) behandelt, welches sich als interdisziplinäres Forschungsgebiet mit Schwerpunkten im Bereich der Statistik, des maschinellen Lernens und Datenbanksystemen versteht.²

Fayyad et al. definieren KDD als „the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.“³ Den KDD-Prozess bezeichnen die Autoren als interaktiven und iterativen Prozess und definieren ein Prozessmodell, das sich in neun Schritte unterteilt. Dieses reicht von der Entwicklung

¹Vgl. [FPS96a], S. 37.

²Vgl. z. B. [ES00], S. 1. Fayyad et al. führen zusätzlich die Forschungsgebiete „pattern recognition“, „artificial intelligence“, „knowledge acquisition for expert systems“ sowie „data visualization“ an (siehe [FPS96b], S. 4).

³[FPS96b], S. 6.

eines Domänenverständnisses und der Definition einer Zieldefinition bis zur Nutzung des erhaltenen Wissens.⁴

Neben dem von Fayyad et al. beschriebenen Prozessmodell existieren weitere Modelle, die in der Folge entstanden sind. Eine Übersicht und einen Vergleich dieser Modelle liefern Kurgan & Musilek.⁵ Die Autoren fassen den Begriff KDD weiter, indem sie ihn auf beliebige Datenquellen erweitern und somit Prozessmodelle für *Knowledge Discovery and Data Mining* (KDDM) beschreiben. Der Hauptunterschied der beschriebenen Prozessmodelle liegt in der Anzahl und dem Umfang durchzuführender Schritte sowie der Fokussierung auf einen industriellen oder akademischen Kontext.⁶

Die Bezeichnung KDDM weist bereits auf die zentrale Rolle hin, die das *Data Mining* (DM) in diesen Prozessmodellen einnimmt. Die Begriffe *Knowledge Discovery* und DM werden häufig synonym verwendet,⁷ dabei ist DM nach Fayyad et al.⁸ lediglich als ein Teilschritt im Gesamtprozess des KDDM bzw. KDD zu sehen. Die Autoren definieren daher DM als „a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data.“⁹

Als die zwei Hauptziele von DM-Algorithmen sehen Fayyad et al.¹⁰ die Prognose und die Beschreibung. Erstere wird dazu genutzt, um auf Basis von Eigenschaften der Daten zukünftige Werte von anderen Eigenschaften zu bestimmen. Die Beschreibung dient dazu, die aktuellen Daten in einer für Benutzer verständlichen Form aufzubereiten. Diese beiden Ziele werden durch die Anwendung verschiedener DM-Verfahren erreicht, die in verschiedene Verfahrensklassen unterteilt werden können.¹¹

In den folgenden Abschnitten liegt der Fokus auf dem siebten Schritt (die eigentliche Datenanalyse durch DM-Algorithmen) des Prozessmodells nach Fayyad et al. Des Weiteren erfolgt eine Einschränkung auf die beiden DM-Verfahrensklassen Clustering bzw. Klassenbildung sowie Klassifizierung. Erstere liefert Verfahren zur Behandlung der im Abschnitt 2.3.1 aufgezeigten Anforderung, Cluster ähnlicher Aufträge zu extrahieren. Letztere stellt Verfahren zur Klassierung zur Verfügung und adressiert somit die in Abschnitt 2.3.1 definierte Anforderung, neue Aufträge anhand ihrer Eigenschaften klassieren zu können. Den Abschluss bildet ein Literaturüberblick.

3.1.1 Verfahren für den Aufgabenbereich der Klassenbildung

In Abschnitt 2.2.1.2 wurde bereits aufgezeigt, dass historische Aufträge hinsichtlich ihrer Eigenschaften selten identisch sind. Hieraus wurde die Notwendigkeit abgeleitet, Cluster

⁴Vgl. [FPS96b], S. 9 ff. sowie [FPS96c], S. 30 f. Vergleiche zudem Ester & Sander [ES00], S. 2–6, sowie Sharafi [Sha13], S. 60–64, für eine umfangreichere Diskussion der einzelnen Schritte des Prozessmodells.

⁵Siehe [KM06].

⁶Siehe ebd., S. 5.

⁷Vgl. [KM06], S. 2; [Pia07], S. 100; [HKP12], S. 6.

⁸Vgl. [FPS96a], S. 39.

⁹Ebd., S. 41.

¹⁰Siehe [FPS96b], S. 12.

¹¹Siehe ebd., S. 13–16. Petersohn führt ähnliche DM-Verfahrensklassen an (siehe [Pet05], S. 25–34).

ähnlicher Aufträge zu bilden. Nachfolgend wird daher eine Übersicht über Klassenbildungs- bzw. Clusterverfahren gegeben.

Das Ziel der Klassenbildungsverfahren (auch: Clusteranalyse) ist die Partitionierung einer Menge $\mathcal{X} = \{x_1, \dots, x_n\}$ von Datenobjekten in eine endliche Menge von k Clustern $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$, wobei die einzelnen Cluster \mathcal{C}_i mit $1 \leq i \leq k$ aus den Daten bestimmt werden müssen. Die Menge \mathcal{C} ist eine Partition von \mathcal{X} für die die folgenden Eigenschaften gelten:¹²

$$\bigcup_{i=1}^k \mathcal{C}_i = \mathcal{C} = \mathcal{X} \quad (3.1a)$$

und

$$\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \quad (i = 1, \dots, k; j = 1, \dots, k; i \neq j) \quad (3.1b)$$

Die erste Eigenschaft beschreibt, dass jedes Datenobjekt einem der Cluster zugeteilt ist. Die zweite Eigenschaft gibt an, dass die Cluster disjunkt¹³ sind und somit kein Datenobjekt gleichzeitig mehreren Clustern angehört. Datenobjekte, die zu einem Cluster gehören, sind dabei einander besonders ähnlich, wohingegen sie zu Datenobjekten anderer Cluster besonders unähnlich sind.

Die Durchführung der Clusteranalyse gliedert sich nach Jain & Dubes¹⁴ in fünf Schritte, wobei die ersten drei verpflichtend und die letzten beiden optional sind:

1. Festlegen der Daten- bzw. Musterrepräsentation
2. Definition eines Proximitätsmaßes
3. Durchführung der Klassenbildung bzw. der Clusteranalyse
4. Abstraktion der Daten
5. Evaluation der Ausgabe des Clusteringverfahrens

Im ersten Schritt findet eine Auswahl von Eigenschaften (engl. *features*¹⁵) statt, die jedes Datenobjekt charakterisieren.¹⁶ Hierbei kann zwischen Eigenschaftsselektion und Eigenschaftsextraktion unterschieden werden.¹⁷ Erstere wählt aus den vorhandenen Eigenschaften diejenigen aus, die für die Clusteranalyse am geeignetsten erscheinen. Hingegen erzeugt die Eigenschaftsextraktion aus den vorhandenen Eigenschaften durch Transformation neue Eigenschaften. Jedes Datenobjekt x_j wird somit durch einen d -dimensionalen

¹²Vgl. [Pet05], S. 91; [KP96], S. 439; [JD88], S. 58.

¹³Klassenbildungsverfahren, bei denen ein Datenobjekt zu mehreren Clustern gehören kann (z. B. Fuzzy Clustering oder probabilistische modell-basierte Clusterverfahren), werden in dieser Arbeit nicht betrachtet. Vergleiche z. B. [KR90], S. 164–197; [HKKR99] oder [JMF99], S. 281 f., für Fuzzy Clustering und [FR02] oder [HKP12], S. 497–508 für probabilistische modell-basierte Clusterverfahren.

¹⁴Vgl. [JD88] und [JMF99], S. 266 ff. Vgl. ebenfalls [AF07], S. 1–4, sowie [Jai10], S. 656–660.

¹⁵Weitere gebräuchliche Begriffe sind *attribute*, *dimension* und *variable* (vgl. [HKP12], S. 40).

¹⁶Vgl. Schritt 4 des KDD-Prozesses.

¹⁷Siehe [AF07], S. 1 f.

Eigenschaftsvektor $\vec{e}_{x_j} = (e_{x_j,1}, \dots, e_{x_j,d})$ beschrieben, wobei $e_{x_j,p}$ mit $1 \leq p \leq d$ den Wert der p -ten Eigenschaft beschreibt. Die Menge der zu gruppierenden Datenobjekte \mathcal{X} wird somit durch die Eigenschaftsvektoren beschrieben. Diese Repräsentation wird von Jain & Dubes als „ $n \times d$ pattern matrix“¹⁸ bezeichnet. Jede Zeile dieser Matrix entspricht einer Musterrepräsentation eines Datenobjekts und jede Spalte der entsprechenden Eigenschaft.¹⁹ Die Eigenschaften können hierbei nach ihrem Typ sowie ihrer Skalierung charakterisiert werden.²⁰ Der Typ der Eigenschaft kann binär (zwei mögliche Ausprägungen), diskret (endliche oder abzählbar unendliche Menge möglicher Ausprägungen) oder stetig (überabzählbare Menge möglicher Ausprägungen) sein.²¹ Weiterhin können die Eigenschaften nominal, ordinal sowie kardinal (Intervall und Verhältnis) skaliert sein.²²

Für die Zuordnung von Datenobjekten zu Clustern muss die Ähnlichkeit bzw. Unähnlichkeit der Datenobjekte zueinander quantifiziert werden können. Dies geschieht über sogenannte Proximitätsmaße. Diese Maße können in Ähnlichkeitsmaße, die die Ähnlichkeit zweier Datenobjekte zueinander bestimmen, und Unähnlichkeitsmaße bzw. Distanzmaße unterteilt werden. Das zu verwendende Proximitätsmaß wird in Abhängigkeit der Typen und Skalierung der vorhandenen Eigenschaften ausgewählt. Bezeichne $\text{dist}(x_i, x_j)$ die Proximität zwischen den beiden Datenobjekten, die durch die Eigenschaftsvektoren \vec{e}_{x_i} und \vec{e}_{x_j} charakterisiert werden. Für das Maß gelten die folgenden Eigenschaften:²³

1. Identität:

a) Für ein Distanzmaß: $\text{dist}(x_i, x_i) = 0$ ($i = 1, \dots, n$)

b) Für ein Ähnlichkeitsmaß: $\text{dist}(x_i, x_i) \geq \max_l \text{dist}(x_i, x_l)$ ($i = 1, \dots, n$)

2. Symmetrie: $\text{dist}(x_i, x_j) = \text{dist}(x_j, x_i)$ ($i = 1, \dots, n; j = 1, \dots, n$)

3. Nichtnegativität: $\text{dist}(x_i, x_j) \geq 0$ ($i = 1, \dots, n; j = 1, \dots, n$)

Für stetige und kardinal²⁴ skalierte Eigenschaften wird häufig eine Minkowski-Metrik verwendet, die die Distanz zwischen zwei Datenobjekten berechnet. Diese Metrik ist folgendermaßen definiert:²⁵

$$\text{dist}(x_i, x_j) = \left(\sum_{p=1}^d |e_{x_i,p} - e_{x_j,p}|^r \right)^{\frac{1}{r}} \quad \text{mit } r \geq 1$$

¹⁸[JD88], S. 8.

¹⁹Siehe ebd.

²⁰Vgl. [And73], S. 26 f.; [JD88], S. 12.

²¹Vgl. [And73], S. 26 f.; [JD88], S. 12.

²²Vgl. [And73], S. 27; [AF07], S. 4.

²³Vgl. [JD88], S. 14; [Ye03], S. 290; [Lar05], S. 99; [HKP12], S. 72 f.

²⁴Kaufman & Rousseeuw beschreiben einen zweiten Ansatz, bei dem verhältnisskalierte Eigenschaften zunächst logarithmisch transformiert und anschließend als intervallskalierte Eigenschaften behandelt werden (vgl. [KR90], S. 32).

²⁵Vgl. [JD88], S. 14; [AF07], S. 6.

Die Minkowski-Metriken erfüllen zusätzlich zu den vorgenannten Eigenschaften die zwei folgenden:²⁶

1. Identität: $\text{dist}(x_i, x_j) = 0$ gdw. $e_{x_i,p} = e_{x_j,p}$ ($p = 1, \dots, d$)
2. Dreiecksungleichung: $\text{dist}(x_i, x_l) \leq \text{dist}(x_i, x_j) + \text{dist}(x_j, x_l)$ ($i, j, l = 1, \dots, n$)

In Abhängigkeit des Parameters r werden verschiedene Minkowski-Metriken definiert. Mit $r = 1$ wird die Manhattan-Distanz²⁷ mit $r = 2$ die euklidische Distanz und mit $r = \infty$ die Chebyshev-Distanz²⁸ berechnet.²⁹

Die Distanzberechnung für binäre³⁰ sowie nominal³¹ oder ordinal³² skalierte Eigenschaften erfolgt durch andere Proximitätsmaße als die zuvor beschriebenen. Für Datenobjekte, die durch verschiedene (hybrid) skalierte Eigenschaften charakterisiert werden, schlagen Han et al.³³ vor, diese in einer Distanzfunktion zu behandeln:³⁴

$$\text{dist}(x_i, x_j) = \frac{\sum_{p=1}^d \delta_{i,j}^{(p)} \text{dist}_{i,j}^{(p)}}{\sum_{p=1}^d \delta_{i,j}^{(p)}}$$

mit

$$\delta_{i,j}^{(p)} = \begin{cases} 0, & \text{wenn 1) } e_{x_i,p} = \text{null} \vee e_{x_j,p} = \text{null} \\ & \text{2) } e_{x_i,p} = e_{x_j,p} = 0 \text{ mit } p \text{ asymmetrisch binär} \\ 1 & \text{in allen anderen Fällen} \end{cases}$$

Der Anteil der Eigenschaft p an der Distanz wird über $\text{dist}_{i,j}^{(p)}$ beschrieben und abhängig von der Skalierung der Eigenschaft berechnet. Für ordinal skalierte Eigenschaften wird der Wert zunächst normiert und anschließend wird so verfahren, als sei dieser kardinal skaliert. Für kardinalskalierte Eigenschaften wird für $\text{dist}_{i,j}^{(p)}$ die Manhattan-Distanz verwendet, wobei eine Normierung auf das Intervall $[0, 1]$ stattfindet:

$$\text{dist}_{i,j}^{(p)} = \frac{|e_{x_i,p} - e_{x_j,p}|}{\max_l(e_{x_l,p}) - \min_l(e_{x_l,p})}$$

Schließlich werden binäre und nominal skalierte Eigenschaften folgendermaßen berechnet:

$$\text{dist}_{i,j}^{(p)} = \begin{cases} 0, & \text{wenn } e_{x_i,p} = e_{x_j,p} \\ 1 & \text{in allen anderen Fällen} \end{cases}$$

²⁶Vgl. [JD88], S. 14 f.

²⁷Auch als City-Block oder Taxifahrer-Distanz bezeichnet.

²⁸Wird ebenfalls als Supremum-Distanz, L_{\max} sowie L_{∞} -Norm bezeichnet (vgl. [HKP12], S. 73).

²⁹Vgl. [JD88], S. 15.

³⁰Vgl. [JD88], S. 16–19; [KR90], S. 22–27; [HKP12], S. 70 ff.

³¹Vgl. [KR90], S. 28 f.; [HKP12], S. 68 ff.

³²Vgl. [KR90], S. 29 ff.; [HKP12], S. 74 f.

³³[HKP12], S. 75 f.

³⁴Siehe ebenfalls [KR90], S. 32–37, sowie Gower [Gow71], auf den dieser Ansatz zurückzuführen ist. Gower betrachtet jedoch lediglich intervall- und nominalskalierte sowie Binärdaten, für die ein Ähnlichkeitsmaß entwickelt wird.

Der dritte Schritt – die eigentliche Durchführung der Klassenbildung – kann nach Petersohn³⁵ durch multivariate statistische Verfahren (Clusteranalyse) oder mittels künstlicher neuronaler Netze (Competitive Strategien)³⁶ erfolgen. Die multivariaten statistischen Verfahren lassen sich weiter in hierarchische sowie partitionierende Verfahren unterteilen,³⁷ auf die in den Abschnitten 3.1.1.1 und 3.1.1.2 näher eingegangen wird.

Im vierten Schritt erfolgt eine kompakte und vereinfachte Repräsentation der ermittelten Cluster und somit eine weitere Abstraktion der Daten. Diese kann einerseits auf eine einfache Verständlichkeit für Benutzer abzielen. Andererseits können die Cluster so aufbereitet werden, dass sie für eine automatische Verarbeitung besser geeignet sind. Eine Abstraktion eines Clusters kann bspw. in Form der Angabe eines Clusterzentrums³⁸ z_i erfolgen. Dieses entspricht je nach angewendetem Clusterverfahren z. B. dem Mittelwert aller in dem Cluster enthaltenen Datenobjekte und somit einem Vektor \vec{e}_{z_i} .³⁹

Im letzten Schritt der Clusteranalyse erfolgt eine Evaluation der Ausgabe des Clusteringverfahrens. Hier lässt sich eine Analyse der Ursprungsdaten auf die Tendenz hin, Cluster zu enthalten, von der Analyse des eigentlichen Outputs des Clusteringverfahrens unterscheiden.⁴⁰ Erstere soll die Anwendung von Clusterverfahren auf Daten, die keine Cluster enthalten, vermeiden.⁴¹ Letztere dient dazu, die Validität der ermittelten Cluster zu beurteilen. Hierbei unterscheiden Jain & Dubes⁴² drei Kriterien zur Beurteilung der Validität: *Externe Kriterien*, die auf einem Vergleich der gefundenen Struktur mit einer bereits bekannten Struktur beruhen. *Interne Kriterien* zur Beurteilung der Eignung der gefundenen Struktur für die vorhanden Daten sowie *relative Kriterien* zum Vergleich der Leistung zweier Strukturen.

3.1.1.1 Hierarchische Verfahren

Ein hierarchisches Clustering entspricht nach Jain & Dubes⁴³ einer Sequenz von Partitionen⁴⁴ \mathcal{C}_i für eine Menge von zu clusternden Datenobjekten \mathcal{X} , wobei jede Partition \mathcal{C}_i in die nächste Partition \mathcal{C}_j mit $1 \leq i < j \leq n$ der Sequenz geschachtelt ist. Eine Partition \mathcal{C}_i ist in die Partition \mathcal{C}_j geschachtelt, wenn die folgende Bedingung gilt:

$$\mathcal{C} \subset \mathcal{C}' \quad \forall \mathcal{C} \in \mathcal{C}_i \wedge \forall \mathcal{C}' \in \mathcal{C}_j$$

Die hierarchischen Clusterverfahren lassen sich weiterhin anhand der verwendeten Strategie in agglomerative (Bottom-up-Strategie) sowie divisive (Top-down-Strategie)

³⁵Siehe [Pet05], S. 26 f.

³⁶Die Klassenbildung mittels künstlicher neuronaler Netze wird hier nicht weiter betrachtet. Für eine ausführliche Auseinandersetzung siehe [Pet05], S. 73–91.

³⁷Han et al. führen weiter dichte-basierte (vgl. [HKP12], Kapitel 10.4) sowie rasterbasierte (vgl. ebd., Kapitel 10.5) Methoden an, auf die hier nicht näher eingegangen werden soll.

³⁸Häufig auch Clusterschwerpunkt, Clustermittelpunkt oder Zentroid (engl. *centroid*) genannt.

³⁹Vgl. [AF07], S. 3; [JMF99], S. 267.

⁴⁰Vgl. [AF07], S. 3 f.; [JMF99], S. 267 f.

⁴¹Zur Beurteilung der Clustertendenz sei auf [JD88], S. 201–222, verwiesen.

⁴²Siehe ebd., S. 161.

⁴³Siehe ebd., S. 58 f.

⁴⁴Vgl. Gleichung (3.1).

Clusterverfahren unterteilen.⁴⁵ Erstere gehen initial davon aus, dass jedes Datenobjekt aus \mathcal{X} ein eigenes Cluster bildet. Diese werden dann schrittweise zusammengefasst, bis alle Elemente Teil eines Clusters sind bzw. ein Abbruchkriterium erreicht worden ist. Divisive Verfahren führen diesen Prozess in umgekehrter Reihenfolge durch und starten mit einem Cluster, in dem alle Datenobjekte vorhanden sind, das schrittweise in kleinere Cluster aufgeteilt wird, bis jedes Cluster nur noch ein Objekt enthält bzw. ein Abbruchkriterium erreicht worden ist. Eine Visualisierung des Clusteringprozesses kann für eine begrenzte Anzahl an Datenobjekten in Form eines Dendrogramms erfolgen.⁴⁶

Für die Durchführung eines agglomerativen hierarchischen Clusterings wird ein Proximitätsmaß⁴⁷ zur Berechnung des Abstandes zwischen zwei Datenobjekten sowie ein Fusionierungsalgorithmus (engl. *linkage criteria/measures*) zur Bestimmung des Abstandes zwischen zwei Clustern benötigt.⁴⁸ Häufig verwendete Fusionierungsalgorithmen zur Berechnung des Abstandes zweier Cluster \mathcal{C}_i und \mathcal{C}_j sind:

- Single-linkage: $\text{dist}_{SL}(\mathcal{C}_i, \mathcal{C}_j) = \min_{\substack{x \in \mathcal{C}_i, \\ x' \in \mathcal{C}_j}} \text{dist}(x, x')$
- Complete-linkage: $\text{dist}_{CL}(\mathcal{C}_i, \mathcal{C}_j) = \max_{\substack{x \in \mathcal{C}_i, \\ x' \in \mathcal{C}_j}} \text{dist}(x, x')$
- Average-linkage: $\text{dist}_{AL}(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{|\mathcal{C}_i||\mathcal{C}_j|} \sum_{\substack{x \in \mathcal{C}_i, \\ x' \in \mathcal{C}_j}} \text{dist}(x, x')$
- Mean distance: $\text{dist}_{MD}(\mathcal{C}_i, \mathcal{C}_j) = \text{dist}(z_i, z_j)$, mit z_i dem Clusterzentrum von \mathcal{C}_i und z_j dem Clusterzentrum von \mathcal{C}_j

Fusioniert werden jeweils diejenigen zwei Cluster, die auf Basis der zuvor beschriebenen Distanzen den geringsten Abstand besitzen. Für die ersten beiden Fusionierungsalgorithmen stehen mit SLINK⁴⁹ für single-linkage und CLINK⁵⁰ für complete-linkage zwei optimal effiziente Verfahren mit der Komplexität $\mathcal{O}(n^2)$ zur Verfügung.

Im Gegensatz zu den agglomerativen Verfahren besitzen divisive Verfahren eine Komplexität von $\mathcal{O}(2^n)$, da es $2^{n-1} - 1$ Möglichkeiten gibt, ein Cluster mit n Objekten in zwei disjunkte Cluster zu zerlegen.⁵¹ Aus diesem Grund werden für divisive Verfahren Heuristiken verwendet. Kaufman & Rousseeuw⁵² beschreiben auf Basis des von Macnaughton-Smith et al.⁵³ vorgeschlagenen Algorithmus ein iteratives Verfahren. Dieses bestimmt in jeder Iteration das Cluster \mathcal{C}_i mit dem größten Durchmesser⁵⁴ und wählt

⁴⁵Vgl. [HTF09], S. 520; [Ber06], S. 27; [HMS01], S. 308.

⁴⁶Vgl. [JD88], S. 59.

⁴⁷Vgl. Abschnitt 3.1.1.

⁴⁸Vgl. [HKP12], S. 461; [HTF09], S. 523 f.; [GMW07], S. 116–137.

⁴⁹Siehe [Sib73].

⁵⁰Siehe [Def77].

⁵¹Siehe [HKP12], S. 461; [KR90], S. 253 f.

⁵²Siehe [KR90], Kapitel 6.

⁵³Siehe [MWD64].

⁵⁴Der Durchmesser entspricht der maximalen Distanz aller Objekte eines Clusters.

aus diesem ein Objekt aus, das die größte durchschnittliche Distanz zu allen anderen Objekten dieses Clusters besitzt. Dieses Objekt bildet das Zentrum eines neuen Clusters⁵⁵ \mathcal{C}_j . Objekte aus \mathcal{C}_i , deren durchschnittliche Distanz zu Objekten aus \mathcal{C}_j geringer ist als zu den verbliebenen Objekten in \mathcal{C}_i , werden dem neuen Cluster zugeordnet.

3.1.1.2 Partitionierende Verfahren

Im Vergleich zu den hierarchischen Verfahren wird durch partitionierende Verfahren lediglich eine Partition \mathcal{C} von Clustern für die zu clusternden Datenobjekte \mathcal{X} erstellt.⁵⁶ Jain & Dubes definieren diese Art der Clusteranalyse wie folgt:

„Given n patterns in a d -dimensional metric space, determine a partition of the patterns into K groups, or clusters, such that the patterns in a cluster are more similar to each other than to patterns in different clusters.“⁵⁷

Zur Durchführung einer partitionierenden Clusteranalyse wird ein Clusteringkriterium sowie häufig die Angabe der Anzahl k der zu bildenden Cluster a priori⁵⁸ benötigt. Zur Erreichung eines globalen Optimums ist eine vollständige Enumeration aller möglichen Partitionen notwendig. Dieses Vorgehen ist allerdings bereits für wenige Objekte aufgrund der sehr hohen Anzahl möglicher Partitionierungsmöglichkeiten nicht zielführend.⁵⁹ Aus diesem Grund werden häufig iterative Verfahren verwendet, die ausgehend von einer initialen Partitionierung Objekte zwischen den k Clustern austauschen, um den Wert des gewählten Clusteringkriteriums bis zu einem lokalen Optimum hin zu verbessern.

Die Clusteringkriterien lassen sich nach Jain & Dubes⁶⁰ in globale sowie lokale Kriterien unterteilen. Erstere verwenden einen Prototypen (z. B. das Clusterzentrum), um das Cluster zu repräsentieren. Zu dieser Gruppe gehören Kriterien, die Varianten der (quadratischen) Abweichung/Fehler darstellen. Auch das Kriterium, das eine Mischverteilung an die Objekte aus \mathcal{X} anpasst, gehört zu dieser Gruppe. Lokale Kriterien hingegen verwenden lokale Strukturen der Daten, um Cluster zu erzeugen. Hierzu gehören z. B. dichte-basierte bzw. mode-seeking, graphenbasierte sowie Nächste-Nachbarn-Kriterien. Im Folgenden wird näher auf zwei Verfahren – k-means und k-medoids – eingegangen, die die (quadratische) Abweichung als Kriterium verwenden.

Das k-means-Verfahren⁶¹ verwendet als Kriterium die Summe der quadratischen Abweichungen, um die Menge \mathcal{X} in k Cluster zu partitionieren. Jedes Cluster \mathcal{C}_i wird

⁵⁵Die Autoren Kaufman & Rousseeuw sprechen hier von „splinter group“ [KR90], S. 255.

⁵⁶Vgl. [JD88], S. 89.

⁵⁷Ebd., S. 90.

⁵⁸Zur Diskussion, wie dieser Parameter bestimmt werden kann, siehe u. a. [HKP12], S. 486 f., und [Rok10], S. 292–295.

⁵⁹Vgl. [JD88], S. 90 f.

⁶⁰Vgl. ebd., S. 90 ff.

⁶¹Der Standardalgorithmus geht zurück auf [Llo57] (im Jahr 1982 schließlich als Artikel [Llo82] veröffentlicht), wobei erst [Mac67] den Begriff *k-means* verwendet, dessen Idee auf [Ste56] zurückgeht (vgl. [Boc08]).

hierbei durch ein Clusterzentrum z_i , das als Prototyp gilt, repräsentiert. Dieses ergibt sich aus dem Mittelwert aller Datenobjekte dieses Clusters:

$$e_{z_i,p} = \frac{\sum_{x \in \mathcal{C}_i} e_{x,p}}{|\mathcal{C}_i|} \quad (p = 1, \dots, d) \quad (3.2)$$

Die Qualität der Partitionierung wird über die Summe der quadrierten Abweichungen zwischen allen Objekten eines Clusters und dem Clusterzentrum berechnet.⁶²

$$SSE = \sum_{\mathcal{C}_i \in \mathcal{C}} \sum_{x \in \mathcal{C}_i} \text{dist}(x, z_i)^2$$

Die Optimierung dieses Kriteriums ist NP-schwer, daher verwendet k-means ein heuristisches Vorgehen zur Bestimmung der Partitionierung:⁶³

1. Zufällige Auswahl von k Clusterzentren (dies können entweder tatsächliche Elemente aus \mathcal{X} sein oder künstlich erzeugte)
2. Zuweisung aller Datenobjekte zu dem Cluster, zu dessen Clusterzentrum die geringste euklidische Distanz besteht
3. Aktualisierung der Clusterzentren, indem Gleichung (3.2) unter Berücksichtigung der aktuellen Zuordnungen erneut berechnet wird
4. Wiederholen der Schritte 2 und 3 bis ein Abbruchkriterium (z. B. stabile Zuordnung, bei der die aktuelle Zuordnung die gleiche wie in der vorherigen Iteration ist) erreicht ist

Das k-means-Verfahren kann ohne Modifikation nur für Eigenschaften verwendet werden, bei denen der Mittelwert definiert ist. Dies ist z. B. für nominale oder binäre Eigenschaften nicht der Fall.⁶⁴ Darüber hinaus ist das Verfahren anfällig für Ausreißer oder Rauschen in den Datenobjekten, da diese einen hohen Einfluss auf den Mittelwert haben können und somit das Auffinden natürlicher Cluster erschweren.⁶⁵

Eine Alternative stellt das k-medoids-Verfahren dar, das robuster im Falle von Ausreißern und Rauschen als k-means ist. Bei diesem wird als Repräsentant für ein Cluster nicht wie bei k-means der Mittelwert aller in dem Cluster vorhandenen Datenobjekte verwendet, sondern ein in dem Cluster vorhandenes Datenobjekt $z_i \in \mathcal{X}$. Die Qualität einer Partitionierung wird hier über die absolute Abweichung aller in einem Cluster vorhandenen Datenobjekte und dem repräsentativen Objekt z_i für dieses Cluster berechnet:⁶⁶

$$SAE = \sum_{\mathcal{C}_i \in \mathcal{C}} \sum_{x \in \mathcal{C}_i} \text{dist}(x, z_i) \quad (3.3)$$

⁶²Vgl. [HKP12], S. 451.

⁶³Vgl. [HKP12], S. 452; [Rok10], S. 281.

⁶⁴Eine Alternative sind die Verfahren k-modes (für nominal skalierte Eigenschaften) und k-prototypes (für hybrid skalierte Eigenschaften); [Hua98].

⁶⁵Vgl. [HKP12], S. 453 f.

⁶⁶Vgl. ebd., S. 455.

Für $k \neq 1$ ist k-medoids NP-schwer und daher ist hier ebenfalls ein heuristisches Vorgehen notwendig. Ein solches Verfahren ist *Partitioning Around Medoids* (PAM) von Kaufman & Rousseeuw^{67,68}:

1. Zufällige Auswahl von k Datenobjekten aus \mathcal{X} als initiale Clusterzentren bzw. repräsentative Objekte
2. Zuweisung der restlichen Datenobjekte zu dem Cluster, zu dessen repräsentativen Objekt die geringste Distanz besteht
3. Zufällige Auswahl eines nicht repräsentativen Objekts und Berechnung der Kosten auf Basis von Gleichung (3.3) für eine Vertauschung mit einem repräsentativen Objekt
4. Sind die Kosten negativ führe die Vertauschung durch
5. Wiederholen der Schritte 2–4 bis keine Veränderung mehr eintritt

Die Berechnung durch PAM ist aufwendiger als für k-means. Für umfangreiche Datenmengen wird daher von Kaufman & Rousseeuw⁶⁹ mit *Clustering Large Applications* (CLARA)⁷⁰ ein Verfahren vorgeschlagen, das die Heuristik PAM nur auf einen repräsentativen Teil der zu clusternden Datenobjekte \mathcal{X} anwendet. Anschließend werden die verbliebenen Datenobjekte dem Cluster zugeordnet, zu dessen repräsentativen Objekt die geringste Distanz besteht. Ein weiteres Verfahren für k-medoids Clustering stellen die Autoren Park & Jun⁷¹ vor, das auch für große Datenmengen effizient ist.

3.1.2 Verfahren für den Aufgabenbereich der Klassifizierung

Zur Durchführung einer adäquaten Abschätzung der Bearbeitungszeit eines neu einzuplanenden Auftrags wurde in Abschnitt 2.2.1.2 aufgezeigt, dass die Zuordnung bzw. Klassierung dieser Aufträge zu Clustern ähnlicher Aufträge (vgl. Abschnitt 3.1.1) zielführend ist. Im folgenden Abschnitt erfolgt daher eine Übersicht zu Klassifikationsverfahren.

Im Unterschied zu den Klassenbildungs- bzw. Clusterverfahren besteht die Aufgabe der Klassierung darin, ein Datenobjekt x , das durch einen Eigenschaftsvektor \vec{e}_x charakterisiert wird und dessen Clusterzugehörigkeit unbekannt ist, zu einer von k im Vorhinein⁷² bekannten Clustern C_i mit $1 \leq i \leq k$ zuzuordnen. Zur Durchführung der Klassierung ist es notwendig, einen Klassifikator zu trainieren. Dies geschieht im Rahmen

⁶⁷Siehe [KR87].

⁶⁸Vgl. [HKP12], S. 457.

⁶⁹Siehe [KR86]; [KR90], S. 126–163.

⁷⁰Die Verfahren *Clustering Large Applications based upon Randomized Search* (CLARANS) [NH94] (erweitert in 2002 [NH02]) sowie hierauf aufbauend das Verfahren von Ester et al. [EKX95] sind Erweiterungen von CLARA.

⁷¹Siehe [PJ09].

⁷²Aus diesem Grund spricht man auch von überwachtem Lernen, da die Clusterzugehörigkeit der Trainingsdaten bekannt ist im Gegensatz zu den Klassenbildungs- bzw. Clusterverfahren, bei denen es sich um unüberwachtes Lernen handelt, da die Cluster sowie evtl. die Clusteranzahl unbekannt sind (vgl. [HKP12], S. 330).

der Klassifizierung,⁷³ bei der die Klassengrenzen zwischen den Clustern erlernt werden. Hierfür wird der Klassifikator zunächst mit einer Menge von n Trainingsdaten $\mathcal{T} = \{(\vec{e}_{x_1}, \mathcal{C}_1), \dots, (\vec{e}_{x_n}, \mathcal{C}_n)\}$ trainiert, bei denen die Clusterzugehörigkeit der Datenobjekte x_j mit $1 \leq j \leq n$ bekannt ist. Die Trainingsdaten bestehen aus Tupeln von Eingabe-Ausgabe-Paaren, wobei der Eingabe \vec{e}_{x_j} eines der bekannten Cluster \mathcal{C}_i mit $1 \leq i \leq k$ in Form eines sogenannten Klassenlabels zugeordnet wird.⁷⁴ Mit Hilfe der Trainingsdaten wird durch ein Klassifikationsverfahren eine Zuordnung bzw. Funktion $\mathcal{C}_i = f(\vec{e}_x)$ erlernt, die ein Datenobjekt x mit unbekannter Clusterzugehörigkeit auf ein Cluster \mathcal{C}_i abbildet.⁷⁵ Die konkrete Repräsentation einer solchen Zuordnung wird während der Klassifizierung festgelegt und hängt von dem verwendeten Klassifikationsverfahren ab.⁷⁶

Kotsiantis⁷⁷ unterscheidet bei den Klassifikationsverfahren zwischen logikbasierten (Entscheidungsbäume, Klassifikationsregeln), perceptronbasierten (künstliche neuronale Netze), statistischen (naive Bayes-Klassifikatoren, Bayes-Netzwerke), instanzbasierten (k-nächste-Nachbarn) sowie Support Vector Machines.⁷⁸ Aufgrund der in Abschnitt 2.3.1 definierten Anforderung nach der Ausgabe von Zugehörigkeitswahrscheinlichkeiten durch den Klassifikator werden in den folgenden beiden Abschnitten probabilistische Klassifikatoren betrachtet, die in der Lage sind, bei der Klassierung eines Objekts x eine Verteilung über die mögliche Klassenzugehörigkeit und somit einen Grad für die Sicherheit der Klassierung auszugeben. Es erfolgt eine Unterscheidung in Verfahren, die auf einem generativen Modell basieren (Abschnitt 3.1.2.1), und solchen, die ein diskriminatives Modell (Abschnitt 3.1.2.2) verwenden.⁷⁹

3.1.2.1 Generative Modelle zur Klassifizierung

Bei der Klassierung eines Datenobjekts x (charakterisiert durch den Eigenschaftsvektor \vec{e}_x), dessen Clusterzugehörigkeit unbekannt ist, zu einem von k Clustern besteht die Aufgabe aus probabilistischer Sicht darin, die bedingte Wahrscheinlichkeit (auch: A-posteriori-Wahrscheinlichkeit) $P(Y | X)$ zu bestimmen. Hierbei beschreibt X einen Vektor $X = (X_1, \dots, X_d)$ von Zufallsvariablen, wobei X_p mit $1 \leq p \leq d$ die Zufallsvariable der Eigenschaft p ist, dessen konkreter Wert mit $e_{x,p}$ bezeichnet wird. Mit Y wird die Zufallsvariable des Clusters, die die Werte \mathcal{C}_i mit $1 \leq i \leq k$ annehmen kann, bezeichnet. Generative Modelle modellieren die gemeinsame Verteilung $p(X, Y)$ und

⁷³Han et al. (siehe [HKP12], S. 328) fassen unter dem Begriff „classification“ sowohl die Klassierung als auch die Klassifizierung zusammen. Letztere wird als „learning step“ (auch: „training phase“) und erstere als „classification step“ eines zweischrittigen bzw. in zwei Phasen durchzuführenden Prozesses zur Klassifizierung bezeichnet.

⁷⁴Vgl. [Bis06], S. 179 f.; [TSK06], S. 146.

⁷⁵Vgl. [HKP12], S. 330; [TSK06], S. 146.

⁷⁶Vgl. [Bis06], S. 2.

⁷⁷Siehe [Kot07].

⁷⁸Ähnliche Unterteilungen sowie Erläuterungen der entsprechenden Verfahren finden sich bei [Pet05], S. 31 ff.; [TSK06], S. 150–172, 207–315, sowie [HKP12], S. 327–442.

⁷⁹Studien, ob generative oder diskriminative Modelle besser geeignet sind, finden sich z. B. in [NJ02] und [XT08]. Bishop & Lasserre schlagen in [BL07] eine Kombination beider Modelle vor.

bestimmen mit Hilfe der Summen- und Produktregel die bedingte Wahrscheinlichkeit (auch: likelihood) $P(X | Y)$ sowie die A-priori-Wahrscheinlichkeit $P(Y)$. Aus diesen beiden Wahrscheinlichkeiten kann mit Hilfe des Satzes von Bayes (auch: Bayes-Theorem) die A-posteriori-Wahrscheinlichkeit berechnet werden:

$$P(Y | X) = \frac{P(X | Y) P(Y)}{P(X)} \quad (3.4a)$$

mit

$$P(X) = \sum_{i=1}^k P(Y = \mathcal{C}_i) P(X | Y = \mathcal{C}_i) \quad (3.4b)$$

Ein Verfahren, das auf dem oben genannten generativen Modell aufbaut, ist der naive Bayes-Klassifikator. Bei diesem wird von der vereinfachenden Annahme ausgegangen, dass bei der Bestimmung von $P(X | Y = \mathcal{C}_i)$ die Eigenschaftswerte X_p gegeben Y , bedingt unabhängig voneinander sind:

$$\begin{aligned} P(X | Y) &= \prod_{p=1}^d P(X_p = e_{x,p} | Y = \mathcal{C}_i) \\ &= P(X_1 = e_{x,1} | Y = \mathcal{C}_i) \times \cdots \times P(X_d = e_{x,d} | Y = \mathcal{C}_i) \end{aligned}$$

Der Klassifikator arbeitet anschließend in zwei Phasen: In der Inferenzphase werden die einzelnen Wahrscheinlichkeiten $P(X | Y = \mathcal{C}_i)$ aus den Trainingsdaten \mathcal{T} bestimmt. Im Falle diskreter Eigenschaften ist dies durch Abzählen möglich, wohingegen bei kontinuierlichen Eigenschaften häufig von einer Normalverteilung ausgegangen wird, deren Erwartungswert und Varianz aus den Trainingsdaten geschätzt werden.⁸⁰ Die A-priori-Wahrscheinlichkeiten $P(Y)$ können ebenfalls aus den Trainingsdaten durch abzählen geschätzt werden. Anschließend kann mit Hilfe des Satzes von Bayes (vgl. Gleichung (3.4)) die A-posteriori-Wahrscheinlichkeiten für jedes Cluster berechnet werden. In der Entscheidungsphase wird schließlich das Cluster mit der höchsten A-posteriori-Wahrscheinlichkeit unter der Bedingung das x gegeben ist ausgewählt.⁸¹

$$\arg \max_{\mathcal{C}_i \in \mathcal{C}} (P(Y = \mathcal{C}_i | X))$$

Wird die Annahme der bedingten Unabhängigkeit für die Eigenschaftswerte nicht erfüllt, so können bestehende Abhängigkeiten zwischen den Eigenschaften durch die Erstellung eines Bayesschen Netzes (engl. *Bayesian Belief Network*) modelliert werden. Bayes-Netzwerke ermöglichen es, diese Abhängigkeiten durch eine systematische Modellierung mittels eines gerichteten azyklischen Graphen (engl. *directed acyclic graph*) sowie bedingten Wahrscheinlichkeitstabellen (engl. *Conditional Probability Table* (CPT)) darzustellen.⁸² Ein Training des Netzes besteht zunächst darin, die Topologie festzulegen

⁸⁰Vgl. [HKP12], S. 352.

⁸¹Alternative Auswahlverfahren für die Entscheidungsphase finden sich in [Bis06], S. 39–42.

⁸²Vgl. [HKP12], S. 393–397; [RN10], S. 510–513; [JN07], S. 32–35.

(z. B. durch einen Domänenexperten) bzw. zu erlernen. Ist die Struktur gegeben, müssen die Werte der CPTs erlernt werden. Obwohl dieses Vorgehen eine genauere Modellierung der Abhängigkeiten zwischen den Eigenschaften erlaubt, wird aufgrund der höheren Komplexität dieses Ansatzes stattdessen häufig der naive Bayes-Klassifikator verwendet, der hinsichtlich Einfachheit, Trainingsdauer, Geschwindigkeit der Klassierungen sowie der inkrementellen Einbeziehung neuer Trainingsdaten Vorteile besitzt.⁸³ Des Weiteren zeigen empirische Studien von Domingos & Pazzani,⁸⁴ Rish,⁸⁵ Hand & Yu⁸⁶ und Zhang,⁸⁷ dass der naive Bayes-Klassifikator, auch wenn die Annahme der bedingten Unabhängigkeit nicht zutrifft, dennoch geeignet ist und in der Praxis gute Ergebnisse liefern kann.⁸⁸ Studien von Caruana & Niculescu-Mizil⁸⁹ zeigen hingegen, dass moderne Verfahren wie z. B. *calibrated boosted trees* (vgl. den nachfolgenden Abschnitt 3.1.2.2) oder *calibrated random forests* zumindest für binäre Klassifikationsprobleme (bei der nur zwei Cluster zur Auswahl stehen) im Vergleich zu dem naiven Bayes-Klassifikator bessere Ergebnisse erzielen können.

3.1.2.2 Diskriminative Modelle zur Klassifizierung

Im Unterschied zu den im vorherigen Abschnitt beschriebenen generativen Modellen werden bei probabilistisch-diskriminativen Modellen die bedingten bzw. A-posteriori-Wahrscheinlichkeiten $P(Y | X)$ direkt berechnet. Verfahren, die auf diesem Modell aufbauen, sind u. a. die multinomiale logistische Regression, künstliche neuronale Netze sowie *calibrated boosted trees*, die im Folgenden kurz dargestellt werden.

Bei der (multinomialen) logistischen Regression wird von einem parametrischen Modell ausgegangen, das auf einer logistischen Funktion (der sogenannten Sigmoidfunktion) beruht und dessen Parameter direkt aus den Trainingsdaten ermittelt werden. Die A-posteriori-Wahrscheinlichkeit $P(Y | X)$ wird somit für $Y = \mathcal{C}_i$ mit $1 \leq i \leq (k - 1)$ modelliert als:⁹⁰

$$P(Y = \mathcal{C}_i | X) = \frac{\exp(w_{i,0} + \sum_{p=1}^d w_{i,p} X_p)}{1 + \sum_{j=1}^{k-1} \exp(w_{j,0} + \sum_{p=1}^d w_{j,p} X_p)} \quad (i = 1, \dots, (k - 1))$$

und für $Y = \mathcal{C}_k$ als:

$$P(Y = \mathcal{C}_k | X) = \frac{1}{1 + \sum_{j=1}^{k-1} \exp(w_{j,0} + \sum_{p=1}^d w_{j,p} X_p)}$$

⁸³Vgl. [Kot07], S. 262 ff.; [KZP06], S. 174; [DP97], S. 127.

⁸⁴[DP97]; [DP96].

⁸⁵[Ris01].

⁸⁶[HY01].

⁸⁷[Zha05]; [Zha04].

⁸⁸Vgl. zudem [HTF09], S. 211.

⁸⁹Siehe [CN06]. Siehe ebenfalls die Studie von Caruana et al. [CKY08] für eine empirische Evaluation überwachter Lernverfahren für Datenobjekte mit vielen Dimensionen (Eigenschaften).

⁹⁰Vgl. [Mit05], S. 13.

Der unbekannte Parametervektor $\vec{w} = (w_0, w_1, \dots, w_d)$ kann durch die Maximum-Likelihood-Methode geschätzt werden.⁹¹ Die Berechnung erfolgt dann bspw. durch iterative Verfahren wie *Iterative reweighted least squares*.⁹²

Künstliche neuronale Netze (KNN) bestehen aus einer Menge \mathcal{NE} von in einem Netzwerk verbundenen Knoten, die Neuronen simulieren. Die einzelnen Neuronen sind häufig in verschiedene Schichten (engl. *Layer*) unterteilt. Im Falle mehrschichtiger Netze sind neben der Eingabeschicht \mathcal{NE}^{In} (die die Eingabe der Eigenschaften \vec{e}_x eines Trainingsdatums x aufnimmt) und der Ausgabeschicht $\mathcal{NE}^{\text{Out}}$ (die der Ausgabe der Klassierung dient) eine oder mehrere verdeckte (engl. *hidden*) Schichten \mathcal{NE}^{H} vorhanden. Jedes Neuron $j \in (\mathcal{NE}^{\text{H}} \cup \mathcal{NE}^{\text{Out}})$ einer verdeckten Schicht oder der Ausgabeschicht besitzt eine Menge von Eingangsverbindungen von anderen Neuronen $i \in (\mathcal{NE}^{\text{In}} \cup \mathcal{NE}^{\text{H}})$, die mit einem Gewicht $w_{i,j}$ versehen sind. Aus den Eingangsdaten y_i wird die gewichtete Summe (engl. *net input*) $ni_j = \sum_{i=0}^n w_{i,j} y_i$ berechnet. Über eine Aktivierungsfunktion g und einen Schwellenwert (engl. *bias*) θ (verwendet als Gewicht $w_{0,j} = -\theta$ für $y_0 = 1$) wird die Ausgabe des Neurons j mit $y_j = g(ni_j)$ ermittelt.⁹³ Diese Aktivierungsfunktion kann dabei u. a. ein harter Grenzwert (Heaviside-Funktion) oder eine logistische Funktion sein. Für die Ausgabe von A-posteriori-Wahrscheinlichkeiten bietet sich die Verwendung einer softmax-Aktivierungsfunktion für die Neuronen der Ausgabeschicht $i \in \mathcal{NE}^{\text{Out}}$ an.⁹⁴

$$y_i = \frac{e^{ni_i}}{\sum_{j \in \mathcal{NE}^{\text{Out}}} e^{ni_j}},$$

womit die Ausgabewerte y_i dieser Neuronen im Intervall $[0, 1]$ liegen und sich zu eins aufsummieren. Sind sowohl die Netzwerkstruktur als auch die Aktivierungsfunktionen gegeben, werden durch Trainingsdaten die Gewichte $w_{i,j}$ gelernt. Die nicht-linearen Aktivierungsfunktionen der Neuronen stellen dabei sicher, dass das KNN nicht-lineare Funktionen darstellen kann und somit Klassifizierungen für nicht linear separierbare⁹⁵ Probleme vornehmen kann.⁹⁶ Der Nachteil bei KNNs liegt in der Bestimmung der richtigen Anzahl von Neuronen und der Interpretierbarkeit der Ausgabe.⁹⁷

Calibrated Boosted Trees (CBT) gehören zu den Ensemble-Techniken, die mehrere (schwache) Basisklassifikatoren (in diesem Fall Entscheidungsbäume) zu einem (starken) Klassifikator kombinieren, um somit ein besseres Klassifikationsmodell zu erzeugen.⁹⁸ Entscheidungsbäume (auch: Klassifikationsbäume), die bei CBT als Basisklassifikator Verwendung finden, bestehen aus einer Menge von Knoten, die durch gerichtete Kanten miteinander verbunden sind und eine Baumstruktur bilden. Jeder innere Knoten

⁹¹Vgl. [Bis06], S. 209; [Mit05], S. 13.

⁹²Vgl. [Bis06], S. 207–210.

⁹³Vgl. [RN10], S. 728 f.

⁹⁴Vgl. [Bri90], S. 233.

⁹⁵Linear nicht separierbare Probleme sind solche, bei denen die Cluster der Trainingsdaten nicht durch lineare Entscheidungsgrenzen (Entscheidungsflächen bei höheren Dimensionen) getrennt werden können (vgl. [Bis06], S. 179).

⁹⁶Vgl. [RN10], S. 730 ff.; [KZP06], S. 169 ff.

⁹⁷Vgl. [HKP12], S. 406 ff.; [KZP06], S. 169 ff.

⁹⁸Vgl. [HKP12], S. 378 f.

entspricht hierbei einem Test für eine Eigenschaft, jede Kante dem Ergebnis des Tests und jeder Blattknoten entspricht einem Klassenlabel bzw. dem klassierten Cluster.⁹⁹ In den Blättern kann zusätzlich die bedingte Wahrscheinlichkeit $P(Y | X)$ gespeichert werden, die die empirische Verteilung über die Klassenlabel der Trainingsdaten beschreibt, die den Blattknoten erreicht haben. Zur Erstellung der Entscheidungsbäume kommen Entscheidungsbaumverfahren, wie z. B. ID3¹⁰⁰, dessen Nachfolger C4.5¹⁰¹ oder CART¹⁰² zum Einsatz. Beim Boosting¹⁰³ werden während der Trainingsphase k Basisklassifikatoren Bk_i mit $1 \leq i \leq k$ iterativ erstellt. Nach der Erstellung des Klassifikators Bk_i erhält jedes Trainingsdatum der Trainingsmenge \mathcal{T} ein Gewicht bzw. dieses wird aktualisiert. Für Trainingsdaten, die fehlerhaft klassiert wurden, wird das Gewicht vergrößert und für korrekt klassierte verringert. Hierdurch wird der Klassifikator Bk_{i+1} der nächsten Iteration die fehlerhaft klassierten Trainingsdaten stärker berücksichtigen.¹⁰⁴ Für die Klassierung eines Datenobjekts x mit unbekannter Clusterzugehörigkeit durchläuft dieses jeden Entscheidungsbaum Bk_i von der Wurzel beginnend bis zu einem Blatt, welches das Klassenlabel für x bzw. die bedingte Wahrscheinlichkeit $P_{Bk_i}(Y | X)$ und somit die Ausgabe des Basisklassifikators angibt. Die Klassierungen der einzelnen Basisklassifikatoren werden schließlich durch gewichtete Mittelwertbildung kombiniert. Die Gewichte basieren auf der Fehlerrate des entsprechenden Klassifikators in der Vergangenheit und sind umso höher, je geringer die Fehlerrate war. Die ermittelten bedingten Wahrscheinlichkeiten sind nach Niculescu-Mizil & Caruana¹⁰⁵ verzerrt, da sie von den Extremwerten 0 und 1 weg bewegt werden. Aus diesem Grund werden häufig Wahrscheinlichkeitskalibrierungen durchgeführt, die im Falle binärer Klassifikationsprobleme z. B. mit Hilfe von Platt Scaling¹⁰⁶ oder Isotonic Regression¹⁰⁷ und für mehr als zwei Klassen bspw. mit einer Dirichlet-Kalibrierung¹⁰⁸ erreicht werden können.

3.1.3 Literaturüberblick

Arbeiten, die explizit historische Betriebsdaten nutzen, um einen Informationsstand über mögliche Unsicherheiten aufzubauen, der bei dem operativen Planungsproblem der Ablaufplanung in Luftfrachtterminals Verwendung findet, sind nicht bekannt. Daher wird im folgenden Abschnitt eine Übersicht zu Beiträgen gegeben, die im Bereich von Luftfrachtterminals implizit (historische) Betriebsdaten verwenden, die als Grundlage für Planungsprobleme innerhalb eines Terminals dienen. Literatur über interne Prozesse in Luftfrachtterminals und deren Optimierung ist allerdings spärlich, daher werden im Folgenden auch Arbeiten aus verwandten Bereichen berücksichtigt.

⁹⁹Vgl. [RM08], S. 8 f.

¹⁰⁰[Qui86].

¹⁰¹[Qui93].

¹⁰²[BFOS84].

¹⁰³Vgl. [SF12].

¹⁰⁴Vgl. [HKP12], S. 380 f.

¹⁰⁵[NC05b]; [NC05a].

¹⁰⁶[Pla00].

¹⁰⁷[ZE02].

¹⁰⁸[Geb09].

Es sind mehrere Beiträge vorhanden, die sich mit der Planung des Personals für die Frachtbearbeitung innerhalb der Luftfrachtterminals auseinandersetzen. Nobert & Roy¹⁰⁹ formulieren ein integriertes Planungsmodell für die Personalplanung. Dieses berücksichtigt simultan den Ausgleich von Lastspitzen¹¹⁰ im Frachtaufkommen (unter Berücksichtigung der für die Fracht geltenden Termine) und den sich hieraus ergebenden Personalbedarf sowie die auf Basis des Personalbedarfs zu erstellenden Schichtpläne. Die Schichtpläne werden hierbei mit einem Horizont von einem Tag erstellt. Die benötigten Parameter für das Optimierungsmodell werden durch Schätzungen gewonnen. Unter der Annahme, dass der Personalbedarf bekannt ist, erstellen Yan et al.¹¹¹ ganzzahlige bzw. gemischt ganzzahlige lineare Programme zur Erzeugung von Schichtplänen für einen kurzen Horizont (eine Woche) bzw. für einen kurzen Horizont (ebenfalls eine Woche) mit stochastischem Personalbedarf.¹¹² Rong & Grunow¹¹³ erweitern das Simultanplanungsmodell von Nobert & Roy, indem sie Qualifikationshierarchien zwischen dem Auf- und Abbaupersonal berücksichtigen sowie eine Unterscheidung beim „demand leveling“ zwischen Auf- und Abbauaufträgen vornehmen. Des Weiteren wird berücksichtigt, dass das Planungsproblem zyklisch ist und die erstellten Schichtpläne täglich ausgeführt werden müssen. Die für das Optimierungsmodell benötigten Parameter basieren auf Schätzungen.

Weitere Arbeiten nutzen Simulation zur Evaluierung und Verbesserung interner Terminalprozesse. Nsakanda et al.¹¹⁴ stellen einen Simulationsansatz zur Evaluierung der Frachtabfertigungsprozesse und der Auswirkungen bei Änderung der Verfahrens- und Bearbeitungsvorschriften innerhalb eines Luftfrachtterminals vor. Als Datengrundlage nutzen sie historische Betriebsdaten, Ergebnisse aus Interviews mit Domainexperten, Schätzungen sowie Zeitstudien. Lee et al.¹¹⁵ nutzen zeitbehaftete gefärbte Petri-Netze (engl. *timed colored petri nets*), um die terminal internen Prozesse zu modellieren. Das entwickelte Simulationsmodell wird anschließend zur Evaluierung der Zuweisungsstrategien der Airlines sowie der Lagerstrategie eines Regalbediengerätes genutzt. Der vorgestellte Ansatz basiert hierbei vollständig auf durchgeführten Zeitstudien.

Assent et al.¹¹⁶ beziehen sich in ihrer Arbeit nicht auf interne Prozesse in Luftfrachtterminals, sondern betrachten Verspätungen bei der Ankunft von Flügen. Die Autoren beschreiben ein Verfahren zur Klassierung von eingehenden Flügen auf Basis von Eigenschaften der Flugdaten in drei Verspätungskategorien, die als „ahead of time“, „on time“ und „delayed“ bezeichnet werden. Die Klassierung der Flüge soll der Unterstützung einer proaktiven bzw. robusten Ablaufplanung des Flughafenbetriebs dienen. Als Datenbasis zum Training des Klassifikators werden historische Betriebsdaten sowie die von Ablaufplanungsexperten bereitgestellten Klassenlabel verwendet. Eine Evaluation des Verfahrens auf Basis von drei realen und einem synthetischen Problem zeigte leichte

¹⁰⁹[NR98].

¹¹⁰Die Autoren sprechen hier von „demand leveling“ ([NR98], S. 296).

¹¹¹[YCC06].

¹¹²[YCC08].

¹¹³[RG09].

¹¹⁴[NTD04].

¹¹⁵[LHLX06].

¹¹⁶[AKW+09].

Vorteile bei der Klassifikationsgenauigkeit gegenüber C4.5, k-nächsten Nachbarn sowie naivem Bayes für drei der vier Probleminstanzen.

Ein multivariates Regressionsmodell zur mehrperiodischen Prognose des Sendungsaufkommens bei Express-, Paket- und Briefdienstleistern bzw. bei Anbietern typisierter Dienstleistungen wird von Lessing¹¹⁷ beschrieben. Grundlage der Prognose sind historische Zeitreihen über vergangene Sendungsmengen. Die unter Berücksichtigung der Einflüsse des deutschen Kalenders erstellten und im Rahmen des Prognoseprozesses bereinigten Prognosen der Sendungsmengen dienen als Grundlage für eine operative Kapazitätsplanung des Personal- und Transportmittelbedarfs in den Logistikzentren. Die Personalbedarfsermittlung basiert anschließend auf der Multiplikation der Sendungsmenge mit einem als bekannt angenommen Zeitbedarf pro Sendung.¹¹⁸ Die Ermittlung der benötigten Transportkapazität erfolgt in ähnlicher Weise, berücksichtigt jedoch produkt- und relationsspezifische Volumenfaktoren.

Die Autoren Tulabandhula & Rudin¹¹⁹ beschreiben ein auf maschinellem Lernen beruhendes Verfahren, um aus historischen Daten eine Unsicherheitsmenge¹²⁰ für unsichere Parameter zu konstruieren, die in einer robusten Optimierung eingesetzt werden kann. Die historischen Daten weisen Eigenschaften auf, die einen Datensatz charakterisieren, und einem Label, dessen Wert durch diese Eigenschaften beeinflusst wird. Die Eigenschaften eines neuen Datensatzes, dessen Label unbekannt ist, können somit genutzt werden, um das Label vorherzusagen. Es werden zwei verschiedene Ansätze mit jeweils zwei Methoden vorgestellt, um die Unsicherheitsmenge zu konstruieren. Für den ersten Ansatz werden empirische Vorhersagemodelle verwendet, wobei die erste Methode auf einer Mengenfunktion basiert und die zweite Methode bedingte Quantilfunktionen einsetzt. Im zweiten Ansatz wird eine Menge von als gut definierten Vorhersagemodellen genutzt, um die Unsicherheitsmenge zu definieren. Die erste Methode verwendet eine einzelne Menge von Vorhersagemodellen, wohingegen die zweite Methode zwei Mengen verwendet.

3.2 Multikriterielle Ablaufplanung unter Unsicherheit

Auf Basis der im vorherigen Abschnitt beschriebenen Verfahren lässt sich ein Informationsstand über die unsicheren Bearbeitungszeiten der Frachtabfertigungsaufträge ermitteln, der als Grundlage für die multikriterielle Ablaufplanung unter Unsicherheit Verwendung finden soll. Hierzu sind wie in Abschnitt 2.2.2.2 erläutert Ablaufplanungsverfahren notwendig, die in der Lage sind, diesen Informationsstand zu verwenden und Ablaufpläne zu erzeugen, die robust gegenüber ungünstigen Umweltsituationen sind. Des Weiteren sollen diese Verfahren zusätzliche Restriktionen, die sich aus benötigten Ressourcen ergeben, beachten.

¹¹⁷[Les04].

¹¹⁸Siehe ebd., S. 100, 169. Eine vergleichbare Rechnung zur Bestimmung der Bearbeitungsdauer von Aufträgen ist in Gleichung (2.1) auf Seite 18 aufgeführt. Die hiermit verbunden Probleme werden in dem Abschnitt ebenfalls angesprochen.

¹¹⁹[TR14].

¹²⁰Vgl. hierzu Abschnitt 3.2.1.1.

In der Literatur¹²¹ lassen sich grundsätzlich vier verschiedene Politiken, die zur Gliederung von Ablaufplanungsverfahren unter Unsicherheit dienen, identifizieren:

- **Dynamische Ablaufplanung:** Bei dieser Politik wird kein Ablaufplan im Vorhinein erstellt, sondern es wird bei Bedarf, z. B. der Ankunft eines neuen Auftrags, durch bspw. Prioritätsregeln (engl. *dispatching rules*) dynamisch über Zuordnungen und Sequenzierungen zu den Maschinen entschieden.¹²²
- **Prädiktiv-reaktive Ablaufplanung:** Hier wird zunächst ein Ablaufplan unter der Annahme deterministischer Eingabedaten erstellt. Störungen, die während der Ausführung dieses Plans auftreten, werden reaktiv behandelt.
- **Proaktive Ablaufplanung:** Mit dieser Politik wird ein Ablaufplan erstellt, der Wissen über stochastische Eingabedaten berücksichtigt, und versucht, einen Plan zu erstellen, der zuverlässiger als ein prädiktiver ist. Entsprechen die während der Ausführung des Plans auftretenden Störungen den bekannten stochastischen Einflüssen, so sind keine Änderungen notwendig, um den Plan ausführen zu können. Ist dies nicht der Fall, ist in der ausschließlich proaktiven Ablaufplanung jedoch kein Mechanismus zur Reaktion vorgesehen.
- **Proaktiv-reaktive Ablaufplanung:** Diese Politik verbindet die Vorteile eines proaktiven Ablaufplans mit der Möglichkeit, im Falle von Störungen, die nicht den berücksichtigten stochastischen Einflüssen entsprechen, reaktiv Anpassungen vornehmen zu können.

Im Folgenden liegt der Fokus auf Verfahren, die eine proaktive Ablaufplanung ermöglichen. Hierbei wird zwischen exakten Verfahren in Abschnitt 3.2.1 und heuristischen Verfahren in Abschnitt 3.2.2 unterschieden. Des Weiteren werden in Abschnitt 3.2.3 Methoden betrachtet, mit denen mehrere Ressourcen bei der Ablaufplanung berücksichtigt werden können und die somit die in Abschnitt 2.3.2 gestellten Anforderungen nach der Berücksichtigung von Personal- und Förderkapazitäten erfüllen. Den Abschluss bildet ein Literaturüberblick.

3.2.1 Exakte Verfahren

Auf mathematischer Optimierung beruhende exakte Verfahren zeichnen sich dadurch aus, dass sie garantiert die optimale Lösung bzw. Lösungen (sofern vorhanden) finden. Für Problemgrößen, wie sie häufig in der Praxis anzutreffen sind, können aufgrund der hohen Komplexität der Probleme optimale Lösungen jedoch nur schwer (Zeit- und Ressourcenaufwand) oder unter Umständen gar nicht ermittelt werden. Im Folgenden werden mit der robusten und stochastischen Optimierung zwei grundlegende Ansätze zur Optimierung unter Unsicherheit vorgestellt.¹²³

¹²¹Vgl. [SG09]; [BVLB09]; [OP09]; [LI08]; [ALM⁺05].

¹²²Vgl. [VHL03], S. 48 f. Auch als reaktive oder Online-Ablaufplanung bezeichnet.

¹²³Ein Literaturüberblick zu konkreten Anwendungen im Bereich der Ablaufplanung findet sich in Abschnitt 3.2.4.

3.2.1.1 Robuste Optimierung

Ein neuerer Ansatz zur Optimierung unter Unsicherheit bzw. zur Behandlung von Optimierungsmodellen mit unsicheren Daten ist die robuste Optimierung. Als eigenständiges Forschungsgebiet mit Ursprüngen im Bereich der robusten Kontrolltheorie, robusten Statistik und des maschinellen Lernens geht die robuste Optimierung auf Arbeiten von Soyster,¹²⁴ Kouvelis & Yu,¹²⁵ El Ghaoui et al.¹²⁶ sowie Ben-Tal & Nemirovski¹²⁷ zurück.¹²⁸

In der robusten Optimierung wird im Gegensatz zur stochastischen Optimierung (vgl. Abschnitt 3.2.1.2) nicht von einem stochastischen, sondern einem deterministischen und mengenbasierten Unsicherheitsmodell ausgegangen. Ziel ist es, eine Lösung zu ermitteln, die unter jeder möglichen Realisierung der unsicheren Eingabedaten innerhalb der definierten Unsicherheitsmenge zulässig ist, statt – wie in der stochastischen Optimierung – eine probabilistische Absicherung gegenüber stochastischer Unsicherheit zu erreichen.¹²⁹

Am Beispiel der allgemeinen Darstellung $\min_x \{c^T x + d \mid Ax \leq b\}$ eines linearen Programms (engl. *Linear Program* (LP)) mit $x \in \mathbb{R}^n$, dem Vektor der Entscheidungsvariablen, $c \in \mathbb{R}^n$ und $d \in \mathbb{R}$ als Koeffizienten bzw. Konstanten der Zielfunktion, A als $m \times n$ -Matrix der Restriktionen und $b \in \mathbb{R}^m$ als *right hand side*-Vektor soll im Folgenden die robuste Optimierung, wie sie von Ben-Tal et al.¹³⁰ beschrieben wird, erläutert werden. In der robusten Optimierung wird ein unsicheres LP durch eine Menge

$$\left\{ \min_x \{c^T x + d \mid Ax \leq b\} \right\}_{(c,d,A,b) \in \mathcal{U}} \quad (3.5)$$

von LPs beschrieben, wobei die Daten (c, d, A, b) zu der Unsicherheitsmenge \mathcal{U} gehören. Es wird nach einer Lösung gesucht, die für jede Realisierung der Daten innerhalb der Unsicherheitsmenge \mathcal{U} zulässig ist und dabei den Zielfunktionswert für den schlechtesten Fall (engl. *Worst Case*) optimiert. Die Qualität einer robusten Lösung wird über den garantierten Wert der Originalzielfunktion quantifiziert, womit die robuste Lösung über das folgende Programm (bezeichnet als *Robust Counterpart* (RC)) gefunden wird:

$$\min_x \left\{ \sup_{(c,d,A,b) \in \mathcal{U}} [c^T x + d] \mid Ax \leq b \forall (c, d, A, b) \in \mathcal{U} \right\}$$

Dieses kann zu dem folgenden Programm umgeformt werden:

$$\min_{x,t} \{t \mid c^T x + d \leq t, Ax \leq b \forall (c, d, A, b) \in \mathcal{U}\}, \quad (3.6)$$

¹²⁴[Soy73].

¹²⁵[KY97].

¹²⁶[EL97]; [EOL98].

¹²⁷[BN98]; [BN99]; [BN00].

¹²⁸Siehe [BEN09], S. xvif., 25 f.

¹²⁹Siehe [BBC11], S. 465.

¹³⁰[BEN09].

dessen Lösung eine robust optimale und robust zulässige Lösung für das LP aus (3.5) darstellt.¹³¹

Die Frage, wie schwer das Problem (3.6) ist, hängt von dem Ursprungsproblem und den Eigenschaften der Unsicherheitsmenge \mathcal{U} ab. Ben-Tal et al. schreiben hierzu: „the RC of the uncertain LO¹³² problem with uncertainty set \mathcal{U} is computationally tractable whenever the convex uncertainty set \mathcal{U} itself is computationally tractable.“¹³³ Mit „computationally tractable“ werden solche Probleme bezeichnet, die in äquivalente Probleme umgeformt werden können, für die Algorithmen mit einer Worst-Case-Laufzeit, die polynomiell von der Eingabegröße abhängt, vorhanden sind.¹³⁴ Ist das Ursprungsproblem ein gemischt-ganzzahliges Programm (engl. *Mixed-integer Program* (MIP)), dann ist die vorstehende Aussage nicht garantiert. Für MIPs sind bereits die Ursprungsprobleme häufig nicht in polynomieller Zeit lösbar und nach Ben-Tal et al. gilt dies somit ebenfalls für deren RCs.¹³⁵

Ehrgott et al.¹³⁶ erweitern das zuvor beschriebene Konzept der minmax-Robustheit nach Ben-Tal et al.¹³⁷ von Einziel- auf Mehrzieloptimierungsprobleme. Hierzu verwenden sie eine Ordnungsrelation, um robust effiziente Lösungen für unsichere Mehrzieloptimierungsprobleme zu definieren. Als Lösungsverfahren werden Ansätze aus dem Bereich der deterministischen Mehrzieloptimierung¹³⁸ diskutiert. Hierzu gehören die Skalarisierung durch gewichtete Summen oder ϵ -Restriktionen sowie die zielfunktionsweise Betrachtung des Worst Cases.

Der Ansatz nach Ehrgott et al. ist aufgrund der gewählten Ordnungsrelation ein pessimistischer Ansatz, der gegen die Menge an Worst Cases absichert.¹³⁹ Vier weitere Ordnungsrelationen für unsichere Mehrzieloptimierungsprobleme werden von Ide & Köbis¹⁴⁰ vorgestellt und untersucht. Erstens wird ein optimistischer Ansatz, der gegen die Menge an Best Cases absichert, vorgestellt. Zweitens wird ein intermediärer Ansatz beschrieben, der eine Kombination des pessimistischen und optimistischen Ansatzes darstellt. Mit der dritten Ordnungsrelation werden Lösungen gefunden, die sowohl in der Menge der Best als auch Worst Cases nicht-dominiert sind. Als letztes wird eine Ordnungsrelation beschrieben, um alle mit Sicherheit dominierten Lösungen zu entfernen.

3.2.1.2 Stochastische Optimierung

Die stochastische Optimierung ist eine weitere Möglichkeit zur Behandlung von Optimierungsmodellen mit unsicheren Eingangsdaten, die auf Dantzig¹⁴¹ zurückgeht. Im

¹³¹Siehe [BEN09], S. 7–15.

¹³²Die Abkürzung LO steht für „Linear Optimization“.

¹³³[BEN09], S. 16.

¹³⁴Siehe [BBC11], S. 465.

¹³⁵Siehe [BEN09], S. 26.

¹³⁶[EIS14].

¹³⁷[BEN09].

¹³⁸Siehe z. B. [Ehr05].

¹³⁹Siehe [IK14], S. 101.

¹⁴⁰[IK14].

¹⁴¹[Dan55].

Unterschied zur robusten Optimierung werden hier die Eingangsdaten (c, d, A, b) probabilistisch, z. B. als Zufallsvariablen mit bekannten Wahrscheinlichkeitsverteilungen, modelliert. Ziel ist es, eine Lösung zu ermitteln, die für jede bzw. fast jede Realisierung der Eingangsdaten zulässig ist und dabei häufig den Erwartungswert der Zielfunktion¹⁴² optimiert.

Es wird erneut von einem linearen Programm ausgegangen, das zu einem mehrstufigen stochastischen Programm erweitert wird. Werden für dieses Modell n Stufen betrachtet, so bezeichne $\xi_1 = (c_1, A_1, b_1)$ einen Vektor der sicheren Daten der ersten Stufe und $\xi_t = (c_t, B_t, A_t, b_t)$ einen Zufallsvektor der Stufe $t = 2, \dots, n$, wobei einige oder alle Elemente dieses Vektors zufällig sein können. Das Modell kann anschließend wie folgt dargestellt werden:¹⁴³

$$\min_{x_1 \in \mathcal{X}_1} f_1(x_1) + \mathbb{E} \left[\inf_{x_2 \in \mathcal{X}_2(x_1, \xi_2)} f_2(x_2, \xi_2) + \mathbb{E} \left[\dots + \mathbb{E} \left[\inf_{x_n \in \mathcal{X}_n(x_{n-1}, \xi_n)} f_n(x_n, \xi_n) \right] \right] \right] \quad (3.7a)$$

mit

$$f_t(x_t, \xi_t) := c_t^T x_t \quad (3.7b)$$

$$\mathcal{X}_1 := \{x_1 \mid A_1 x_1 = b_1, x_1 \geq 0\} \quad (3.7c)$$

$$\mathcal{X}_t(x_{t-1}, \xi_t) := \{x_t \mid B_t x_{t-1} + A_t x_t = b_t, x_t \geq 0\} \quad (t = 2, \dots, n) \quad (3.7d)$$

In jeder Stufe $t = 1, \dots, (n-1)$ werden die Entscheidungen der Entscheidungsvariable x_t umgesetzt, bevor die unsicheren Daten ξ_{t+1} bekannt sind. Nach Bekanntwerden der Daten auf der nächsten Stufe werden die Entscheidungen x_{t+1} ermittelt und ausgeführt.

Die Anforderung an eine Lösung, für jede Realisierung der unsicheren Daten zulässig zu sein, ist sehr restriktiv und kann dazu führen, dass keine Lösung gefunden wird. Aus diesem Grund können probabilistische Nebenbedingungen (engl. *chance/probabilistic constraints*)¹⁴⁴ eingeführt werden, die lediglich mit einer hohen Wahrscheinlichkeit (beschrieben über das Signifikanzniveau $\alpha_i \in (0, 1)$) eingehalten werden müssen:¹⁴⁵

$$\mathbb{P}(a_i^T x \leq b^i) \geq 1 - \alpha_i \quad (i = 1, \dots, m)$$

Hierbei bezeichnet a_i mit $1 \leq i \leq m$ die i -te Restriktion bzw. Zeile der Matrix A und α_i das entsprechende Signifikanzniveau dieser Restriktion. Diese Formulierung beschreibt einzelne probabilistische Nebenbedingungen, bei denen keine Korrelation zwischen den Reihen der Matrix A besteht oder eine Korrelation angenommen wird. Besteht eine Korrelation, so müssen die Nebenbedingungen zusammen betrachtet werden:

$$\mathbb{P}(a_i^T x \leq b^i, \forall i = 1, \dots, m) \geq 1 - \alpha$$

¹⁴²Für die Berücksichtigung von mehreren Zielfunktionen sei auf [Abd12] und die dort aufgeführten Quellen verwiesen.

¹⁴³Siehe [SDR09], S. 63 f.

¹⁴⁴Das Konzept der probabilistische Nebenbedingungen geht zurück auf Charnes & Cooper (siehe [CC58]; [CC59]).

¹⁴⁵Vgl. [BL11], S. 124; [SDR09], S. 90.

Für stochastische Modelle, wie sie in dem Modell (3.7) beschrieben sind, können häufig keine geschlossenen Lösungen angegeben werden. Die Lösungsfindung kann dann z. B. über die Betrachtung einer endlichen Menge von n Szenarien erfolgen. Jedes Szenario k mit $1 \leq k \leq n$ beschreibt hierbei eine mögliche Realisierung der unsicheren Daten $\xi_{t,k}$ der Stufe t und besitzt ein bestimmtes Gewicht p_k . Der Erwartungswert kann aus der gewichteten Summe der n Szenarien berechnet werden.¹⁴⁶ Für MIPs, insbesondere für in der Praxis vorkommende relevante Problemgrößen, sind diese Modelle nur schwer oder gar nicht lösbar.

3.2.2 Heuristische Verfahren

Die im vorherigen Abschnitt beschriebenen exakten Verfahren liefern auf der einen Seite garantiert eine optimale Lösung bzw. pareto-optimale Lösungen, sofern diese existiert bzw. existieren, auf der anderen Seite sind sie für praxisrelevante Problemgrößen und die hieraus resultierenden Optimierungsprobleme aufgrund der Komplexität häufig nur bedingt einsetzbar. Aus diesem Grund werden häufig Heuristiken¹⁴⁷ verwendet, die zwar nicht feststellen können, dass eine gefundene Lösung optimal ist, aber häufig für die Praxis annehmbare Lösungen ermitteln. Eine Heuristik ist somit eine Methode „which, on the basis of experience or judgement, seems likely to yield a reasonable solution to a problem, but which cannot be guaranteed to produce the mathematically optimal solution.“¹⁴⁸

Nach Domschke & Drexl¹⁴⁹ lassen sich Heuristiken unterteilen in *Eröffnungsverfahren* zur Bestimmung einer zulässigen Lösung; *Verbesserungsverfahren* zur iterativen Verbesserung einer gegebenen zulässigen Lösung; vorzeitig abgebrochene, d. h. *unvollständig exakte Verfahren* sowie Kombinationen aus den zuvor genannten. Eine weitere Unterscheidung ist diejenige in problemspezifische Heuristiken und problemunabhängige Metaheuristiken.¹⁵⁰ Während erstere auf ein spezielles Optimierungsproblem zugeschnitten sind und daher nur bei diesen angewendet werden können, sind Metaheuristiken allgemeine Strategien, die auf fast beliebige Problemstellungen übertragbar sind.¹⁵¹ Eine Definition liefern Voß et al.: „A meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. [...]“¹⁵². Metaheuristiken lassen sich somit u. a. weiter anhand der verwendeten Anzahl von Lösungen für das Optimierungsproblem klassifizieren. Hier kann zwischen solchen, die eine einzige Lösung verbessern (S-Metaheuristik), und solchen, die

¹⁴⁶Vgl. [SDR09], S. 3f.

¹⁴⁷Heuristik leitet sich aus dem griechischen *heuriskein* ab, was sich mit „finden“ oder „entdecken“ übersetzen lässt.

¹⁴⁸[Sil04], S. 936.

¹⁴⁹Siehe [DD11], S. 128.

¹⁵⁰Die griechische Präposition *meta*, die hier als Präfix Verwendung findet, beschreibt, dass sich Metaheuristiken auf einer höheren bzw. abstrakteren Ebene als Heuristiken befinden.

¹⁵¹Vgl. [Tal09], S. 1, 21.

¹⁵²[VMOR99], S. ix.

populationsbasiert vorgehen (P-Metaheuristik) und somit gleichzeitig mehrere Lösungen verbessern, unterschieden werden.¹⁵³

Aufgrund der Anforderung einer Mehrzieloptimierung (vgl. Abschnitt 2.3.2) wird die Betrachtung im Folgenden auf Metaheuristiken für Mehrzieloptimierungsprobleme eingeschränkt. Der Fokus liegt hierbei auf den P-Metaheuristiken,¹⁵⁴ da diese aufgrund der gleichzeitigen Betrachtung mehrerer möglicher Lösungen besser für die Approximation der Pareto-Front¹⁵⁵ innerhalb einer einzigen Ausführung und damit für die Verwendung bei MOPs geeigneter erscheinen als S-Metaheuristiken.¹⁵⁶ Hierzu werden in Abschnitt 3.2.2.1 zunächst die Grundlagen von multikriteriellen P-Metaheuristiken vorgestellt. In Abschnitt 3.2.2.2 folgt eine Beschreibung, wie mit Unsicherheit umgegangen werden kann.

3.2.2.1 Multikriterielle populationsbasierte Metaheuristiken

Die Berücksichtigung mehrerer Ziele bei der Lösung von Optimierungsproblemen ist in vielen Bereichen der Wissenschaft und Industrie eine wichtige Anforderung. Aufgrund der zunehmenden Komplexität dieser Probleme wächst das Interesse und somit die Entwicklung verschiedener multikriterieller Metaheuristiken seit Ende der 1980er.¹⁵⁷ Grundlage sind hierbei häufig Naturanalogien (z. B. Evolution, Abkühlungsprozesse, Schwarm-Intelligenz u. v. m.) und insbesondere populationsbasierte Verfahren. Ungeachtet der Vielfältigkeit¹⁵⁸ der entwickelten Verfahren bestehen Gemeinsamkeiten, die im Folgenden näher erläutert werden.

Nach Talbi¹⁵⁹ ist für jede Metaheuristik zunächst festzulegen, wie die Lösungen repräsentiert, die Zielfunktion(en) gestaltet und Restriktionen bei der Optimierung beachtet werden. Die Lösungsrepräsentation, die den Repräsentationsraum R beschreibt und auf der eine Metaheuristik aufbaut, muss die folgenden Eigenschaften erfüllen, um eine effektive und effiziente Suche nach einer Lösung zu ermöglichen.¹⁶⁰

- Vollständigkeit: Es muss möglich sein, jede mögliche Lösung für das Optimierungsproblem abbilden zu können.
- Konnektivität: Zwischen zwei beliebigen Lösungen des Lösungsraumes muss es eine Verbindung geben, d. h., jede Lösung muss durch Suchen erreichbar sein.
- Effizienz: Die Repräsentation sollte einfach durch die Suchoperatoren der Heuristik geändert werden können.

¹⁵³Vgl. [Tal09], S. 25.

¹⁵⁴In der Literatur sind ebenfalls Arbeiten zu finden, die Erweiterungen der S-Metaheuristiken (z. B. Tabu-Suche oder Simulated Annealing) beschreiben, um diese für Mehrzieloptimierungsprobleme einsetzen zu können. Eine Übersicht zu diesen Arbeiten liefern Talbi et al. [TBNA12], S. 287–292.

¹⁵⁵Vgl. Abschnitt 2.2.2.2.

¹⁵⁶Vgl. [Tal09], S. 309, 337, 373.

¹⁵⁷Vgl. ebd., S. 308.

¹⁵⁸Siehe z. B. [GP10].

¹⁵⁹Siehe [Tal09], S. 34–53.

¹⁶⁰Siehe ebd., S. 35.

Zu *genetischen Algorithmen* (GAs)¹⁶¹ für Ablaufplanungsprobleme werden von den Autoren Cheng et al.¹⁶² drei weitere Anforderungen an eine Repräsentation angeführt:

- Zulässigkeit: Es sollten nach Möglichkeit nur solche Lösungen repräsentiert werden können, die alle Restriktionen einhalten und damit zulässig sind.
- Gültigkeit: Es dürfen nur Lösungen dargestellt werden, die eine Lösung für das Optimierungsproblem darstellen.
- Eindeutigkeit: Aus Effizienzgründen sollte die Abbildung zwischen Repräsentation und Lösung eine bijektive Abbildung sein, da diese den Suchraum einschränkt.

Des Weiteren kann zwischen direkten und indirekten Kodierungsansätzen unterschieden werden. Die direkte Kodierung verwendet eine Repräsentation, die direkt evaluiert werden kann, da alle notwendigen Informationen encodiert sind. Die indirekte Kodierung muss hingegen zunächst zu einer Lösung dekodiert werden, bevor sie evaluiert werden kann.¹⁶³

Mit Hilfe der Zielfunktion(en) wird die Qualität (auch: Fitness) einer Lösung quantitativ bewertet und dient dazu, die Suche der Metaheuristik zu führen. Hierfür ist eine Lösung, im Falle einer indirekten Kodierung, zunächst mit einer Dekodierfunktion $d: R \rightarrow S$ vom Repräsentationsraum R in den Lösungsraum S zu überführen. Anschließend kann diese Lösung mit einer Evaluationsfunktion $f: S \rightarrow \mathbb{R}$ bewertet werden.¹⁶⁴

Zur Berücksichtigung von Restriktionen wird häufig an der Zielfunktion angesetzt und diese mit Strafkosten versehen, sofern eine Lösung nicht zulässig ist. Hierfür sind jedoch Strafkostenparameter zu definieren und somit ist dieses Vorgehen mit zusätzlichem Aufwand verbunden. Deb¹⁶⁵ beschreibt einen alternativen Ansatz für GAs, der ohne die Definition von Strafkostenparametern auskommt und auf einem paarweisen Vergleich von zwei Lösungen während der Auswahlphase beruht. Weitere alternative Ansätze versuchen die Zulässigkeit einer Lösung zu erhalten, unzulässige Lösungen auszuschließen bzw. zu reparieren oder durch die Dekodierfunktion lediglich zulässige Lösungen zu erzeugen.¹⁶⁶

Für P-Metaheuristiken sind nach Talbi¹⁶⁷ zusätzlich eine initiale Menge (auch: Population) von Lösungen zu bestimmen sowie ein Abbruchkriterium festzulegen. Die Initialpopulation spielt eine entscheidende Rolle, da mit dieser die iterative P-Metaheuristik beginnt. Neben den Anforderungen insbesondere nach der Zulässigkeit und Gültigkeit der initialen Lösungen, die problemspezifisch zu bestimmen sind, sollte die Initialpopulation diversifiziert genug sein, um eine verfrühte Konvergenz zu suboptimalen Lösungen zu vermeiden. Der Abbruch der iterativen Suche erfolgt schließlich über ein Abbruchkriterium. Dieses kann entweder statisch nach einer gewissen Anzahl Iterationen erfolgen oder

¹⁶¹Genetische Algorithmen stellen ein Teilgebiet evolutionärer Metaheuristiken dar, die wiederum zu den P-Metaheuristiken gehören und auf dem Prinzip der Evolution basieren. Genetische Algorithmen gehen zurück auf Holland [Hol75]. Eine umfangreiche Beschreibung der Grundlagen erfolgt bei Goldberg [Gol89].

¹⁶²Siehe [CGT96], S. 985.

¹⁶³Vgl. [Tal09], S. 41 ff.

¹⁶⁴Siehe ebd., S. 43.

¹⁶⁵[Deb00].

¹⁶⁶Vgl. [Tal09], S. 48–53; und für evolutionäre Metaheuristiken [MS96].

¹⁶⁷Siehe [Tal09], S. 193–199.

adaptiv im Falle von P-Metaheuristiken z. B. nach einer gewissen Anzahl von Iterationen, in denen keine Verbesserung der Population mehr eingetreten ist.

Speziell für evolutionäre Algorithmen, die zu den meist untersuchten P-Metaheuristiken gehören, sind zusätzlich Entscheidungen bzgl. des Selektionsmechanismus sowie der Reproduktionsoperatoren zu treffen.¹⁶⁸ Der Selektionsmechanismus dient dazu, die Lösungen (auch: Individuen) auszuwählen, aus denen neue Lösungen für die nächste Iteration (auch: Generation) erstellt werden. Individuen mit einer höheren Fitness, gemessen anhand der Zielfunktion(en), besitzen hierbei eine höhere Auswahlwahrscheinlichkeit. Die Reproduktionsoperatoren (Mutation und Rekombination/Crossover) setzen an der Repräsentation (auch: Genotyp) einer Lösung an und verändern diese, um neue Individuen zu erzeugen. Bei der Mutation wird ein einzelnes Individuum leicht verändert, um die Diversität zu erhöhen und lokale Minima zu überwinden. Die Rekombination, bei der zwei oder mehr Individuen miteinander kombiniert werden, dient dazu, Eigenschaften an Nachfolgeindividuen weiterzugeben. Die Ausgestaltung der Reproduktionsoperatoren hängt erneut von dem spezifischen Problem und insbesondere von der gewählten Repräsentation einer Lösung ab und sollte so gestaltet sein, dass ausschließlich zulässige und gültige Lösungen erzeugt werden können.

Für multikriterielle P-Metaheuristiken¹⁶⁹ ist schließlich festzulegen, wie die Fitness einer Lösung unter mehreren Zielfunktionen ermittelt und die Diversität der ermittelten pareto-optimalen Lösungen sichergestellt wird sowie die besten Lösungen (Elite-Prinzip) über die Iterationen erhalten werden.¹⁷⁰ Bei Problemen mit mehreren Zielfunktionen besitzt jede Lösung einen Vektor von Zielfunktionswerten statt eines einzelnen Zielfunktionswerts. Dieser Vektor ist mit einem skalaren Fitnesswert zu bewerten. Hierfür sind verschiedene Verfahren vorhanden, wie z. B. die Skalarisierung der einzelnen Zielfunktionswerte durch Gewichte zu einem einzigen Wert oder die Anwendung von Dominanz- bzw. Paretoansätzen, die den Lösungen verschiedene Ränge zuordnen. Zur Approximation der Pareto-Front sollten möglichst diversifizierte Lösungen erzeugt werden, die pareto-optimal sind. Hierzu werden Verfahren angewendet, die Lösungen mit einer hohen Anzahl weiterer Lösungen in der direkten Nachbarschaft abwerten, da von diesen kein so hoher Informationsgewinn zu erwarten ist wie von Lösungen mit einer geringeren Dichte von Nachbarlösungen. Die während der Iterationen gefundenen besten Lösungen, z. B. die pareto-optimalen Lösungen, werden in einem Archiv gespeichert. Dieses Archiv kann passiv genutzt und somit lediglich zur Speicherung verwendet werden oder die Lösungen dieses Archivs werden aktiv in den einzelnen Iterationen mit einbezogen.

3.2.2.2 Multikriterielle evolutionäre Metaheuristiken unter Unsicherheit

Für die im vorherigen Abschnitt beschriebenen Entwurfsentscheidungen für multikriterielle P-Metaheuristiken wurde von deterministischen Eingabedaten ausgegangen.

¹⁶⁸Vgl. [Tal09], S. 199, 206–220; [GF13], S. 964 f.

¹⁶⁹Ein bekannter Vertreter multikriterieller evolutionärer Metaheuristiken ist der *Nondominated Sorting Genetic Algorithm II* (NSGA-II) von Deb et al. [DPAM02]. Siehe zudem [KCS06], S. 995, für einen Vergleich mit weiteren bekannten mehrzielorientierten GAs.

¹⁷⁰Siehe [Tal09], S. 323–350.

Im Folgenden werden Erweiterungen dargestellt, um Unsicherheit bei den Eingabedaten zu berücksichtigen. Der Fokus liegt hierbei auf den multikriteriellen evolutionären Metaheuristiken (engl. *Multi-objective Evolutionary Algorithms* (MOEAs)),¹⁷¹ da diese in den letzten Jahren umfangreich erforscht¹⁷² wurden und besonders geeignet¹⁷³ zur Lösung von MOPs erscheinen.

Jin & Branke¹⁷⁴ unterteilen Unsicherheit bei evolutionärer Optimierung in vier Kategorien:

- Rauschen in der Fitnessfunktion: In diesem Fall besteht die Unsicherheit in der Evaluation der Zielfunktion $f(\vec{x}) + \delta$ mit \vec{x} (Vektor der Entscheidungsvariablen), f (originale Zielfunktion) und δ (Rauschen).
- Unsicherheit von Design-/Entscheidungsvariablen oder Umgebungsparametern: Nach der Ermittlung der optimalen Lösung können sich die Umgebungsparameter ändern oder die Entscheidungsvariablen können nicht exakt umgesetzt werden. Die ermittelten Lösungen und ihre Zielfunktion(en) sollten robust gegenüber diesen Änderungen sein.
- Approximation der Fitnessfunktion: Wenn die Evaluation der Fitnessfunktion zu aufwendig ist, kann sie approximiert werden. Dies führt zu Approximationsfehlern.
- Dynamische Fitnessfunktionen: In diesem Fall ist die Fitnessfunktion deterministisch, aber zeitabhängig. Die Lösung sollte daher kontinuierlich an das sich ändernde Optimum angepasst werden.

Der Fokus liegt im Folgenden auf Erweiterungen von klassischen evolutionären Algorithmen, um die Unsicherheit, die sich aus der Änderung von Umgebungsparametern ergibt, in den Algorithmus einzubeziehen und somit robuste Lösungen zu ermitteln.

In Abhängigkeit davon, wie das Maß der Robustheit in die evolutionären Algorithmen integriert wird, können diese in zwei Kategorien unterteilt werden:¹⁷⁵

- Der einkriterielle Ansatz verwendet das Robustheitsmaß anstelle der ursprünglichen Zielfunktionen.
- Der multikriterielle Ansatz berücksichtigt das Robustheitsmaß als zusätzliche Zielfunktion oder Restriktion neben den ursprünglichen Zielfunktionen.

Für den einkriteriellen Ansatz wird als Robustheitsmaß häufig der Erwartungswert der ursprünglichen Zielfunktion verwendet. Der Erwartungswert kann jedoch häufig nicht analytisch bestimmt werden und muss daher geschätzt werden.¹⁷⁶ Hierzu wird die

¹⁷¹MOEAs werden häufig auch als *evolutionary multi-objective optimization (algorithms)* (EMO, EMOO oder EMOA) bezeichnet. Für eine Einführung in MOEAs siehe z. B. [Deb01] oder [CLV07].

¹⁷²Coello Coello listet Referenzen auf, in denen MOEAs Verwendung finden [Coe14].

¹⁷³Vgl. [Coe06], S. 29.

¹⁷⁴Siehe [JB05], S. 303 f.

¹⁷⁵Siehe [GT09], S. 214 ff.; [TG08], S. 282; [Goh07], S. 176.

¹⁷⁶Vgl. [BAM13], S. 3.

ursprüngliche Zielfunktion unter Berücksichtigung der Verteilung des unsicheren Umgebungsparameters und hieraus gezogener Stichproben wiederholt evaluiert und es wird der Durchschnittswert gebildet. Dieses Vorgehen wird als explizite Durchschnittswertbildung bezeichnet. Der zusätzliche Aufwand bei der Evaluation einer einzelnen Lösung kann bspw. durch implizite Durchschnittswertbildung vermindert werden. Hierbei wird in jeder Generation auf die Entscheidungsvariablen einer Lösung ein zufälliger Störeinfluss ausgeübt, bevor die Lösung evaluiert wird. Unter der Annahme einer unbegrenzten Populationsgröße ist dieses Vorgehen äquivalent zur Optimierung des Erwartungswerts der Zielfunktion.¹⁷⁷ Weitere Möglichkeiten wie z. B. Techniken zur Varianzreduktion, häufigere Evaluierung wichtiger Lösungen oder die Nutzung von Nachbarlösungen werden von Branke et al.¹⁷⁸ und den dort aufgeführten Quellen beschrieben.

Beim multikriteriellen Ansatz kann nach Goh¹⁷⁹ zwischen zwei Möglichkeiten unterschieden werden, das Robustheitsmaß im MOEA zu berücksichtigen. Bei der ersten wird das Robustheitsmaß als weitere Zielfunktion neben der oder den ursprünglichen Zielfunktion(en) aufgenommen. Hiermit sind Trade-off-Analysen zwischen Performance und Robustheit möglich.¹⁸⁰ Die zweite Möglichkeit wird von Deb & Gupta¹⁸¹ beschrieben und berücksichtigt das Robustheitsmaß als zusätzliche Restriktion.

3.2.3 Ablaufplanung unter multiplen Ressourcenbeschränkungen

Im Gegensatz zu der klassischen Ablaufplanung, bei der lediglich Maschinen als einschränkende Ressource betrachtet werden, müssen für reale Systeme häufig weitere Restriktionen, wie Personal- und Förderkapazitäten, Berücksichtigung finden.¹⁸² Eine Restriktion kann hierbei jede Ressource sein, die die Leistung des Systems einschränkt bzw. reguliert und somit Auswirkungen auf die Performance hat.¹⁸³

In der Literatur wird neben den Maschinen häufig das Personal als weitere einschränkende Ressource für Systeme betrachtet, bei denen weniger Personal als Maschinen vorhanden ist. Diese Systeme werden mit dem Begriff *Dual-resource Constrained Systems/Scheduling* (DRCS) bzw. präziser als „labour and machine-limited system“¹⁸⁴ bezeichnet. Die Mitarbeiter werden hierbei nach Verfügbarkeit und ihren Fähigkeiten bzgl. der Anforderungen der einzelnen Aufträge diesen zugewiesen. Hierbei stehen insbesondere die Zuweisungsregeln, mit denen entschieden wird, wer zu welchem Zeitpunkt an welcher Maschine arbeiten soll, im Zentrum des Interesses.

Bokhorst et al.¹⁸⁵ beschreiben vier Wer-Regeln (engl. *who-rules*), mit denen entschieden wird, welcher Mitarbeiter aus einer Gruppe bereitstehender Mitarbeiter einer Aufgabe bzw. Maschine zugewiesen wird. Die Mitarbeiter unterscheiden sich in ihren Kenntnissen

¹⁷⁷Vgl. [JB05], S. 306; [TG08], S. 282, und die dort genannten Quellen.

¹⁷⁸Siehe [BAM13], S. 4.

¹⁷⁹Siehe [Goh07], S. 178.

¹⁸⁰Vgl. [JB05], S. 306.

¹⁸¹[DG06].

¹⁸²Vgl. Abschnitte 2.2.2.3 und 2.2.2.4.

¹⁸³Vgl. [GD96], S. 2078.

¹⁸⁴[CM07], S. 219.

¹⁸⁵[BSG04].

und Fähigkeiten bzgl. des Auftrags sowie den Maschinen, für die sie zuständig sind. Der Fokus liegt hierbei auf der Anwendung der Wann-Regeln in einer Simulationsstudie.

Mit Wann-Regeln (engl. *when-rules*) wird bestimmt, wann ein Mitarbeiter zu einer anderen Maschine wechseln darf. Hierbei stellen zentralisierte und dezentralisierte Kontrolle die zwei Hauptarten der Wann-Regeln dar. Während bei ersterer der Mitarbeiter mit dem Ende eines Auftrags die Maschine wechseln darf, ist bei letzterer zunächst die Warteschlange an der aktuellen Maschine vollständig abzuarbeiten.

Mit Wohin-Regeln (engl. *where-rules*) wird schließlich bestimmt, zu welcher Maschine der Mitarbeiter wechselt.¹⁸⁶ Die *Dual-resource Constrained*(DRC)-Probleme werden hierbei analytisch, über Simulation oder mittels (Meta-)Heuristiken gelöst.¹⁸⁷

Werden neben den Maschinen und dem Personal weitere Ressourcen (Hilfsressourcen) betrachtet, die für die Produktivität dieser beiden notwendig sind, so handelt es sich um *Multiple-resource Constrained Systems/Scheduling* (MRCS).¹⁸⁸ Zu diesen weiteren Ressourcen können sowohl Werkzeuge für die Bearbeitung des Auftrags als auch Transportsysteme (engl. *Material Handling System* (MHS)) für den Materialfluss gezählt werden.¹⁸⁹ Eine integrierte Betrachtung der Ablaufplanung für Maschinen erfolgt hierbei häufig in Kombination mit *Automated Guided Vehicles* (AGVs) als Transportsystem.¹⁹⁰ Als Lösungsverfahren stehen die gleichen Verfahren wie für DRC-Probleme zur Verfügung. Khayat et al.¹⁹¹ unterscheiden hier zwischen Online- und Offline-Verfahren. Während erstere bspw. auf Basis von Prioritätsregeln das System in der Ausführung steuern, wird mit letzteren eine Planung im Vorhinein für das statische System ausgeführt.

3.2.4 Literaturüberblick

Für das in Abschnitt 2.2.2 dargestellte Ablaufplanungsproblem sind keine Arbeiten bekannt, die alle definierten Anforderungen an ein Verfahren zur Lösung (vgl. Abschnitt 2.3.2) erfüllen. Im Folgenden werden daher Beiträge betrachtet, die einen Teil dieser Anforderungen berücksichtigen. Zunächst wird ein Literaturüberblick über verschiedene Maße zur Quantifizierung der Robustheit sowie Stabilität einer Lösung gegeben. Hierauf aufbauend werden Arbeiten angeführt, in denen Modellierungsmöglichkeiten zur Erstellung von weniger konservativen Lösungen vorgestellt werden. Anschließend wird eine Übersicht zu konkreten Ablaufplanungsproblemen gegeben, die dem in Abschnitt 2.2.2.2 definierten Ablaufplanungsproblem ähneln sowie Teilaspekte der definierten Anforderungen (vgl. Abschnitt 2.3.2) berücksichtigen und dabei durch analytische oder heuristische Verfahren gelöst werden. Den Abschluss bildet eine Übersicht zu Beiträgen, die sich mit Personal- oder Förderkapazitäten als einschränkender Ressource beschäftigen.

¹⁸⁶Siehe [XXX11], S. 313, und die dort aufgeführten Quellen für eine Übersicht und Schlussfolgerungen für verschiedene Wann- und Wohin-Regeln.

¹⁸⁷Vgl. ebd., S. 314 f.

¹⁸⁸Vgl. [CM07], S. 219.

¹⁸⁹Siehe [GD96], S. 2077.

¹⁹⁰Vgl. [KLR06], S. 1819 ff., und die dort aufgeführten Quellen.

¹⁹¹Siehe ebd., S. 1819.

Kouvelis & Yu¹⁹² modellieren die Unsicherheit der Eingabedaten über die Definition einer Menge von Szenarien und beschreiben drei Maße zur Quantifizierung der Robustheit: Mit dem *absoluten Robustheitsmaß* (auch: minimax-Kriterium) wird nach einer Lösung gesucht, die für alle Eingabedatenszenarien und zulässigen Entscheidungen die maximalen Kosten minimiert und somit gegen den Worst Case absichert. Das Maß der *absoluten Abweichung*¹⁹³ wird verwendet, um eine Lösung zu ermitteln, die die beste Worst-Case-Abweichung vom Optimum (gemessen über ein Performancekriterium) über alle zulässigen Lösungen und für alle Eingabedatenszenarien besitzt. Das Maß der *relativen Abweichung* entspricht dem vorherigen, jedoch wird hier die prozentuale Abweichung betrachtet.

Ein Stabilitätsmaß zur Quantifizierung der Ähnlichkeit von zwei Ablaufplänen beschreibt Jensen.¹⁹⁴ Grundlage bildet ein Maß, mit dem die Überdeckung zwischen zwei Aufträgen bemessen werden kann. Als Überdeckung wird die Dauer bezeichnet, in denen ein Auftrag in beiden Ablaufplänen zur gleichen Zeit ausgeführt wird. Das Stabilitätsmaß basiert auf der normierten Summe der Überdeckungen aller Aufträge.

Beyer & Sendhoff¹⁹⁵ beschreiben verschiedene Robustheitsmaße für deterministisch, probabilistisch und durch Fuzzy-Logik modellierte Unsicherheiten. Deterministische Unsicherheiten werden von den Autoren als Mengen, in denen die unsicheren Parameter liegen können, beschrieben. Das vorgeschlagene Robustheitsmaß entspricht einer Worst-Case-Betrachtung und basiert auf dem maximalen Wert einer Zielfunktion f in einer nutzerdefinierten Nachbarschaft um die Lösung x . Für die probabilistischen Unsicherheiten werden die Parameter als Zufallsvariablen mit bekannten Verteilungen modelliert. Hierfür werden von den Autoren Robustheitsmaße basierend auf Erwartungswerten, probabilistischen Grenzwerten sowie probabilistischen Nebenbedingungen beschrieben. Für die durch Fuzzy-Logik modellierten Unsicherheiten werden die Parameter durch Fuzzy-Mengen beschrieben. Das Robustheitsmaß beschreibt den Grad der Zugehörigkeit der Lösung zur Menge der gültigen Lösungen.

Goren & Sabuncuoglu¹⁹⁶ beschreiben für ein Einmaschinenproblem einen auf einer Tabu-Suche basierenden Algorithmus zur Generierung von stabilen und robusten Ablaufplänen unter zufälligen Störeinflüssen wie Maschinenausfällen. Als Robustheits- und Stabilitätsmaß werden zwei Ersatzkriterien für die originalen Zielfunktionen vorgeschlagen, da diese nur sehr aufwendig zu berechnen sind. Die Ersatzkriterien berücksichtigen dabei die Verteilungen der Betriebs- und Reparaturdauern für die Maschine.

Sabuncuoglu & Goren¹⁹⁷ stellen verschiedene Robustheits- und Stabilitätsmaße vor, wobei erstere in zwei Kategorien unterteilt werden: In der ersten Kategorie werden sieben Maße beschrieben, die auf der realisierten Performance (gemessen anhand der verwendeten Zielfunktion) eines Ablaufplans beruhen. Die zweite Kategorie enthält vier Maße, die auf dem Verlust/Regret der realisierten Lösung im Vergleich zur optimalen Lösung

¹⁹²Siehe [KY97], S. 8 ff.

¹⁹³Häufig auch als absolutes Verlust- bzw. Regret-Kriterium bezeichnet, da die Abweichung von der optimalen Lösung für ein Szenario als Verlust angesehen wird.

¹⁹⁴Siehe [Jen01], S. 79 f.

¹⁹⁵Siehe [BS07], S. 3194–3199.

¹⁹⁶[GS08].

¹⁹⁷[SG09].

aufbauen. Für die Stabilitätsmaße werden sechs Maße vorgestellt, die auf der Differenz der Fertigstellungszeitpunkte der Aufträge zweier verschiedener Ablaufpläne beruhen.

Probleme mit sehr konservativen Lösungen in Folge der ausschließlichen Konzentration auf den Worst Case sind in der robusten Optimierung von großer Bedeutung¹⁹⁸ und einer der Kritikpunkte an diesem Konzept.¹⁹⁹ Aus diesem Grund sind mehrere Alternativansätze zu Robustheitsmaßen und Modellierungsansätzen vorgeschlagen worden, die im Folgenden näher erläutert werden.²⁰⁰

Daniels & Carrillo²⁰¹ beschreiben für ein Einmaschinenproblem mit unsicheren Bearbeitungszeiten der Aufträge (abgebildet über Bearbeitungszeitszenarien) einen probabilistischen Ansatz, bezeichnet als „ β -Robust“. Die generierten Ablaufpläne werden anhand der jeweiligen Wahrscheinlichkeit, eine summierte Durchlaufzeit zu erzielen, die ein bestimmtes Anspruchsniveau T nicht überschreitet, evaluiert. Das Ziel der Ablaufplanung ist es, den Ablaufplan, der diese Wahrscheinlichkeit maximiert, zu ermitteln. Dieser Plan wird als β -robuster-Ablaufplan bezeichnet und durch die Anwendung eines Branch-and-Bound-Algorithmus sowie der Verwendung einer Heuristik ermittelt.

Bertsimas & Sim²⁰² behandeln „The Price of Robustness“ und beschreiben Modellierungsmöglichkeiten, um weniger konservative Lösungen für robuste Optimierungsprobleme zu erstellen. Die Autoren erachten die Koeffizienten $a_{i,j}$ der Restriktionsmatrix A eines linearen Programms als unsicher. Die Koeffizienten nehmen dabei Werte aus einer symmetrischen Verteilung im Intervall $[a_{i,j} - \hat{a}_{i,j}, a_{i,j} + \hat{a}_{i,j}]$ an, deren Erwartungswert dem nominalen²⁰³ Wert dieses Koeffizienten entspricht und wobei $\hat{a}_{i,j}$ die Abweichung von diesem beschreibt. Die Autoren nehmen weiterhin an, dass es unwahrscheinlich ist, dass alle Koeffizienten von ihrem nominalen Wert abweichen. Daher wird für jede Restriktion ein Parameter Γ_i eingeführt, der die maximale Anzahl an Koeffizienten angibt, die gleichzeitig von ihrem nominalen Wert abweichen dürfen. Über diesen Parameter kann somit der Grad der Robustheit gesteuert werden.

Snyder & Daskin²⁰⁴ beschreiben ein gegenüber dem Maß der relativen Abweichung bzw. relativem Verlust-/Regret-Kriterium erweitertes Maß der p -Robustheit für Probleme mit einer endlichen Zahl an Szenarien. Eine Lösung wird hierbei als p -robust bezeichnet, wenn ihr relativer Regret für kein Szenario den definierten Parameter p übersteigt. Die Autoren wenden das Robustheitsmaß für zwei *Facility Location*-Probleme (k -Median-Problem und *Uncapacitated Fixed-charge Location*-Problem) an, bei denen die Kundenbedarfe und Transportkosten unsicher sind.

Die Autoren Fischetti & Monaci²⁰⁵ beschreiben einen heuristischen Ansatz zur Modellierung von Unsicherheiten, den sie „Light Robustness“ nennen und der als ein flexibles

¹⁹⁸Siehe [GMT14], S. 472.

¹⁹⁹Siehe [IT14], S. 131.

²⁰⁰Vgl. ebenfalls [GS15], S. 5–13, für eine Übersicht zu verschiedenen Modellierungsansätzen.

²⁰¹[DC97].

²⁰²[BS03]; [BS04].

²⁰³Der nominale Wert eines Koeffizienten ist der Wert, der in einem deterministischen Optimierungsmodell verwendet wird und der unter anderem auf Basis von Messungen, Schätzungen oder Erfahrung ermittelt werden kann.

²⁰⁴[SD06].

²⁰⁵[FM09].

Gegenstück zur robusten Optimierung angesehen werden kann. Der Ansatz basiert zum einen auf der Definition einer maximalen Abweichung von dem optimalen Zielfunktionswert für das nominale Problem²⁰⁶ in Form einer zusätzlichen Restriktion. Zum anderen wird ein Robustheitsziel vorgegeben und durch ein klassisches Framework, wie das von Ben-Tal & Nemirovski²⁰⁷ oder Bertsimas & Sim,²⁰⁸ modelliert; ferner werden Schlupfvariablen eingeführt, um Zulässigkeitsprobleme zu berücksichtigen. Als künstliche Zielfunktion wird die Minimierung der Schlupfvariablen eingeführt. Die Autoren stellen zwei Varianten der Light Robustness vor, wobei die erste auf dem Robustheitsmodell von Bertsimas & Sim aufbaut und die zweite Variante direkt an den Schlupfvariablen ansetzt. Das vorgeschlagene Konzept wird auf Basis von künstlich erzeugten Probleminstanzen für ein ein- bzw. mehrdimensionales Rucksackproblem und ein Portfolioproblem sowie ein reales *Train Timetabling*-Problem evaluiert.

Ben-Tal et al.²⁰⁹ stellen einen als „soft robustness“ bezeichneten Modellierungsprozess vor. Hierbei soll der Erwartungswert einer Auszahlungsfunktion maximiert werden, wobei bezüglich des Wissens der unterliegenden Verteilungsfunktion von Ungewissheit ausgegangen wird. Im Gegensatz zur Light Robustness werden hier verschiedene Grade der Zulässigkeit durch Restriktionen erzwungen und der Grad der Optimalität wird als Zielfunktion verwendet. Der Ansatz wird schließlich auf zwei Portfolio-Probleme angewendet.

Roy²¹⁰ schlägt drei Robustheitsmaße für eine endliche Zahl möglicher Szenarien vor, die auf den drei von Kouvelis & Yu²¹¹ beschriebenen Robustheitsmaßen beruhen. Hierfür werden zwei zusätzliche Parameter w und b definiert, die als Grenzwerte Verwendung finden. Der erste Parameter w gibt einen garantierten Wert an, den der Entscheidungsträger für jedes mögliche Szenario erfüllt sehen möchte. Der Parameter b dient als Grenzwert, der in möglichst vielen Szenarien über- bzw. unterschritten (abhängig davon, ob es sich um ein Maximierungs- oder Minimierungsproblem handelt) werden soll. Über den Grenzwert w lässt sich die Relevanz des Worst-Case-Szenarios abschwächen und somit können Lösungen berücksichtigt werden, die zwar eine schlechtere Worst-Case-Performance besitzen, in der überwiegenden Zahl der Fälle (gemessen über den Parameter b) aber gute Ergebnisse erzielen. Eine Erweiterung, bei der die bw -Robustheit mit Unsicherheitsintervallen verwendet wird, beschreiben Gabrel et al.²¹² für eine robuste Lösung des Problems des kürzesten Weges.

Die Autoren Kalaï et al.²¹³ schlagen ein weiteres Robustheitsmaß basierend auf dem minimax- und dem p -Robustheitskriterium vor, das sie als „lexicographic α -robustness“ bezeichnen. Lösungen werden hierbei auf Basis ihres absteigend sortierten Kostenvektors

²⁰⁶In einem nominalen Problem nimmt jeder unsichere Parameter seinen nominalen Wert an. Dieser entspricht dem Wert des Parameters aus dem deterministischen Optimierungsmodell.

²⁰⁷[BN99].

²⁰⁸[BS04].

²⁰⁹[BBB10].

²¹⁰[Roy10]; [Roy07] sowie [AR10] für eine Betrachtung bei mehreren Zielfunktionen.

²¹¹[KY97].

²¹²[GMW13].

²¹³[KLV12]; [KL07].

für eine endliche Menge von Szenarien, einer theoretischen Idealpunktlösung und einer Relation, die die Autoren „ α -leximax relation“ nennen, evaluiert. Als α -robuste-Lösungen werden die Lösungen bezeichnet, bei denen in jedem betrachteten Szenario die Differenz aus den Kosten der Lösung und der Idealpunktlösung kleiner gleich α ist.

Die vorgenannten Ansätze werden von Iancu & Trichakis²¹⁴ kritisiert, da sie einen Teil der Robustheit für eine evtl. bessere Performance aufgeben. Von den Autoren wird daher ein Vorgehen vorgeschlagen, um Lösungen zu ermitteln, die die gleiche Worst-Case-Szenario-Performance wie die vom Standardframework der robusten Optimierung ermittelten Lösungen besitzen, dabei aber eine bessere Performance für die weiteren Szenarien erzielen. Hierfür weisen die Autoren nach, dass das Standardframework der robusten Optimierung nicht notwendigerweise eine robuste Lösung x ermittelt, die pareto-optimal ist. Eine Lösung wird dabei als pareto-optimal robust angesehen, wenn sie die gleiche Worst-Case-Performance wie x besitzt und zusätzlich von keiner anderen Lösung pareto-dominiert wird, d. h., es gibt keine Lösung mit einer strikt besseren Performance für ein vom Worst-Case-Szenario verschiedenes Szenario. Das Vorgehen wird für ein Portfolio-, ein Inventar- und ein Projektmanagementproblem angewendet.

Eine Erweiterung der Alternativansätze zu Robustheitsmaßen und Modellierungsansätzen auf mehrzielorientierte Optimierungsprobleme wird von Ide & Schöbel²¹⁵ beschrieben. Insgesamt werden drei Ansätze, unter anderem das „Light Robustness“ Konzept, erweitert und mit verschiedenen minmax-robust-effizienten Ansätzen sowie Konzepten basierend auf Ordnungsrelationen verglichen.

Beiträge zu Ablaufplanungsproblemen in Luftfrachtterminals sind nicht bekannt. Im Folgenden werden daher Arbeiten betrachtet, die das Grundproblem der Ablaufplanung für mehrere Maschinen mit unsicheren Bearbeitungszeiten der Aufträge und Teile der für diese Arbeit definierten Anforderungen berücksichtigen und mit exakten oder heuristischen Methoden lösen.

Herrmann²¹⁶ stellt einen auf zwei parallelen Populationen basierenden genetischen Algorithmus zur Lösung von minimax-Optimierungsproblemen vor. Hierbei werden in der einen Population die Lösungen und in der anderen die Szenarien für die Unsicherheiten entwickelt. Die Individuen einer Population werden dabei jeweils auf Basis der Individuen der anderen Population evaluiert und konvertieren somit zu einer robusten Lösung und dem Worst-Case-Szenario. Der Autor testet das Verfahren auf Basis eines identischen Mehrmaschinenproblems mit unsicheren Bearbeitungszeiten der Aufträge.

Anglani et al.²¹⁷ beschreiben eine robuste Ablaufplanung für ein identisches Mehrmaschinenproblem mit sequenzabhängigen Rüstkosten und unter der Annahme unsicherer Bearbeitungszeiten, deren exakte Verteilungen nicht bekannt sind. Die unsicheren Bearbeitungszeiten werden hierfür als Fuzzy-Zahlen modelliert. Die Autoren formulieren ein nicht-lineares gemischt-ganzzahliges Programm (engl. *Non-linear Mixed-integer Program* (NLMIP)), das sie zu einem äquivalenten gemischt-ganzzahligen linearen Programm (engl. *Mixed-integer Linear Program* (MILP)) umformen und als Zielfunktion die

²¹⁴[IT14].

²¹⁵[IS15].

²¹⁶[Her99].

²¹⁷[AGGM05].

Summe der Rüstkosten minimieren. Für größere Probleminstanzen wird eine approximierte Formulierung vorgeschlagen.

Jia & Ierapetritou²¹⁸ stellen für ein *batch*-Ablaufplanungsproblem der chemischen Industrie mit unsicherem Bedarf und unsicheren Bearbeitungszeiten (abgebildet über Szenarien) einen multikriteriellen Optimierungsansatz vor. Als Zielfunktionen werden der maximale Fertigstellungszeitpunkt (engl. *makespan*), die Modellrobustheit (quantifiziert über die unbefriedigte Nachfrage und modelliert durch Schlupfvariablen) sowie die Lösungsrobustheit (quantifiziert über die positive Abweichung des maximalen Fertigstellungszeitpunkts in einem Szenario vom Erwartungswert) betrachtet. Zur Lösung des Modells wird die *Normal Boundary Intersection*-Methode verwendet, um Lösungen zu generieren, die gleichmäßig über die Pareto-Front verteilt sind.

Ranjbar et al.²¹⁹ stellen zwei Branch-and-Bound-Algorithmen für ein identisches Mehrmaschinenproblem vor, bei dem vom Kunden ein Fälligkeitstermin für den spätesten Fertigstellungszeitpunkt aller Aufträge vorgegeben wird. Die Bearbeitungszeiten der einzelnen Aufträge werden als normalverteilte Zufallsvariablen mit bekanntem Erwartungswert und Varianz modelliert. Ziel ist es, einen Ablaufplan zu erzeugen, der den Servicelevel für den Kunden maximiert. Der Servicelevel wird anhand der Wahrscheinlichkeit, dass der Fertigstellungszeitpunkt den Fälligkeitstermin nicht überschreitet, gemessen.

Von den Autoren Xu et al.²²⁰ wird ebenfalls das identische Mehrmaschinenproblem unter der Annahme unsicherer Bearbeitungszeiten, deren Verteilungen nicht bekannt sind, betrachtet. Die Bearbeitungszeiten der Aufträge werden als geschlossene Intervalle modelliert und die Menge der Szenarien ergibt sich als kartesisches Produkt aller Bearbeitungszeitintervalle. Als Zielfunktion wird ein absolutes Verlust-/Regret-Kriterium auf Basis einer *makespan*-Zielfunktion verwendet, um eine Lösung zu ermitteln, die die maximale Abweichung zur jeweils besten Lösung der Szenarien minimiert. Die unendliche Menge möglicher Szenarien wird reduziert, indem die Autoren nachweisen, dass ein Regret-maximierendes Szenario zu einer endlichen Menge von Extrempunktszenarien²²¹ gehört. Zur Lösung des Problems werden zwei exakte, zwei reine Heuristiken (Lokale Suche, Simulated Annealing) sowie vier Mischheuristiken aus den zwei zuvor genannten und einem *pairwise interchange algorithm* beschrieben. Xu et al.²²² verwenden den vorgenannten Ansatz für das uniforme²²³ Mehrmaschinenproblem mit einem absoluten Verlust-/Regret-Kriterium auf Basis der summierten Durchlaufzeit als Zielfunktion.

Beiträge, die Personal- oder Förderkapazitäten als einschränkende Ressource berücksichtigen, werden im Folgenden betrachtet.

²¹⁸[JI07].

²¹⁹[RDL12].

²²⁰[XCLQ13].

²²¹Ein Extrempunktszenario ist ein Szenario, bei dem alle Bearbeitungszeiten der im Ablaufplan eingeplanten Aufträge bzw. alle betrachteten unsicheren Parameter entweder ihren kleinsten oder größten Wert im Intervall annehmen.

²²²[XLC14].

²²³Bei diesem Problem besitzt jede Maschine eine individuelle Produktionsgeschwindigkeit q_j . Die Bearbeitungszeit eines Auftrags unterscheidet sich somit um diesen Faktor für verschiedene Maschinen.

Lau & Zhao²²⁴ beschreiben für ein automatisiertes Luftfrachtterminal ein Ablaufplanungsverfahren, das in einer gemeinsamen Planung das unterschiedliche Fördertechnik-equipment integriert. Für das Schlüssequipment – AGVs und ETVs²²⁵ – wird eine gemeinsame Planung mit Hilfe einer Heuristik zur Minimierung der Wartezeiten der ETVs sowie der Transportzeiten der AGVs durchgeführt.

Chaudhry & Drake²²⁶ verwenden einen genetischen Algorithmus, um für ein identisches Mehrmaschinenproblem die Gesamtverspätung des Ablaufplans zu minimieren. Hierbei wird das Personal als weitere einschränkende Ressource neben den Maschinen betrachtet und direkt innerhalb der Kodierung des GAs berücksichtigt. Ein ähnlicher Ansatz wird von ElMaraghy et al.²²⁷ für einen Job Shop sowie von Costa et al.²²⁸ für ein heterogenes²²⁹ Mehrmaschinenproblem vorgestellt.

Tang et al.²³⁰ stellen ein *Multi-commodity Network Flow*(MCNF)-Modell für ein Luftfrachtterminal vor, in dem sowohl Export- als auch Transferfracht bearbeitet werden. Mit Hilfe des Modells soll die Auslastung des Terminals ausgeglichen werden, um Stausituationen zu vermeiden. Aufbauend auf dem MCNF-Modell wird eine Heuristik für die praktische Umsetzbarkeit entworfen und mittels Simulation validiert.

Joe et al.²³¹ entwerfen ein Kontrollframework zur Steuerung eines komplexen MHS. Als Beispiel nutzen die Autoren ein Luftfrachtterminal, das sie als Netzwerk abbilden. In dem Netzwerk bezeichnen die Knoten Puffer und Fahrzeuge (hier: *Rail-guided Vehicles* (RGVs)), die Kanten die Fahrwege der RGVs und ULDs die verschiedenen Güter, die auf den Strecken zu transportieren sind. Für das Netzwerk wird ein *Multi-commodity Flow Problem* (MCFP) zur Minimierung der benötigten Anzahl der Transporte für die ULDs definiert. Die durch das MCFP berechneten Routen werden zur Konfiguration einer Matrix-basierten diskreten Ereignissteuerung verwendet, die als Echtzeitdispatcher Steuerungsanweisungen an das MHS sendet.

Zäpfel et al.²³² entwickeln für eine Fließfertigung in der Automobilindustrie ein mathematisches Modell (spezifischer ein NLMIP), bei dem sowohl die Zuordnung von Aufträgen zu den Fertigungslinien und die Festlegung von Start- und Endzeiten der Aufträge betrachtet wird als auch die Zuweisung von Personal zu den Arbeitsstationen. Ziel ist es, die Fertigstellungszeit des Systems zu minimieren. Für das Modell wird anschließend eine zweistufige Heuristik vorgeschlagen. Im ersten Schritt erfolgt die Zuweisung der Aufträge zu den Fertigungslinien und die Festlegung der Start- und Endzeiten. Im zweiten Schritt wird das Personal den Arbeitsstationen zugewiesen und die Start- und Endzeiten der Aufträge werden aktualisiert.

²²⁴[LZ06].

²²⁵Die Autoren bezeichnen diese als „stacker cranes“.

²²⁶[CD09].

²²⁷[EPA00].

²²⁸[CCF13].

²²⁹Bei diesem Problem ergibt sich die Bearbeitungszeit eines Auftrags aus der Kombination aus Maschine und Auftrag. Ein Auftrag kann somit für jede Maschine eine unterschiedliche Bearbeitungszeit besitzen.

²³⁰[TNL10].

²³¹[JGL12].

²³²[ZWB12].

Mehravaran & Logendran²³³ beschreiben für eine Fließfertigung ohne Permutation, bei der sowohl Maschinen als auch Personal mit verschiedenen Fähigkeitsleveln als einschränkende Ressourcen betrachtet werden, ein MILP zur Minimierung der gewichteten Fertigstellungszeit sowie der gewichteten Verspätung der Aufträge. Des Weiteren werden drei Metaheuristiken beschrieben, die jeweils aus zwei verschiedenen Ebenen bestehen. Während in der äußeren Ebene ein auf der Tabu-Suche basierender Suchalgorithmus verwendet wird, um die Maschinenbelegung zu ermitteln, wird in der inneren Ebene eine dreistufige Heuristik zur Einplanung des Personals verwendet.

3.3 Ablaufsteuerung und Revision von Ablaufplänen

Die im vorherigen Abschnitt beschriebenen Verfahren finden für eine proaktive Ablaufplanung Verwendung, bei der die Absicherung gegen mögliche Störungen vor der Planausführung stattfindet. Im Folgenden werden ergänzend hierzu reaktive Verfahren zur Revision von Ablaufplänen betrachtet, die während der Planausführung auf Störungen reagieren und den Ablaufplan entsprechend anpassen können.²³⁴ Hierzu werden in Abschnitt 3.3.1 zunächst verschiedene Vorgehensweisen zur Revision von Ablaufplänen vorgestellt. Anschließend wird in Abschnitt 3.3.2 ein Literaturüberblick im Kontext von Mehrmaschinenproblemen gegeben.

3.3.1 Vorgehensweisen zur Revision von Ablaufplänen

Treten während der Planausführung Störungen auf, die nicht den bereits bei der proaktiven Ablaufplanung berücksichtigten stochastischen Einflüssen entsprechen, kann eine Neu- bzw. Umplanung (engl. *rescheduling*) des Ablaufplans notwendig werden. Die Notwendigkeit besteht immer dann, wenn der aktuelle Plan nicht länger ausgeführt werden kann. Weicht die tatsächliche Performance von der geplanten ab, ist eine Umplanung optional und unter Berücksichtigung der entstehenden Kosten vorzunehmen.²³⁵ Die in der Literatur beschriebenen Vorgehensweisen zur Umplanung lassen sich grob in drei verschiedene Kategorien einteilen,²³⁶ die im Folgenden vorgestellt werden.

In der ersten Kategorie (als „lokales Rescheduling“ oder „Planreparatur“ bezeichnet) finden sich einfache Vorgehensweisen, die auf eine schnelle Reparatur des ungültigen Ablaufplans abzielen. Hierzu gehört bspw. die *Right Shift Rule*, bei der alle von der Störung betroffenen Aktivitäten unter Beibehaltung der geplanten Reihenfolge in die Zukunft verschoben werden. Bei der *Right Shift and Jump Rule* kann zusätzlich die Reihenfolge angepasst werden.²³⁷ Die beiden Regeln sind leicht anwendbar, können aber aufgrund der lokalen Anwendung zu einer schlechten Performance des resultierenden Ablaufplans führen.²³⁸

²³³[ML13].

²³⁴Vgl. Abschnitt 2.2.3.

²³⁵Vgl. [VHL03], S. 39 f.

²³⁶Vgl. [KJF10], S. 266; [Bid05], S. 32; [VHL03], S. 54; [SB00], S. 571.

²³⁷Siehe [Bid05], S. 32.

²³⁸Vgl. [KJF10], S. 266.

In die zweite Kategorie fallen Neuplanungsverfahren (engl. *full/total/complete rescheduling*), bei denen der noch nicht ausgeführte Ablaufplan verworfen und unter Berücksichtigung der Störungen neu geplant wird.²³⁹ Hierfür können alle Ablaufplanungsverfahren verwendet werden, die auch für die initiale Ablaufplanung geeignet sind. Die Nachteile dieses Vorgehens sind der Rechenaufwand unter Berücksichtigung der knappen Reaktionszeit im Falle einer Störung sowie die fehlende Planstabilität, sofern diese nicht explizit berücksichtigt wird.²⁴⁰

Zwischen diesen beiden Extremen finden sich partielle Verfahren (engl. *partial rescheduling*), die einen Teil der Aufträge, die direkt oder indirekt durch die Störung betroffen sind, umplanen.²⁴¹ Als Verfahren sind *Affected Operations Rescheduling*²⁴² und *Matchup Scheduling*²⁴³ zu nennen. Ersteres berücksichtigt bei der Umplanung betroffener Aufträge die Planstabilität und -effizienz. Letzteres versucht eine Umplanung der Aufträge bis zu einem gewissen *matchup point* durchzuführen, ab dem der ursprüngliche Ablaufplan fortgesetzt werden kann. Ein generischer Ansatz für partielle Verfahren wird von Kuster et al.²⁴⁴ vorgeschlagen, von den Autoren als „Local Rescheduling“ bezeichnet. Das Verfahren basiert auf der Verwendung eines iterativ größer werdenden Zeitfensters, innerhalb dessen eine Revision des Plans vorgenommen wird.

3.3.2 Literaturüberblick

Vieira et al.²⁴⁵ beschreiben für ein uniformes Mehrmaschinenproblem unter Berücksichtigung verschiedener Auftragsstypen und Rüstzeiten analytische Modelle, die zur Vorhersage der Performance von drei verschiedenen Umplanungsstrategien verwendet werden. Es werden die periodische, die Event-basierte und die hybride Umplanungsstrategie betrachtet. Erstere führt eine Umplanung zu äquidistanten Zeitpunkten durch. Bei der Event-basierten Strategie erfolgt eine Umplanung, wenn ein bestimmter Grenzwert neu eingetrossener Aufträge erreicht worden ist. Die hybride Strategie erfolgt sowohl periodisch als auch bei spezifischen Ereignissen (hier: Maschinenausfall und -reparatur).

Alagöz & Azizoglu²⁴⁶ stellen für ein identisches Mehrmaschinenproblem unter der Einschränkung, dass nicht jede Maschine jeden Auftrag bearbeiten kann, ein Umplanungsproblem vor. Es wird davon ausgegangen, dass zu einem Zeitpunkt nur eine Störung eintritt und dass bis zur Reparatur des Ablaufplans keine weitere auftritt. Der Zeitpunkt und die Dauer der Störung sind darüber hinaus bekannt. Als Performancekriterium wird die summierte Durchlaufzeit und als Stabilitätsmaß die Anzahl der Aufträge, für die ein Maschinenwechsel zwischen dem initialen und dem geänderten Ablaufplan durchgeführt

²³⁹Siehe [KJF10], S. 266.

²⁴⁰Vgl. [KJF10], S. 266; [VHL03], S. 55.

²⁴¹Siehe [VHL03], S. 54 f., und [Bid05], S. 32 ff., für eine ausführliche Übersicht zu verschiedenen Verfahren.

²⁴²Siehe z. B. [AS97].

²⁴³Siehe z. B. [BBMN91].

²⁴⁴[KJF10].

²⁴⁵[VHL00].

²⁴⁶[AA03].

wurde, verwendet. Ein sehr ähnliches Problem betrachten Azizoğlu & Alagöz in einer späteren Publikation. Hier kann jedoch jede Maschine jeden Auftrag bearbeiten. Darüber hinaus wird ein Algorithmus mit polynomieller Laufzeit vorgestellt.²⁴⁷ In den Beiträgen von Ozlen & Azizoglu²⁴⁸ beschreiben die Autoren für ein heterogenes Mehrmaschinenproblem einen Ansatz zur Umplanung. Die beiden Beiträge unterscheiden sich in der Anwendung verschiedener Lösungsalgorithmen.

Branke & Mattfeld²⁴⁹ stellen für das Beispiel eines Job Shops mit dynamisch ankommenden Aufträgen einen Ansatz vor, der sowohl die Gesamtverspätung als auch die Flexibilität²⁵⁰ bei der Ablaufplanung berücksichtigt. Das dynamische Problem wird durch einen rollierenden Zeithorizont in deterministische Teilprobleme zerlegt. Zur Lösung der Teilprobleme testen die Autoren einen randomisierten Prioritätsregeldispatcher sowie einen evolutionären Algorithmus. Die für ein deterministisches Teilproblem durch den evolutionären Algorithmus ermittelten Prioritätsinformationen werden nach dem Ansatz von Bierwirth & Mattfeld²⁵¹ für das jeweils nachfolgende deterministische Teilproblem verwendet, da sich diese zu einem gewissen Grad ähneln. Durch dieses Vorgehen kann die Laufzeit reduziert und evtl. die Lösungsqualität verbessert werden.

Lee et al.²⁵² berücksichtigen für ein Zweimaschinenproblem die Transportkosten und -zeiten, die entstehen, wenn Aufträge nach dem Auftreten einer Störung von der einen zur anderen Maschine verschoben werden müssen. Für verschiedene Zielfunktionen führen die Autoren mögliche Lösungsalgorithmen an und ermitteln die Komplexität der entsprechenden Probleme.

Arnaout²⁵³ beschreibt für ein heterogenes Mehrmaschinenproblem mit stochastischen Bearbeitungs- und Rüstzeiten unter Berücksichtigung von Maschinenausfällen und Eilaufträgen die Reparaturregel *Minimum Weighted C_{max} Difference*. Diese Regel berücksichtigt die Änderung der Bearbeitungs- und Rüstzeiten beim Wechsel eines Auftrags von einer Maschine zu einer anderen sowie die Gesamtzahl aller verschobenen Aufträge. Die Regel wird mit zwei weiteren Regeln auf Basis eines Performance- sowie Stabilitätsmaßes in einer Simulation verglichen.

3.4 Fazit zum Stand der Forschung

Der Stand der Forschung zu den drei für diese Arbeit identifizierten Problemfeldern bildet die Grundlage für das Erstellen einer Ablaufplanung in Luftfrachtterminals unter Berücksichtigung der in Abschnitt 2.3 definierten Anforderungen. Die Analyse der verschiedenen Beiträge zeigte, dass einige der definierten Anforderungen in der Literatur durch geeignete Verfahren bereits berücksichtigt werden. Hinsichtlich der Integration der Einzelverfahren in ein Gesamtkonzept sowie der problemspezifischen Anpassung

²⁴⁷[AA05].

²⁴⁸[OA09]; [OA11].

²⁴⁹[BM05].

²⁵⁰Die Flexibilität wird erhöht, indem frühe Wartezeiten vermieden werden.

²⁵¹[BM99].

²⁵²[LLY06].

²⁵³[Arn14].

einzelner Verfahren und der Erweiterung um praxisrelevante Anforderungen besteht jedoch Handlungsbedarf.

Von den vorgestellten multivariaten statistischen Verfahren zur Klassenbildung entsprechen partitionierende Verfahren den gestellten Anforderungen, insbesondere, da eine hierarchische Aufarbeitung der historischen Betriebsdaten nicht notwendig und eine visuelle Darstellung in Form eines Dendrogramms bei vielen Datensätzen nicht sinnvoll ist. Das Wissen über die Anzahl verschiedener Auftragsarten kann dabei als Information für die Festlegung der gewünschten Clusteranzahl verwendet werden. Zur Berücksichtigung verschiedener Skalenniveaus der Eigenschaften sowie zur Vermeidung einer zu starken Gewichtung von Ausreißern bietet sich darüber hinaus das *k-medoids*-Verfahren mit einer problemspezifisch angepassten Distanzfunktion an. Zur Klassierung neuer Aufträge anhand von charakterisierenden Eigenschaften bietet sich ein generatives Modell zur Klassierung an, da es abgesehen von dem höheren Aufwand zur Modellierung einige Vorteile im Vergleich zu den diskriminativen Modellen bietet. So können beispielsweise als Ausreißer erkannte Aufträge einen Hinweis auf ein weiteres Cluster liefern, das bei der Klassenbildung berücksichtigt werden sollte. Im Hinblick auf die Klassierung neuer Aufträge bei mehr als zwei Clustern und der Bewertung der Wahrscheinlichkeit für eine Clusterzugehörigkeit wird der naive Bayes-Klassifikator verwendet. Dieser ist direkt für *k*-Cluster-Probleme anwendbar und gibt die Clusterzugehörigkeit in Form von A-posteriori-Wahrscheinlichkeiten aus.

Der Literaturüberblick in Abschnitt 3.1.3 zeigte, dass die explizite Einbeziehung von historischen Daten für das hier behandelte Problem nur unzureichend durch bestehende Arbeiten abgedeckt wird. Hier ist eine weitergehende Integration der Datenbasis und des erarbeiteten Informationsstands mit der Ablaufplanung zu leisten.

Im Hinblick darauf, dass es sich bei der hier beschriebenen Ablaufplanung um ein sehr komplexes Problem handelt und für die Planung nur eine begrenzte Zeit zur Verfügung steht, bietet sich die Verwendung von (Meta-)Heuristiken an. Hierbei erwiesen sich die populationsbasierten Metaheuristiken am geeignetsten, da sie für multikriterielle Zielsysteme mehrere Lösungen in einer einzigen Ausführung ermitteln können. Evolutionäre Metaheuristiken sollen in dieser Arbeit verwendet werden, da sie umfangreich erforscht und für Optimierungsprobleme unter Unsicherheit bereits erste Vorarbeiten geleistet wurden. Ablaufplanungsprobleme wurden in der Literatur bereits eingehend untersucht und praxisrelevante Erweiterungen wurden in den letzten Jahren zunehmend betrachtet. Ein Verfahren, das die für diese Arbeit definierten Anforderungen vollständig integriert, ist jedoch nicht bekannt. Für die Berücksichtigung der Robustheit in der Ablaufplanung bietet sich das von Kouvelis & Yu beschriebene absolute Verlust-/Regret-Kriterium an. Zur Steuerung, wie konservativ ein generierter Ablaufplan sein soll, entspricht das *bw*-Robustheitsmaß nach Roy am ehesten der in Abschnitt 2.3.2 definierten Anforderung. Stabilitätsmaße, die bereits geplante Fertigstellungszeitpunkte berücksichtigen, sind den Arbeiten von Jensen sowie Sabuncuoglu & Goren zu entnehmen. Des Weiteren kann das dynamische Planungsumfeld durch die Verwendung einer rollierenden Planung einbezogen werden, wie von Branke & Mattfeld beschrieben. Den Anforderungen nach der Berücksichtigung des Personals und der Förderkapazitäten wird durch Arbeiten in den Bereichen DRCS und MRCS entsprochen.

Der in Abschnitt 3.2.4 aufgeführte Literaturüberblick zeigte, dass keine Arbeiten bekannt sind, die für ein identisches Mehrmaschinenproblem die Zielfunktionen Anzahl säumiger Aufträge, Robustheit und Stabilität sowie alle definierten praxisrelevanten Anforderungen berücksichtigen. Um diese Anforderungen zu erfüllen, ist weitere Arbeit zu leisten.

Für das Problemfeld der Ablaufsteuerung und Revision von Ablaufplänen erscheinen partielle Umplanungsverfahren am vielversprechendsten, da diese dem Trade-off zwischen Performance und Stabilität gerecht werden. Hierzu sind keine neuen Verfahren zu entwickeln, stattdessen ist die Integration in die zuvor beschriebenen Verfahren sowie eine problemspezifische Anpassung zu leisten.

4 Zu leistende Arbeit

Zur Realisierung eines Ablaufplanungs- und -steuerungssystems für die im Rahmen der Umschlagprozesse in Luftfrachtterminals durchzuführenden Frachtabfertigungsaufträge wird im Folgenden beschrieben, welche Arbeit zu leisten ist. Ziel ist es, für jedes der in Abschnitt 2.2 identifizierten Problemfelder und die dort beschriebenen Ziele ein Lösungsverfahren zu erarbeiten und diese Lösungsverfahren zu einem Gesamtsystem zu kombinieren. Hierbei sind die in Abschnitt 2.3 definierten Anforderungen ebenso wie der in Kapitel 3 analysierte Stand der Forschung zu berücksichtigen. Die zu leistende Arbeit wird hierzu in Arbeitspakete zerlegt, die den einzelnen Problemfeldern zugeordnet werden.

Eine konzeptionelle Umsetzung der beschriebenen Arbeitspakete sowie deren Einordnung in ein Gesamtsystem zur Ablaufplanung und -steuerung erfolgt in Kapitel 5. Die für die Arbeitspakete umgesetzten Verfahren werden anschließend in Kapitel 6 validiert, was zugleich das letzte Arbeitspaket darstellt. Der Aufbau beider Kapitel orientiert sich hierbei an den in Abschnitt 2.2 identifizierten Problemfeldern.

4.1 Entwicklung eines Verfahrens zur Datenanalyse der Abfertigungsprozesse

In einem ersten Schritt ist aufbauend auf der vorhandenen Datenbasis über historische Betriebsdaten ein Informationsstand zu erarbeiten, der eine probabilistische Bewertung von Bearbeitungszeiten zukünftiger Aufträge erlaubt. Zur Erreichung dieses Ziels werden für dieses Problemfeld zwei Arbeitspakete definiert.

Ziel des ersten Arbeitspakets ist es, die Daten aus der Datenbasis zu verdichten und für die spätere Ablaufplanung verwertbar aufzubereiten. Eine erste Verdichtung wird durchgeführt, indem historische Aufträge, die sich hinsichtlich ausgewählter Eigenschaften ähneln, zu Clustern zusammengefasst werden (vgl. Abschnitt 2.2.1.2). Hierzu sind aus der Menge der in der Datenbasis erfassten Eigenschaften historischer Frachtabfertigungsaufträge zunächst die relevanten Eigenschaften zu bestimmen, die eine möglichst hohe Korrelation mit der benötigten Bearbeitungszeit eines Auftrags aufweisen. Anschließend ist eine problemspezifische Distanzfunktion zu entwickeln, mit der die Ähnlichkeit bzw. Unähnlichkeit der historischen Aufträge auf Basis der ausgewählten Eigenschaften quantifiziert werden kann. Die Analyse von Clusteringverfahren in Abschnitt 3.1.1 hat das partitionierende Clusteranalyseverfahren *k-medoids* als geeignet identifiziert, um die anschließende Clusteranalyse von historischen Aufträgen durchzuführen. Auf Basis der identifizierten Cluster wird eine weitere Verdichtung der Daten vorgenommen, indem Intervallgrenzen und Häufigkeitsverteilungen der Bearbeitungszeitabweichungen aller

einem Cluster zugeordneten Aufträge ermittelt werden. Die zu berücksichtigende Intervallausdehnung sowie die Granularität der Häufigkeitsverteilungen sollen dabei in Abhängigkeit des Clusters parametrisierbar sein.

In dem zweiten Arbeitspaket wird ein Klassifikator erstellt, der eine Klassierung und damit Zuordnung von einzuplanenden Aufträgen während der Ausführung der Ablaufplanung vornehmen kann. Hierfür wird ein naiver Bayes-Klassifikator verwendet (vgl. Abschnitt 3.1.2), da dieser die Anforderung nach der Ausgabe von Zugehörigkeitswahrscheinlichkeiten erfüllt (vgl. Abschnitt 2.3.1). Für den Klassifikator sind zunächst Trainingsdaten auf Basis der zuvor identifizierten Cluster zu generieren und anschließend ist der Klassifikator mit diesen zu trainieren. Ergebnis einer Klassierung eines einzuplanenden Auftrags ist die für ein Cluster hinterlegte Intervallgrenze sowie Häufigkeitsverteilung der Bearbeitungszeitabweichungen.

4.2 Entwicklung einer multikriteriellen Ablaufplanung der Abfertigungsprozesse

Aufbauend auf den Informationen, die der trainierte Klassifikator über einen einzuplanenden Auftrag zurückliefert, kann in einem zweiten Schritt eine proaktive Ablaufplanung zur Generierung robuster Ablaufpläne, die die Anzahl säumiger Aufträge minimieren, erstellt werden. Hierfür wird dieses Problemfeld in drei Arbeitspakete unterteilt:

In dem ersten Arbeitspaket für dieses Problemfeld wird zunächst unter der Annahme einer statischen Umgebung (vgl. Abschnitt 2.2.2.1) ein konzeptionelles Grundmodell des Ablaufplanungsproblems aufgestellt, das als mathematisches Modell formalisiert wird. Für das Grundmodell wird ein Zielsystem aufgestellt, das die Minimierung der Anzahl säumiger Aufträge sowie die Maximierung der Robustheit des ermittelten Ablaufplans als Zielfunktionen betrachtet. Die Messung der Robustheit basiert hierbei auf dem in Abschnitt 3.2.4 identifizierten absoluten Verlust-/Regret-Kriterium von Kouvelis & Yu in Kombination mit dem *bw*-Robustheitskriterium von Roy, da dieses eine Parametrisierung des gewünschten Anspruchsniveaus erlaubt und somit der Anforderung der nicht expliziten Konzentration auf den Worst-Case-Fall Rechnung trägt (vgl. Abschnitt 2.3.2). Zur Berücksichtigung des dynamischen Planungsumfeldes ist das Grundmodell zu erweitern und in eine rollierende Ablaufplanung zu integrieren. Die definierte Anforderung, einen bereits aktiven und in der Ausführung befindlichen Ablaufplan zu berücksichtigen (vgl. Abschnitt 2.3.2), wird durch die Verwendung des Stabilitätsmaßes nach Jensen beachtet (vgl. Abschnitt 3.2.4). Dieses erweitert das Zielsystem um eine weitere Zielfunktion und das Grundmodell wird entsprechend ergänzt.

Aufgrund der Komplexität des mathematischen Modells und der Anforderung, diese mit dem zu entwickelnden Verfahren zu beherrschen (vgl. Abschnitt 2.3.2), ist im zweiten Arbeitspaket eine Heuristik für dieses Problemfeld zu entwickeln. Die Analyse möglicher Heuristiken für Ablaufplanungsprobleme in Abschnitt 3.2.2 zeigte die Eignung populationsbasierter Metaheuristiken. Diese sind in der Lage, die sich aufgrund der konfliktären Zielfunktionen des Zielsystems ergebenden pareto-optimalen Ablaufpläne

in Form einer Population möglicher Ablaufpläne zu ermitteln und auszugeben. Als spezifisches Verfahren soll hier der NSGA-II Algorithmus, der problemspezifisch angepasst wird, verwendet werden.

Das dritte Arbeitspaket dient schließlich der Betrachtung der Ressource Personal sowie der Förderkapazitäten innerhalb der Ablaufplanung. Hierzu sind die Ablaufpläne, wie in Abschnitt 2.2.2.3 aufgezeigt, durch eine Personaleinsatzplanung zu ergänzen. Mit der Berücksichtigung der Förderkapazitäten (vgl. Abschnitt 2.2.2.4) sollen die Ablaufpläne schließlich derart gestaltet werden, dass es zu keinen Überlastungen der innerbetrieblichen Transportsysteme kommt.

4.3 Entwicklung eines Verfahrens zur Ablaufsteuerung der Abfertigungsprozesse

Die durch den vorherigen Schritt erstellten Ablaufpläne bzw. der aus diesen ausgewählte umzusetzende Ablaufplan muss in einem dritten Schritt durch ein Ablaufsteuerungssystem ausgeführt werden, das die Planausführung überwacht. Für dieses Problemfeld wird ein Arbeitspaket definiert.

In diesem Arbeitspaket ist eine Ablaufsteuerung zu entwerfen, die Abweichungen und Ereignisse feststellen kann, die eine Anpassung des aktiven Ablaufplans notwendig machen. Für die Anpassung sollen Reparaturmechanismen zur Verfügung gestellt werden, die unter Berücksichtigung des bereits aktiven Ablaufplans lokale Änderungen vornehmen können. Ist dies nicht möglich, soll die Möglichkeit bestehen, die Ablaufplanung aus dem vorherigen Schritt für eine Um- oder Neuplanung aufzurufen.

5 Konzeption

In diesem Kapitel erfolgt die Konzeption des Ablaufplanungs- und -steuerungssystems für die Frachtabfertigungsprozesse in Luftfrachtterminals zur Lösung der vorgestellten Problemstellung und unter Berücksichtigung der in Kapitel 4 identifizierten zu leistenden Arbeit. In Abschnitt 5.1 werden die zu entwickelnden Verfahren zunächst in eine Architektur in Form von Modulen eingeordnet und ihr Zusammenspiel erläutert. Die Verfahren zur Analyse historischer Betriebsdaten und den Aufbau des benötigten Informationsstands in Form eines Klassifikators zur Klassierung neuer Aufträge werden in Abschnitt 5.2 entwickelt. Darauf aufbauend wird das Ablaufplanungsproblem in Abschnitt 5.3 in einem mathematischen Modell formalisiert und auf Basis dessen wird die Umsetzung einer evolutionären Heuristik zur proaktiven Ablaufplanung erläutert. Besondere Beachtung findet hierbei die dynamische und stochastische Systemumgebung sowie die Berücksichtigung von Personal- und Förderkapazitätsrestriktionen. Abschließend wird die Ablaufsteuerung in Abschnitt 5.4 dargestellt.

5.1 Komposition des Gesamtsystems in einer Architektur

Im Folgenden erfolgt die Komposition einer Architektur, in der die in Kapitel 4 erläuterten Problemfelder berücksichtigt werden. Die Architektur ist in insgesamt drei Ebenen unterteilt, wobei jede Ebene weiter in Module untergliedert wird. Die Module der einzelnen Schichten adressieren dabei die spezifischen Anforderungen der Problemfelder, indem in diesen Verfahren implementiert werden, die eine Lösung der aufgezeigten Probleme ermöglichen.

In Abschnitt 5.1.1 werden hierzu zunächst die einzelnen Ebenen und die in diesen enthaltenen Module erläutert. Anschließend wird in Abschnitt 5.1.2 dargelegt, wie die einzelnen Module zusammenarbeiten, um ein Ablaufplanungs- und -steuerungssystem zu realisieren.

5.1.1 Aufbau der Architektur

Abbildung 5.1 gibt eine Übersicht über den Aufbau der Architektur. Die Unterteilung erfolgt in die reaktive, proaktive und adaptive Ebene. Die reaktive Ebene adressiert die in Abschnitt 2.2.3 aufgezeigten Probleme im Zusammenhang mit der Ablaufsteuerung und Revision von Ablaufplänen. Die zweite (proaktive) Ebene behandelt die in Abschnitt 2.2.2 dargestellten Herausforderungen, die für eine multikriterielle Ablaufplanung unter Unsicherheit bestehen. Die adaptive Ebene schließlich berücksichtigt die in Abschnitt 2.2.1

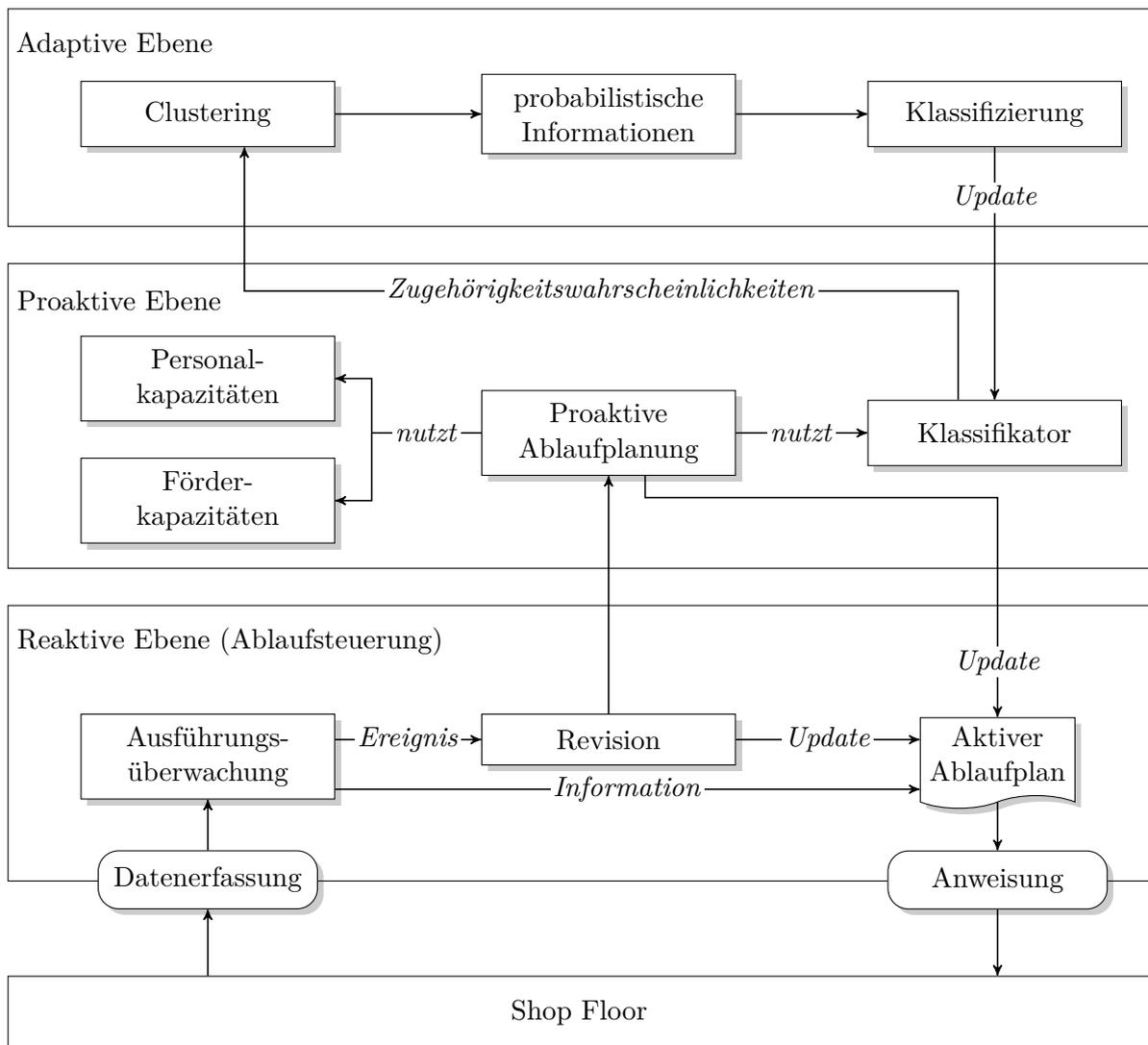


Abbildung 5.1: Übersicht über die Architektur (in Anlehnung an eine Architektur für selbst-adaptive Systeme: [Box11], S. 48)

aufgezeigten Probleme bei der Datenanalyse der Abfertigungsprozesse und Erstellung eines Informationsstands.

Der ersten Ebene der Architektur sind zwei Module zugeordnet. Das erste Modul, *Ausführungsüberwachung*, dient der Überwachung des in der Ausführung befindlichen Ablaufplans und dem Feststellen von Ereignissen, die bspw. über die mobile Datenerfassung gemeldet bzw. aus den übermittelten Daten abgeleitet werden. Das Modul *Revision* führt lokale Änderungen am Ablaufplan durch, sofern diese den Gesamtplan nicht beeinflussen. Die Module der reaktiven Ebene werden in Abschnitt 5.4 dargestellt.

Die mittlere Ebene besteht aus insgesamt vier Modulen. Das Modul *Klassifikator* dient dazu, für einzuplanende Aufträge mögliche Bearbeitungszeiten zu ermitteln. Die Erstellung der proaktiven Ablaufpläne wird im Modul *Proaktive Ablaufplanung* geleistet.

Die Berücksichtigung der Personal- sowie Förderkapazitäten in der Ablaufplanung erfolgt in den gleichnamigen Modulen. Diese Module und die umgesetzten Verfahren werden in Abschnitt 5.3 ausführlich beschrieben.

Schließlich sind der adaptiven Ebene drei Module zugeordnet. Das Clusteringmodul übernimmt die Aufgabe, die historischen Betriebsdaten zu Clustern gleichartiger Aufträge aufzubereiten. Aus den Clustern werden im Modul *probabilistische Informationen* Häufigkeitsverteilungen und Intervalle der im Cluster enthaltenen Aufträge ermittelt. Das Modul *Klassifizierung* dient als letztes Modul der Ebene dazu, den Klassifikator für die zweite Ebene zu erstellen, mit dem Klassierungen neuer Aufträge durchgeführt werden können. Die adaptive Ebene und die enthaltenen Module werden in Abschnitt 5.2 erläutert.

5.1.2 Ablaufprozess innerhalb der Architektur

Die Architektur mit den enthaltenen Modulen dient dazu, den Prozess der Frachtabfertigung im System „Luftfrachtterminal“ zu regeln, indem Anweisungen in Form der Ablaufpläne erstellt werden, die auf eine möglichst umfassende Zielerreichung (gemessen anhand der Anzahl säumiger Aufträge) abzielen. Die aus dem System über die Datenerfassung zurückgelieferten Informationen ermöglichen die Adaption der innerhalb der Architektur ermittelten Anweisungen. Hierfür ist innerhalb der Architektur ein Ablaufprozess, der die einzelnen Ebenen und in diesen die Module miteinander verbindet, notwendig. Dieser Prozess wird im Folgenden erläutert.

Die erste Ebene der Architektur dient in erster Linie der Ablaufsteuerung der Abfertigungsprozesse unter Berücksichtigung des aktiven Ablaufplans. Hierzu werden zu den im Ablaufplan vorgegebenen Zeiten die entsprechenden Aufträge auf den geplanten Maschinen gestartet. Die mobile Datenerfassung ermöglicht einen Abgleich der tatsächlich realisierten Zeiten mit den durch den Ablaufplan vorgegebenen. Werden keine Störungen festgestellt, verläuft die Ausführung auf dieser Ebene nach den Anweisungen des Plans. Auf eine Information (z. B. der Fertigstellung eines Auftrags) erfolgt eine Reaktion in Form bspw. der Anweisung, einen weiteren Auftrag zu starten, der durch den Ablaufplan vorgegeben wird. Im Falle von Ereignissen/Störungen (z. B. der Ankunft eines Eilauftrags) kann der Ablaufplan fallweise lokal repariert werden oder es erfolgt eine Neu- bzw. Umplanung unter Verwendung der zweiten Ebene der Architektur. Die zweite Ebene wird ebenfalls aufgerufen, wenn seit der letzten Ablaufplanung ein definierter Zeitraum verstrichen ist und der aktuelle Ablaufplan ergänzt werden soll.

Die mittlere Ebene wird für die proaktive Ablaufplanung verwendet. Diese Ebene wird einerseits in regelmäßigen Abständen aktiv, um neu hinzugekommene Aufträge in den in der Ausführung befindlichen Ablaufplan zu integrieren. Andererseits wird fallweise eine Revision des aktuellen Ablaufplans vorgenommen. Über den Klassifikator werden zunächst probabilistische Informationen über die Bearbeitungszeiten der einzuplanenden Aufträge ermittelt, die anschließend in dem Modul für die proaktive Ablaufplanung verwendet werden, um eine Menge pareto-optimaler Ablaufpläne zu erstellen. Der ausgewählte Ablaufplan ersetzt schließlich den in der Ausführung befindlichen Plan auf der reaktiven Ebene.

Die dritte Ebene dient der Adaption des auf der mittleren Ebene verwendeten Klassifikators. Dieser wird zum einen angepasst, wenn auf der mittleren Ebene festgestellt wird, dass die erstellten Klassierungen geringe Zugehörigkeitswahrscheinlichkeiten aufweisen. Zum anderen erfolgt eine Anpassung in regelmäßigen Abständen, um die seit der letzten Erstellung hinzugekommenen Betriebsdaten zu berücksichtigen. Über das Clusteringmodul werden hierzu zunächst Cluster gleichartiger Aufträge ermittelt, für die anschließend die probabilistischen Informationen der Bearbeitungszeiten erstellt werden. Schließlich erfolgt die Erstellung des Klassifikators, der den auf der mittleren Ebene verwendeten Klassifikator ersetzt.

5.2 Erstellung des benötigten Informationsstands aus historischen Betriebsdaten

In diesem Abschnitt erfolgt die Konzeption eines Verfahrens, um aus den zur Verfügung stehenden historischen Betriebsdaten über Abfertigungsaufträge Informationen zu gewinnen, die eine probabilistische Bewertung von Bearbeitungszeiten zukünftiger Aufträge erlauben.¹ Hierzu wird im Abschnitt 5.2.1 zunächst ein Verfahren konzipiert, dass Cluster ähnlicher Aufträge bildet. Hierauf aufbauend wird in Abschnitt 5.2.2 ein Klassifikator entwickelt, der neu einzuplanende Aufträge im Rahmen der Ablaufplanung einem Cluster zuordnen kann und somit eine probabilistische Bewertung der zu erwartenden Bearbeitungszeit erlaubt.

5.2.1 Clusterbildung der Abfertigungsaufträge

Durch die Betriebsdatenerfassung liegt über vergangene Frachtabfertigungsaufträge eine Vielzahl von Informationen über diese Aufträge vor (vgl. Abschnitt 2.2.1), insbesondere die realisierten Bearbeitungszeiten sind dabei in diesem Abschnitt von Interesse. Ein historischer Frachtabfertigungsauftrag wird als Tripel $ha = (\vec{e}_{ha}, \hat{\tau}_{ha}, \tau_{ha})$ definiert. Dieses Tripel besteht aus den folgenden drei Komponenten:

- \vec{e}_{ha} : Ein d -dimensionaler Eigenschaftsvektor $\vec{e}_{ha} = (e_1, \dots, e_d)$, der den Auftrag ha charakterisiert. Der Wert der p -ten Eigenschaft wird als e_p mit $1 \leq p \leq d$ bezeichnet.
- $\hat{\tau}_{ha}$: Die geschätzte² Bearbeitungszeit für den Auftrag vor dessen Bearbeitung.
- τ_{ha} : Die realisierte Bearbeitungszeit für den Auftrag.

Die Menge $\mathcal{A}^H = \{ha_1, \dots, ha_n\}$ ergibt sich aus allen Frachtabfertigungsaufträgen bzw. einer Teilmenge³ der n historischen Frachtabfertigungsaufträge.

¹Auszüge dieses Kapitels wurden bereits in [BLW14] veröffentlicht.

²Für eine Beschreibung, wie diese Schätzung durchgeführt wird, vgl. Abschnitt 2.2.1.1.

³Da aufgrund der vielen am Tag zu bearbeiten Frachtabfertigungsaufträge von einer umfangreichen Datenbasis auszugehen ist, ist die Menge \mathcal{A}^H entsprechend zu wählen, sodass das Verfahren zur Clusteranalyse (vgl. Abschnitt 5.2.1.3) dieses handhaben kann.

Für Aufträge aus \mathcal{A}^H , die sich hinsichtlich ihres Eigenschaftsvektors \vec{e}_{ha} ähneln, ist anzunehmen, dass die entsprechenden realisierten Bearbeitungszeiten ebenfalls ähnlich sind. Diese Aufträge bilden somit Cluster ähnlicher Aufträge innerhalb der Menge \mathcal{A}^H . Im Gegensatz zu der später stattfindenden Klassifikatorentwicklung (vgl. Abschnitt 5.2.2), sind zu diesem Zeitpunkt die möglichen Cluster ähnlicher Aufträge jedoch noch nicht bekannt und müssen zunächst durch eine Clusteranalyse ermittelt werden.

Für die Durchführung der Clusteranalyse ist zunächst zu definieren, welche Eigenschaften verwendet werden sollen, die einen Auftrag charakterisieren. Die Auswahl geeigneter Eigenschaften erfolgt in Abschnitt 5.2.1.1. Anschließend ist eine Distanzfunktion zu entwickeln, die anhand der Eigenschaften zweier Aufträge eine Quantifizierung der Ähnlichkeit vornehmen kann. Aufbauend auf den definierten Eigenschaften und der Distanzfunktion kann anschließend eine Clusteranalyse durchgeführt werden, die die Menge \mathcal{A}^H in eine Menge $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ von k disjunkten Clustern unterteilt. Für die so gefundenen Cluster werden anschließend Häufigkeitsverteilungen und Intervallgrenzen der Bearbeitungszeitabweichungen der enthaltenen Aufträge ermittelt.

5.2.1.1 Auftragsbeschreibende Eigenschaften

Für die historischen Aufträge $ha \in \mathcal{A}^H$ sind repräsentative Eigenschaften zu bestimmen, anhand derer ein Auftrag möglichst genau charakterisiert werden kann. Die Eigenschaften sollten dabei so gewählt werden, dass eine Änderung im Wert e_p einer Eigenschaft p mit einer Änderung der realisierten Bearbeitungszeit τ_{ha} des Auftrags einhergeht, d. h., es sollten nur solche Eigenschaften betrachtet werden, die einen direkten Einfluss auf die Bearbeitungszeit eines Auftrags ausüben.

Im Folgenden werden die in dieser Arbeit eingesetzten Eigenschaften näher beschrieben. Dabei gelten diese sowohl für Aufbau-, Abbau- sowie Transferaufträge.

Eigenschaft 1: Typ des Auftrags

Diese Eigenschaft beschreibt den Typ des Auftrags (Auf- und Abbau- sowie Transferauftrag). Aufbau- und Transferaufträge weisen eine höhere Komplexität und damit tendenziell einen höheren Zeitbedarf als reine Abbauaufträge auf, da bei Ersteren die Fracht anhand ihrer Charakteristika (Ausmaße, Gewicht etc.) in einem Luftfrachtcontainer oder auf einer Luftfrachtpalette konsolidiert werden muss. Bei der Konsolidierung sind Restriktionen hinsichtlich der Lastverteilung und im Falle der Paletten der Kontur zu beachten, die bei reinen Abbauaufträgen entfallen.

$$e_1 = \begin{cases} 0, & \text{wenn Abbauauftrag} \\ 1, & \text{wenn Aufbauauftrag} \\ 2, & \text{wenn Transferauftrag} \end{cases}$$

Eigenschaft 2: Typ des Ladehilfsmittels

Diese Eigenschaft beschreibt den Typ des Ladehilfsmittels. Unterschieden wird hierbei nach den zwei grundlegenden Typen Luftfrachtcontainer

und -palette (vgl. Abschnitt 2.1.2). Für Luftfrachtpaletten ist von einem anderen Zeitbedarf als bei den Luftfrachtcontainern auszugehen, da die Paletten nach dem Aufbau/Abbau zunächst mit Netzen zu sichern bzw. diese zu entfernen sind, wohingegen dieser Schritt bei Containern entfällt. Eigenschaft 2 kann somit binär definiert werden:

$$e_2 = \begin{cases} 0, & \text{wenn Ladehilfsmittel Luftfrachtcontainer} \\ 1, & \text{wenn Ladehilfsmittel Luftfrachtpalette} \end{cases}$$

Eigenschaft 3: Volumen der Flugzeugladeeinheit

Das Volumen der Flugzeugladeeinheit fl_{ha} , die zu Auftrag ha gehört, gibt Aufschluss darüber, mit welchem Aufwand bei der Bearbeitung zu rechnen ist. So sind Ladeeinheiten mit einem größeren Volumen⁴ tendenziell mit einem größeren Arbeitsaufwand verbunden.

$$e_3 = \text{vol}(fl_{ha})$$

Eigenschaft 4: Volumen der zu bearbeitenden Fracht

Das Volumen der zu bearbeitenden Fracht ergibt sich als Summe der Volumina $\text{vol}(f)$ der einzelnen Frachtstücke aus der Menge \mathcal{F}_{ha} der Frachtstücke, die zu einem Auftrag ha gehören. Ladeeinheiten, die eine geringere Auslastung haben, benötigen bei einem Aufbauauftrag unter Umständen weniger Zeit, da für die Ausrichtung der Frachtstücke innerhalb der Ladeeinheit mehr Freiheiten zur Verfügung stehen, als dies bei einer höheren Auslastung der Fall wäre.

$$e_4 = \sum_{f \in \mathcal{F}_{ha}} \text{vol}(f)$$

Eigenschaft 5: Anzahl zu bearbeitender Frachtstücke

Die Anzahl zu bearbeitender Frachtstücke gibt Aufschluss darüber, welcher Aufwand bei der Bearbeitung der Fracht entsteht. Viele kleine und lose Frachtstücke können bspw. einen höheren Umladeaufwand bedeuten als einige wenige größere, dafür aber z. B. auf einer Euro-Palette gebündelte Frachtstücke.

$$e_5 = |\mathcal{F}_{ha}|$$

Eigenschaft 6: Anteil der Fracht, die mit einem Hilfsmittel bearbeitet werden muss

Diese Eigenschaft beschreibt den prozentualen Anteil der Frachtstücke, die aufgrund ihres Gewichts $\text{gew}(f)$, das einen bestimmten Grenzwert g_{\max}^{mh} überschreitet, mit einem Hilfsmittel (hier insbesondere Gabelstapler) bearbeitet werden müssen. Es bezeichne $\mathcal{F}_{ha}^{\text{mh}}$ die Menge der Frachtstücke

⁴Das Volumen wird hierbei über den SAE Aerospace Standard (AS) AS1825 berechnet (siehe [AS1825]).

$\mathcal{F}_{ha}^{\text{mh}} = \{f \mid \text{gew}(f) > g_{\text{max}}^{\text{mh}} \wedge \forall f \in \mathcal{F}_{ha}\}$, auf die diese Eigenschaft zutrifft. Dann kann Eigenschaft 6 wie folgt definiert werden:

$$e_6 = \frac{|\mathcal{F}_{ha}^{\text{mh}}|}{|\mathcal{F}_{ha}|}$$

Eigenschaft 7: Anteil der Fracht mit *Special Handling Code* (SHC)

Fracht, die durch einen oder mehrere SHC gekennzeichnet ist, unterliegt im Vergleich zu normaler Fracht weiteren Restriktionen, wie z. B. einer besonderen Ladungssicherung, und nimmt somit bei der Bearbeitung tendenziell mehr Zeit in Anspruch als normale Fracht. Bezeichne $\mathcal{F}_{ha}^{\text{SHC}}$ die Menge der Fracht mit SHC-Kennzeichnung. Eigenschaft 7 ist dann folgendermaßen definiert:

$$e_7 = \frac{|\mathcal{F}_{ha}^{\text{SHC}}|}{|\mathcal{F}_{ha}|}$$

Eigenschaft 8: Anteil der SHC-Fracht, die mit einem Hilfsmittel bearbeitet werden muss Frachtstücke, die sowohl eine SHC Kennzeichnung besitzen als auch mit einem Hilfsmittel bearbeitet werden müssen, nehmen bei der Bearbeitung die längste Zeit in Anspruch. Der Anteil dieser Fracht an der Gesamtfracht wird über diese Eigenschaft abgebildet:

$$e_8 = \frac{|\mathcal{F}_{ha}^{\text{SHC}}| \cap |\mathcal{F}_{ha}^{\text{mh}}|}{|\mathcal{F}_{ha}|}$$

Eigenschaft 9: Anzahl an benötigten Ladehilfsmitteln aus dem Sendungslager

Die Menge \mathcal{LS}_{ha} definiere die Ladeeinheiten (hier: Gitterboxen) im Sendungslager, in denen sich die Fracht für den Aufbauauftrag ha befindet. Dann beschreibt diese Eigenschaft die Anzahl der Ladeeinheiten, die für den Aufbauauftrag verwendet werden. Für Abbauaufträge beschreibt e_9 die Anzahl der Ladehilfsmittel des Sendungslagers, die verwendet werden, um die Fracht der Flugzeugladeeinheit zu vereinzeln.

$$e_9 = |\mathcal{LS}_{ha}|$$

5.2.1.2 Distanzfunktion für Aufträge

Auf Basis der im vorherigen Abschnitt beschriebenen Eigenschaften eines Auftrags soll im Folgenden eine Distanzfunktion $\text{dist}(ha_i, ha_j)$ entwickelt werden, die es erlaubt, die Ähnlichkeit bzw. Unähnlichkeit zweier historischer Aufträge ha_i und ha_j anhand ihrer Eigenschaftsvektoren \vec{e}_{ha_i} und \vec{e}_{ha_j} zu quantifizieren. Diese Distanzfunktion wird später in Abschnitt 5.2.1.3 für die Durchführung der Clusteranalyse der Menge historischer Aufträge \mathcal{A}^H verwendet.

Die beschriebenen Eigenschaften weisen unterschiedliche Wertebereiche auf, die sich negativ auf die Distanzberechnung von Aufträgen auswirken können. So wird bei einigen

Distanzfunktionen den Eigenschaften mit einem größeren Wertebereich implizit ein höheres Gewicht zugewiesen als solchen, auf die dies nicht zutrifft.⁵ Diese Eigenschaften würden somit die Berechnung der Distanz dominieren.

Daher wird jede Eigenschaft zunächst normalisiert und somit in einen einheitlichen Wertebereich überführt. Hierfür bezeichne e_p^* den nicht normalisierten Wert und nf_p einen Normalisierungsfaktor für die p -te Eigenschaft. Der normalisierte Wert e_p der p -ten Eigenschaft ergibt sich dann wie folgt:

$$e_p = nf_p \cdot e_p^* \quad (p = 1, \dots, d)$$

Für die Bestimmung der Normalisierungsfaktoren nf_p sei $e_{ha,p}$ der Wert der p -ten Eigenschaft von Auftrag ha und sei

$$\mathcal{EU}_p = \{e_{ha,p} \mid \forall ha \in \mathcal{A}^H\} \quad (p = 1, \dots, d)$$

die Menge der Eigenschaftswerte der Eigenschaft p aller historischen Aufträge $ha \in \mathcal{A}^H$. Für die beschriebenen Eigenschaften ergibt sich somit die Menge \mathcal{NF} von Normalisierungsfaktoren:⁶

$$\mathcal{NF} = \left\{ 1, 1, \frac{1}{\max(\mathcal{EU}_3)}, \frac{1}{\max(\mathcal{EU}_4)}, \frac{1}{\max(\mathcal{EU}_5)}, 1, 1, 1, \frac{1}{\max(\mathcal{EU}_9)} \right\} \quad (5.1)$$

Des Weiteren werden Gewichtungsfaktoren eingeführt, die eine Gewichtung der entsprechenden Eigenschaft innerhalb der Distanzfunktion erlauben. Hierfür wird die Menge \mathcal{W} definiert, die für jede Eigenschaft einen Gewichtungsfaktor w_p enthält:

$$\mathcal{W} = \{w_1, \dots, w_d\} \quad , \text{ wobei } \sum_{p=1}^d w_p = 1 \quad (5.2)$$

Auf Basis der zuvor definierten Normalisierungsfaktoren \mathcal{NF} und Gewichtungsfaktoren \mathcal{W} kann nun die Distanzfunktion $\text{dist}: \mathcal{A}^H \times \mathcal{A}^H \rightarrow \mathbb{R}$ definiert werden, die für $ha_i, ha_j, ha_l \in \mathcal{A}^H$ den folgenden Anforderungen genügen muss:⁷

1. Nichtnegativität: $\text{dist}(ha_i, ha_j) \geq 0$
2. Identität: $\text{dist}(ha_i, ha_j) = 0$ gdw. $e_{ha_i,p} = e_{ha_j,p}$ ($p = 1, \dots, d$)
3. Symmetrie: $\text{dist}(ha_i, ha_j) = \text{dist}(ha_j, ha_i)$
4. Dreiecksungleichung: $\text{dist}(ha_i, ha_l) \leq \text{dist}(ha_i, ha_j) + \text{dist}(ha_j, ha_l)$

⁵Vgl. [JD88], S. 24; [WM97], S. 4.

⁶Für die Eigenschaften 6, 7 und 8 ist keine Normalisierung notwendig, da die Eigenschaften bereits auf das Intervall $[0, 1]$ beschränkt sind. Die Eigenschaften 1 und 2 werden über die Funktion (5.4) und damit in der Distanzfunktion (5.5) gesondert berücksichtigt.

⁷Vgl. [Ye03], S. 290; [Lar05], S. 99.

Die einzelnen Komponenten eines Vektors \vec{e}_{ha} , der einen Auftrag ha charakterisiert, enthalten Werte aus unterschiedlichen Dimensionen und weisen unterschiedliche Einheiten auf (z. B. Volumen für Eigenschaft 3 oder Prozent für Eigenschaft 6). Für zwei Vektoren \vec{e}_{ha_i} und \vec{e}_{ha_j} mit $i \neq j$ und $ha_i, ha_j \in \mathcal{A}^H$ gilt jedoch, dass die Komponenten $e_{ha_i,p}$ und $e_{ha_j,p}$ mit $1 \leq p \leq d$ die gleichen Einheiten besitzen und Elemente der gleichen Dimension sind. Daher soll für die numerischen Elemente des Eigenschaftsvektors als Distanzfunktion der Betrag⁸ der paarweisen Differenz gebildet werden:

$$\text{dist}_{i,j}^{(p)} = |e_{ha_i,p} - e_{ha_j,p}| \quad (5.3)$$

Für die nominale Eigenschaft 1 und die binäre Eigenschaft 2 wird lediglich auf Gleichheit geprüft:

$$\text{dist}_{i,j}^{(p)} = \begin{cases} 0, & \text{wenn } e_{ha_i,p} = e_{ha_j,p} \\ 1 & \text{in allen anderen Fällen} \end{cases} \quad (5.4)$$

Die vollständige Distanzfunktion ergibt sich aus der gewichteten Summe über die normierten Eigenschaften und je nach Eigenschaftstyp Gleichung (5.3) oder (5.4):

$$\text{dist}(ha_i, ha_j) = \sum_{p=1}^d w_p \cdot (nf_p \cdot \text{dist}_{i,j}^{(p)}), \quad (5.5)$$

mit der die Distanz bzw. Unähnlichkeit zweier Aufträge ha_i und ha_j , die über ihre Eigenschaftsvektoren \vec{e}_{ha_i} und \vec{e}_{ha_j} charakterisiert sind, berechnet werden kann. Je ähnlicher sich zwei Aufträge dabei sind, desto kleiner ist die berechnete Distanz, die im Fall der Gleichheit der Eigenschaftswerte (siehe Anforderung 2 für Distanzfunktionen) den Wert 0 und im Fall der maximalen Unähnlichkeit den Wert 1 annimmt.

5.2.1.3 Verfahren zur Clusteranalyse von Aufträgen

In diesem Abschnitt wird das Verfahren zur partitionierenden Clusteranalyse der Menge historischer Aufträge \mathcal{A}^H beschrieben. Dieses Verfahren verwendet die im vorherigen Abschnitt dargestellte Distanzfunktion und dient dazu, jeden historischen Auftrag $ha \in \mathcal{A}^H$ anhand der Ausprägung seiner beschreibenden Eigenschaften genau einem von k Clustern $\mathcal{C}_1, \dots, \mathcal{C}_k$ zuzuordnen. Die Menge aller Cluster ist eine Partition von \mathcal{A}^H und wird mit der Menge $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ beschrieben. Für diese gilt⁹

$$\bigcup_{i=1}^k \mathcal{C}_i = \mathcal{C} = \mathcal{A}^H$$

und

$$\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \quad (i = 1, \dots, k; j = 1, \dots, k; i \neq j).$$

⁸Aufgrund von Anforderung 1 für Distanzfunktionen.

⁹Vgl. [KP96], S. 439.

Für die Aufträge $ha \in \mathcal{C}_i$ eines Clusters \mathcal{C}_i gilt, dass sie unter der verwendeten Distanzfunktion eine größere Ähnlichkeit zu den Aufträgen des selben Clusters aufweisen¹⁰ als zu den Aufträgen aller anderen Cluster.¹¹

Die Datengrundlage der Clusteranalyse bildet die Menge der n historischen Aufträge \mathcal{A}^H . Für jeden historischen Auftrag $ha \in \mathcal{A}^H$ liegen die in Abschnitt 5.2.1.1 beschriebenen Eigenschaften in Form eines d -dimensionalen Eigenschaftsvektors \vec{e}_{ha} vor. Damit wird die Menge \mathcal{A}^H als $n \times d$ -Matrix PM modelliert, wobei jede Zeile der Matrix ein Muster und jede Spalte eine Eigenschaft definiert.¹²

$$PM = [\vec{e}_{ha_1}, \vec{e}_{ha_2}, \dots, \vec{e}_{ha_n}]^T = \begin{pmatrix} e_{ha_1,1} & e_{ha_1,2} & \dots & e_{ha_1,d} \\ e_{ha_2,1} & e_{ha_2,2} & \dots & e_{ha_2,d} \\ \vdots & \vdots & \ddots & \vdots \\ e_{ha_n,1} & e_{ha_n,2} & \dots & e_{ha_n,d} \end{pmatrix}$$

Auf Basis der Matrix PM kann anschließend die $n \times n$ -Distanzmatrix aufgebaut werden. Hierfür werden je zwei historische Aufträge ha_i und ha_j mit $i \neq j$ durch Gleichung (5.5) verglichen und das Ergebnis dm_{ha_i,ha_j} wird in der Matrix DM gespeichert.¹³ Aufgrund der Anforderungen 2 und 3 der Distanzfunktion handelt es sich um eine entlang der Hauptdiagonalen symmetrische Matrix, in der jeder Auftrag zu sich selbst die Distanz 0 besitzt. Die so erzeugte Matrix dient anschließend als Eingabe für das Verfahren zur Clusteranalyse.

$$DM = \begin{pmatrix} 0 & dm_{ha_1,ha_2} & dm_{ha_1,ha_3} & \dots & dm_{ha_1,ha_n} \\ & 0 & dm_{ha_2,ha_3} & \dots & dm_{ha_2,ha_n} \\ & & 0 & \ddots & \vdots \\ & & & 0 & dm_{ha_{n-1},ha_n} \\ & & & & 0 \end{pmatrix}$$

Zur Durchführung der Clusteranalyse müssen die einem Cluster $\mathcal{C}_i \in \mathcal{C}$ zugeordneten Aufträge $ha_j \in \mathcal{C}_i$ durch ein Clusterzentrum z_i , welches durch den Eigenschaftsvektor \vec{e}_{z_i} charakterisiert wird, repräsentiert werden. Die Menge aller Clusterzentren wird mit \mathcal{Z} beschrieben. Anhand eines Clusterzentrums z_i und auf Basis der Distanzfunktion (vgl. Gleichung (5.5)) wird bestimmt, ob ein Auftrag zu dem Cluster \mathcal{C}_i gehört. In diesem Fall gilt, dass der Abstand des Auftrags ha_j zu z_i kleiner ist als zu allen anderen Clusterzentren z_l mit $i \neq l$:

$$\text{dist}(z_i, ha_j) \leq \text{dist}(z_l, ha_j) \quad (l = 1, \dots, k; i \neq l)$$

¹⁰Die Aufträge eines Clusters sollten möglichst homogen, d. h. hinsichtlich ihrer Eigenschaftsausprägungen möglichst ähnlich, sein.

¹¹Zwischen den Clustern sollte eine möglichst große Heterogenität bestehen, d. h., die Aufträge aus zwei unterschiedlichen Clustern sollten hinsichtlich ihrer Eigenschaftsausprägungen möglichst unähnlich sein.

¹²Vgl. [JD88], S. 8 ff., 24; [AF07], S. 5.

¹³Vgl. [JD88], S. 11 f.; [AF07], S. 5.

Zur Bestimmung des Clusterzentrums z_i eines Clusters \mathcal{C}_i wird häufig der Mittelwert aus den Eigenschaften der enthaltenen Elemente (hier die Eigenschaftsvektoren der Aufträge) gebildet:¹⁴

$$e_{z_i,p} = \frac{\sum_{ha \in \mathcal{C}_i} e_{ha,p}}{|\mathcal{C}_i|} \quad (p = 1, \dots, d)$$

Aufgrund der verwendeten Eigenschaften (vgl. Abschnitt 5.2.1.1) ist die Bestimmung des Clusterzentrums eines Clusters durch Mittelwertbildung der enthaltenen Aufträge jedoch nur eingeschränkt möglich. Die Interpretierbarkeit reeller Zahlen, wie dem Mittelwert, für nominale oder binäre Eigenschaften (vgl. Eigenschaft 1 und 2) ist nicht gegeben und auch für diskrete Wertebereiche, wie die ganzzahligen Eigenschaften (bspw. Eigenschaft 5 oder 9), ist sie nur begrenzt möglich.¹⁵ Zudem entspricht das so ermittelte Clusterzentrum nicht notwendigerweise einem tatsächlich durchgeführten Auftrag, sondern kann bedingt durch die Mittelwertbildung ein künstlich erzeugter Auftrag sein.¹⁶

Daher wird hier das von Kaufman & Rousseeuw¹⁷ beschriebene k-medoids-Verfahren zur Clusteranalyse verwendet. Bei diesem wird als Clusterzentrum ein repräsentativer Auftrag $z_i \in \mathcal{C}_i$ aus den Aufträgen eines Clusters \mathcal{C}_i gewählt, der als Clusterzentrum (auch Medoid genannt) verwendet werden soll.¹⁸ Als repräsentativ wird der Auftrag bezeichnet, dessen durchschnittliche Distanz zu allen anderen Aufträgen desselben Clusters minimal ist. Somit gilt für die Menge der Clusterzentren: $\mathcal{Z} \subset \mathcal{A}^H$.

Ziel des k-medoids-Verfahrens ist es, die Menge \mathcal{Z} mit $|\mathcal{Z}| = k$ zu bestimmen,¹⁹ die die historischen Aufträge \mathcal{A}^H am besten repräsentiert.²⁰ Hierzu sind zwei Entscheidungen zu treffen. Die erste bestimmt, welche k Aufträge als repräsentative Aufträge die Clusterzentren bilden (vgl. Entscheidungsvariable y_z). Die zweite Entscheidung legt fest, welche Aufträge $ha \in \mathcal{A}^H \setminus \mathcal{Z}$ welchem Clusterzentrum $z \in \mathcal{Z}$ zugeordnet werden (vgl. Entscheidungsvariable $x_{ha,z}$). Das entsprechende Optimierungsmodell (5.6) lässt sich folgendermaßen formalisieren:²¹

Mengen:

\mathcal{A}^H Menge historischer Aufträge

Parameter:

k Die Anzahl der zu bildenden Cluster

¹⁴So beispielsweise beim k-means-, aber auch einigen hierarchischen Clusterverfahren (vgl. [BPW10], S. 345).

¹⁵Vgl. ebd.

¹⁶Vgl. [TK09], S. 745 f.

¹⁷Siehe [KR87].

¹⁸Vgl. [TK09], S. 745; [BPW10], S. 347.

¹⁹Der Parameter k ist ein zu definierender Parameter, der projektspezifisch zu bestimmen ist. Siehe Abschnitt 6.1.2.2 für die Beschreibung eines Parametrisierungsexperiments, bei der die Anzahl zu verwendender Cluster k bestimmt wird.

²⁰Vgl. [TK09], S. 745.

²¹Vgl. [KR87], S. 406 f.

Entscheidungsvariablen:

- $x_{ha,z}$ Binär: 1, wenn Auftrag $ha \in \mathcal{A}^H$ dem Cluster, dessen Clusterzentrum z mit $z \in \mathcal{A}^H$ ist, zugeordnet wird; 0 in allen anderen Fällen
- y_z Binär: 1, wenn Auftrag $z \in \mathcal{A}^H$ als repräsentativer Auftrag für ein Clusterzentrum ausgewählt wird; 0 in allen anderen Fällen

Zielfunktion:

$$\min c^{\text{medoid}} = \sum_{ha \in \mathcal{A}^H} \sum_{z \in \mathcal{A}^H} x_{ha,z} \text{dist}(ha, z) \quad (5.6a)$$

u. B. d. N.:

$$\sum_{z \in \mathcal{A}^H} x_{ha,z} = 1 \quad , \forall ha \in \mathcal{A}^H \quad (5.6b)$$

$$\sum_{z \in \mathcal{A}^H} y_z = k \quad (5.6c)$$

$$x_{ha,z} \leq y_z \quad , \forall ha \in \mathcal{A}^H; \forall z \in \mathcal{A}^H \quad (5.6d)$$

$$x_{ha,z} \in \{0, 1\} \quad , \forall ha \in \mathcal{A}^H; \forall z \in \mathcal{A}^H \quad (5.6e)$$

$$y_z \in \{0, 1\} \quad , \forall z \in \mathcal{A}^H \quad (5.6f)$$

Die Zielfunktion in Gleichung (5.6a) minimiert die Distanz, gemessen über die Distanzfunktion (5.5) aller Aufträge zu ihren Clusterzentren. Die Nebenbedingung (5.6b) stellt sicher, dass jeder Auftrag genau einem Clusterzentrum und somit nur einem Cluster zugeordnet wird. Die Aufträge können dabei nur Clusterzentren und nicht anderen Aufträgen, die keine Zentren sind, zugeordnet werden (Nebenbedingung (5.6d)). Die Gleichung (5.6c) stellt dar, dass aus den vorhandenen Aufträgen genau k Aufträge als Clusterzentren zu bestimmen sind. Schließlich stellen die Nebenbedingungen (5.6e) und (5.6f) die binäre Eigenschaft der Entscheidungsvariablen sicher.

Aufgrund der Komplexität des Modells (5.6) wird das heuristische Verfahren PAM²² verwendet.²³ Sei hierfür die Kostenfunktion $\text{cost}(\mathcal{Z})$ gegeben, mit der die Qualität einer gefundenen Lösung \mathcal{Z} bewertet werden kann:²⁴

$$\text{cost}(\mathcal{Z}) = \sum_{ha \in \mathcal{A}^H \setminus \mathcal{Z}} \sum_{z \in \mathcal{Z}} u_{ha,z} \text{dist}(ha, z)$$

und

$$u_{ha,z} = \begin{cases} 1, & \text{wenn } \text{dist}(ha, z) = \min_{z' \in \mathcal{Z}} \text{dist}(ha, z') \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

²²Vgl. [KR87], S. 410 ff.

²³Als Alternative für die effiziente Berechnung bei großen Datenmengen wird in der Implementierung das Verfahren von Park & Jun [PJ09] verwendet.

²⁴Vgl. [TK09], S. 745.

Das Verfahren zur Clusteranalyse von Aufträgen, welches in Algorithmus 5.1 aufgeführt ist, setzt die folgenden fünf Schritte der Heuristik²⁵ um:

1. Wahl einer der folgenden Strategien zur Auswahl von k Aufträgen $z \in \mathcal{A}^H$, die die initiale Menge der Clusterzentren \mathcal{Z} bilden (Zeilen 2–4):
 - a) Random: Zufällige Auswahl der k Aufträge
 - b) Most-Middle: Auswahl der k zentralsten Aufträge²⁶
 - c) Farthest-First: Iterative Auswahl (k -mal) eines Clusterzentrums, das am weitesten von den bisherigen Clusterzentren entfernt ist²⁷
2. Für jedes Clusterzentrum $z \in \mathcal{Z}$ und jeden Auftrag $ha \in \mathcal{A}^H \setminus \mathcal{Z}$, der kein Clusterzentrum ist, vertausche die Rollen. Es ergibt sich die Menge $\mathcal{Z}_{z,ha} = \mathcal{Z} \setminus \{z\} \cup \{ha\}$. Für jede durchgeführte Vertauschung berechne die hieraus resultierende Kostendifferenz $\Delta c_{z,ha}^{\text{medoid}} = \text{cost}(\mathcal{Z}_{z,ha}) - \text{cost}(\mathcal{Z})$ (Zeilen 8–16).
3. Wähle die Menge $\mathcal{Z}_{z',ha'}$ mit $z' \in \mathcal{Z}$ und $ha' \in \mathcal{A}^H \setminus \mathcal{Z}$, für die die Kosten minimal sind: $\Delta c_{z',ha'}^{\text{medoid}} = \min_{z \in \mathcal{Z}, ha \in \mathcal{A}^H \setminus \mathcal{Z}} \Delta c_{z,ha}^{\text{medoid}}$. Wenn $\Delta c_{z',ha'}^{\text{medoid}} < 0$ gilt, ersetze \mathcal{Z} mit $\mathcal{Z}_{z',ha'}$ (Zeilen 17–19).
4. Wiederholen der Schritte 2 und 3, solange $\Delta c_{z',ha'}^{\text{medoid}} < 0$ gilt, d. h. solange durch eine Vertauschung die Kosten gesenkt werden können (Zeilen 5–19).
5. Zuordnung aller Aufträge $ha \in \mathcal{A}^H \setminus \mathcal{Z}$ zu dem gemessen auf Basis der Distanzfunktion (5.5) nächsten Clusterzentrum $\arg \min_{z \in \mathcal{Z}} \text{dist}(ha, z)$ und damit zu dem Cluster, dessen Clusterzentrum z ist (bezeichnet mit \mathcal{C}_z) (Zeilen 20–29).

5.2.1.4 Ermittlung von Intervallgrenzen der Bearbeitungszeitabweichungen

Mit Hilfe des im vorherigen Abschnitt beschriebenen Verfahrens zur Clusteranalyse wurden Aufträge, die sich hinsichtlich ihrer Eigenschaftsausprägungen ähneln, zu Clustern zusammengefasst. Für jedes Cluster ist im Folgenden ein Intervall der aufgetretenen Bearbeitungszeitabweichungen der Aufträge zu bestimmen.

Für die historischen Aufträge $ha \in \mathcal{C}$ eines Clusters $\mathcal{C} \in \mathcal{C}$ sind sowohl die vor der Durchführung des Auftrags durch z. B. Gleichung (2.1) berechnete Bearbeitungszeit $\hat{\tau}_{ha}$ als auch die realisierte Bearbeitungszeit τ_{ha} gegeben (vgl. Abschnitt 5.2.1). Aus diesen beiden Werten lässt sich die Qualität der Schätzung der Bearbeitungszeit durch die Berechnung der Bearbeitungszeitabweichung $\Delta\tau_{ha}$ ermitteln:

$$\Delta\tau_{ha} = \frac{\tau_{ha} - \hat{\tau}_{ha}}{\hat{\tau}_{ha}}$$

²⁵Vgl. [TK09], S. 746 ff.; [KR87], S. 411.

²⁶Siehe [PJ09], S. 3337.

²⁷Vgl. [HS85]; [DL02]; [DL05].

Algorithmus 5.1: Clusteranalyse von Aufträgen

Input : \mathcal{A}^H – Die Menge der historischen Aufträge
 DM – Die Distanzmatrix
 k – Anzahl zu erzeugender Cluster

Output : \mathcal{C} – Menge der ermittelten Cluster

```

1  $\mathcal{Z} \leftarrow \text{null}$ ;
  /* Schritt 1 des Algorithmus */
2 for  $i \leftarrow 1$  to  $k$  do
3    $z \leftarrow \text{strat}_s(\mathcal{A}^H \setminus \mathcal{Z})$ ; // Auswahl eines Auftrags nach Strategie  $s$ 
4    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{z\}$ ;
  /* Schritt 4 des Algorithmus */
5 changed  $\leftarrow \text{true}$ ;
6 while changed do
7   changed  $\leftarrow \text{false}$ ;
  /* Schritte 2 und 3 des Algorithmus */
8    $\mathcal{Z}_{z',ha'} \leftarrow \text{null}$ ;
9    $\Delta_{z',ha'}^{\text{medoid}} \leftarrow \infty$ ;
10  foreach  $z \in \mathcal{Z}$  do
11    foreach  $ha \in \mathcal{A}^H \setminus \mathcal{Z}$  do
12       $\mathcal{Z}_{z,ha} \leftarrow \mathcal{Z} \setminus \{z\} \cup \{ha\}$ ;
13       $\Delta_{z,ha}^{\text{medoid}} \leftarrow \text{cost}(\mathcal{Z}_{z,ha}) - \text{cost}(\mathcal{Z})$ ;
14      if  $\Delta_{z,ha}^{\text{medoid}} < \Delta_{z',ha'}^{\text{medoid}}$  then
15         $\mathcal{Z}_{z',ha'} \leftarrow \mathcal{Z}_{z,ha}$ ;
16         $\Delta_{z',ha'}^{\text{medoid}} \leftarrow \Delta_{z,ha}^{\text{medoid}}$ ;
17  if  $\Delta_{z',ha'}^{\text{medoid}} < 0$  then
18     $\mathcal{Z} \leftarrow \mathcal{Z}_{z',ha'}$ ;
19    changed  $\leftarrow \text{true}$ ;
  /* Schritt 5 des Algorithmus */
20 foreach  $ha \in \mathcal{A}^H \setminus \mathcal{Z}$  do
21    $z^{\min} \leftarrow \text{null}$ ;
22   distancemin  $\leftarrow \infty$ ;
23   foreach  $z \in \mathcal{Z}$  do
24     distance  $\leftarrow 0$ ;
25     if  $dm_{ha,z} = \text{null}$  then distance  $\leftarrow dm_{z,ha}$  else distance  $\leftarrow dm_{ha,z}$ ;
26     if distance  $<$  distancemin then
27        $z^{\min} \leftarrow z$ ;
28       distancemin  $\leftarrow \text{distance}$ ;
29    $\mathcal{C}_{z^{\min}} \leftarrow \mathcal{C}_{z^{\min}} \cup \{ha\}$ ;
30 return  $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ ;

```

Es bezeichne zudem $\Delta\tau_{\mathcal{C}}^{\min}$ den minimalen und $\Delta\tau_{\mathcal{C}}^{\max}$ den maximalen Wert der Bearbeitungszeitabweichungen aller in einem Cluster \mathcal{C} enthaltenden Aufträge:

$$\Delta\tau_{\mathcal{C}}^{\min} = \min_{ha \in \mathcal{C}} \Delta\tau_{ha} \quad (5.7a)$$

und

$$\Delta\tau_{\mathcal{C}}^{\max} = \max_{ha \in \mathcal{C}} \Delta\tau_{ha} \quad (5.7b)$$

Dann bezeichne $I_{\mathcal{C}} = [\Delta\tau_{\mathcal{C}}^{\min}, \Delta\tau_{\mathcal{C}}^{\max}]$ das Intervall der Bearbeitungszeitabweichungen für Cluster \mathcal{C} .

Das Intervall $I_{\mathcal{C}}$ enthält alle aufgetretenen Bearbeitungszeitabweichungen, auch solche, die beispielsweise sehr selten vorgekommen und daher als Ausreißer zu betrachten sind. Aus diesem Grund soll die Möglichkeit bestehen, die Ausdehnung des Intervalls zu begrenzen und damit Aufträge, deren Bearbeitungszeitabweichung außerhalb des Intervalls liegt, nicht zu berücksichtigen. Hierfür wird die empirische kumulative Verteilungsfunktion²⁸

$$\widehat{F}_{\mathcal{C}}(x) = \frac{1}{|\mathcal{C}|} \sum_{ha' \in \mathcal{C}} \delta(\Delta\tau_{ha'} \leq x)$$

mit der Indikatorfunktion

$$\delta(\Delta\tau_{ha'} \leq x) = \begin{cases} 1, & \text{wenn } \Delta\tau_{ha'} \leq x \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

gebildet. Bezeichne $p_{\mathcal{C}} \in (0, 1)$ ²⁹ den gewünschten Prozentsatz des Quantils, das für das Cluster \mathcal{C} ermittelt werden soll. Des Weiteren sei folgende Quantilfunktion zur Berechnung des kleinstmöglichen $p_{\mathcal{C}}$ -Quantils gegeben:

$$Q(p) = \inf \left\{ \Delta\tau_{ha} \mid ha \in \mathcal{C} \wedge \widehat{F}_{\mathcal{C}}(\Delta\tau_{ha}) \geq p \right\}$$

Das $p_{\mathcal{C}}$ -Quantil $\Delta\tau_{\mathcal{C}}^{p_{\mathcal{C}}} = Q(p_{\mathcal{C}})$ beschreibt dann die Bearbeitungszeitabweichung, für die gilt, dass $p_{\mathcal{C}} \cdot 100\%$ aller Bearbeitungszeitabweichungen der in dem Cluster enthaltenen Aufträge kleiner sind.

Schließlich sind aus Cluster \mathcal{C} alle Aufträge zu entfernen, deren Bearbeitungszeitabweichung größer als das ermittelte Quantil ist. Es ergibt sich das verkleinerte Cluster \mathcal{C}^* :

$$\mathcal{C}^* = \{ha \mid ha \in \mathcal{C} \wedge \Delta\tau_{ha} \leq \Delta\tau_{\mathcal{C}}^{p_{\mathcal{C}}}\}$$

²⁸Vgl. [ZM14], S. 33 f.

²⁹Der Prozentsatz wird für jedes Cluster $\mathcal{C} \in \mathcal{C}$ gesondert definiert. Hierdurch kann gesteuert werden, welche Intervalllänge pro Cluster verwendet wird. Dies hat Einfluss auf die spätere Ablaufplanung, da für neu einzuplanende Aufträge unter anderem die hier ermittelten Intervalle Verwendung finden (vgl. Abschnitt 5.3).

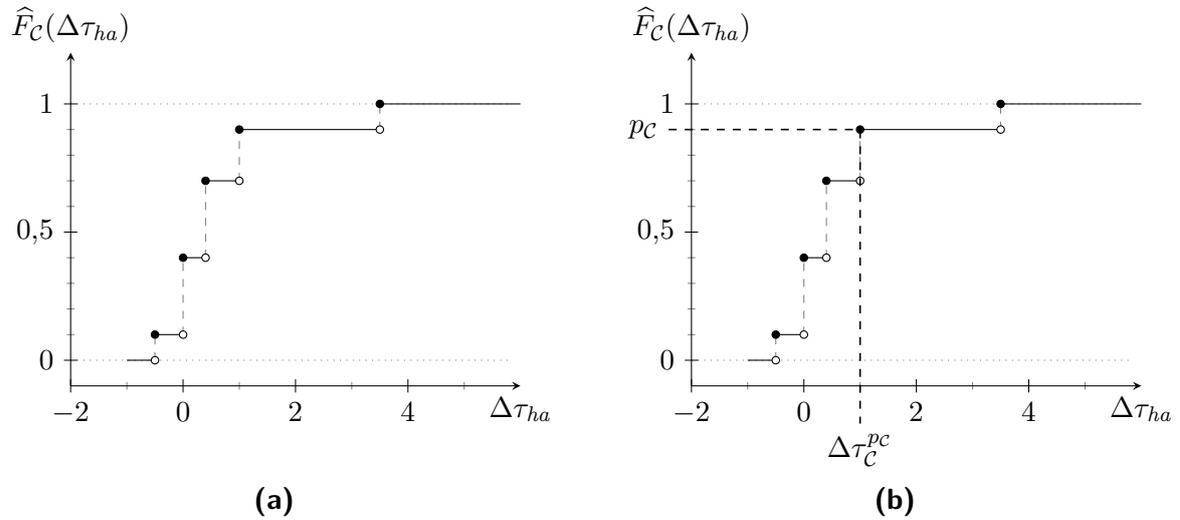


Abbildung 5.2: Kumulative Verteilungs- und Quantilfunktion der Bearbeitungszeitabweichungen im Intervall I_C : (a) Verteilungsfunktion für Cluster \mathcal{C} (b) Mit der Quantilfunktion $Q(p_C)$ ermitteltes p_C -Quantil für Cluster \mathcal{C}

Für dieses Cluster wird das verkleinerte Intervall $I_{C^*} = [\Delta\tau_C^{\min}, \Delta\tau_C^{p_C}]$ festgelegt bzw. durch Gleichung (5.7b) mit dem neuen Cluster \mathcal{C}^* ermittelt. Die Menge der neuen Cluster \mathcal{C}^* wird mit \mathcal{C}^* bezeichnet.

Ein Beispiel für einen möglichen Verlauf der Verteilungsfunktion für ein Cluster \mathcal{C} ist in Abbildung 5.2(a) gegeben. Hier sind 90% der Bearbeitungszeitabweichungen kleiner als $\Delta\tau_{ha} = 1$. In Abbildung 5.2(b) wird mit Hilfe der Quantilfunktion und dem gewünschten Prozentsatz $p_C = 0,9$ das p_C -Quantil $\Delta\tau_C^{p_C}$ ermittelt, für das 90% der Bearbeitungszeitabweichungen der Aufträge des Clusters kleiner sind. Die 10% der Werte mit einer Bearbeitungszeitabweichung von $\Delta\tau_{ha} = 3,5$ sind somit von der weiteren Betrachtung ausgenommen.

In Algorithmus 5.2 ist das zuvor beschriebene Vorgehen als Pseudocode aufgeführt. Zunächst wird eine aufsteigend sortierte Liste aller in einem Cluster \mathcal{C} vorkommenden Bearbeitungszeitabweichungen erstellt (Zeilen 4–7). Aus dieser wird anschließend das p_C -Quantil $\Delta\tau_C^{p_C}$ bestimmt (Zeilen 8–9). Mit dem ermittelten Quantil wird die verkleinerte Menge \mathcal{C}^* gebildet (Zeilen 10–13) und schließlich der Menge \mathcal{C}^* hinzugefügt (Zeile 14).

5.2.1.5 Ermittlung von Häufigkeitsverteilungen der Bearbeitungszeitabweichungen

Für die ermittelten Cluster $\mathcal{C}^* \in \mathcal{C}^*$ ähnlicher historischer Aufträge und das für jedes Cluster bestimmte Intervall I_{C^*} sind im Folgenden aggregierte probabilistische Informationen über die Bearbeitungszeitabweichungen der in einem Cluster enthaltenen Aufträge zu generieren. Die Aggregation wird durch die Erstellung einer Häufigkeitsverteilung über

Algorithmus 5.2: Ermittlung der verkleinerten Clustermenge \mathcal{C}^*

Input : \mathcal{C} – Menge der ermittelten Cluster
 p_C – Prozentsatz des Quantils für Cluster \mathcal{C}
Output : \mathcal{C}^* – Menge der neuen Cluster. Diese enthalten nur Aufträge, deren Bearbeitungszeitabweichungen kleiner als das ermittelte Quantil sind.

```

1  $\mathcal{C}^* \leftarrow \text{null};$ 
2 foreach  $\mathcal{C} \in \mathcal{C}$  do
3    $\Delta\tau_{\mathcal{C}}^{p_C} \leftarrow 0;$ 
4 foreach  $\mathcal{C} \in \mathcal{C}$  do
5   foreach  $ha \in \mathcal{C}$  do
6      $\text{list} \leftarrow \text{list.Add}(\Delta\tau_{ha});$            // Hinzufügen von  $\Delta\tau_{ha}$  zu einer Liste
7      $\text{list} \leftarrow \text{sort}(\text{list});$            // Aufsteigende Sortierung der Liste
8      $\text{element} \leftarrow \lceil |\mathcal{C}| \cdot p_C \rceil;$ 
9      $\Delta\tau_{\mathcal{C}}^{p_C} \leftarrow \text{list.get}(\text{element});$ 
10     $\mathcal{C}^* \leftarrow \text{null};$ 
11    foreach  $ha \in \mathcal{C}$  do
12      if  $\Delta\tau_{ha} \leq \Delta\tau_{\mathcal{C}}^{p_C}$  then
13         $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{ha\};$ 
14     $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{\mathcal{C}^*\};$ 
15 return  $\mathcal{C}^*;$ 

```

die in einem Cluster enthaltenden Aufträge und deren Bearbeitungszeitabweichungen erreicht.

Für eine genauere Beschreibung der innerhalb des Intervalls $I_{\mathcal{C}^*}$ auftretenden Bearbeitungszeitabweichungen sollen diese mittels einer Häufigkeitsverteilung dargestellt werden. Bei der Erstellung dieser Verteilung kann es im Extremfall vorkommen, dass jede Bearbeitungszeitabweichung $\Delta\tau_{ha}$ der Aufträge $ha \in \mathcal{C}^*$ eines Clusters unterschiedlich ist und somit eine absolute Häufigkeit

$$h_{\Delta\tau_{ha}} = \sum_{ha' \in \mathcal{C}^* | \Delta\tau_{ha'} = \Delta\tau_{ha}} 1$$

von eins vorliegt bzw. sich nur sehr wenige Werte exakt gleichen.

Ein Beispiel ist in Abbildung 5.3(a) aufgeführt. In diesem sind die Bearbeitungszeitabweichungen nur für zwei Aufträge identisch und die absolute Häufigkeit beträgt somit zwei. Alle anderen Aufträge unterschieden sich in der Ausprägung ihrer Bearbeitungszeitabweichungen teilweise nur um wenige Prozent. Dies führt jedoch dazu, dass alle anderen absoluten Häufigkeiten eins betragen.

Für die später durchzuführende Ablaufplanung sind jedoch aussagekräftige Informationen über die aufgetretenen Häufigkeiten notwendig. Aus diesem Grund wird das

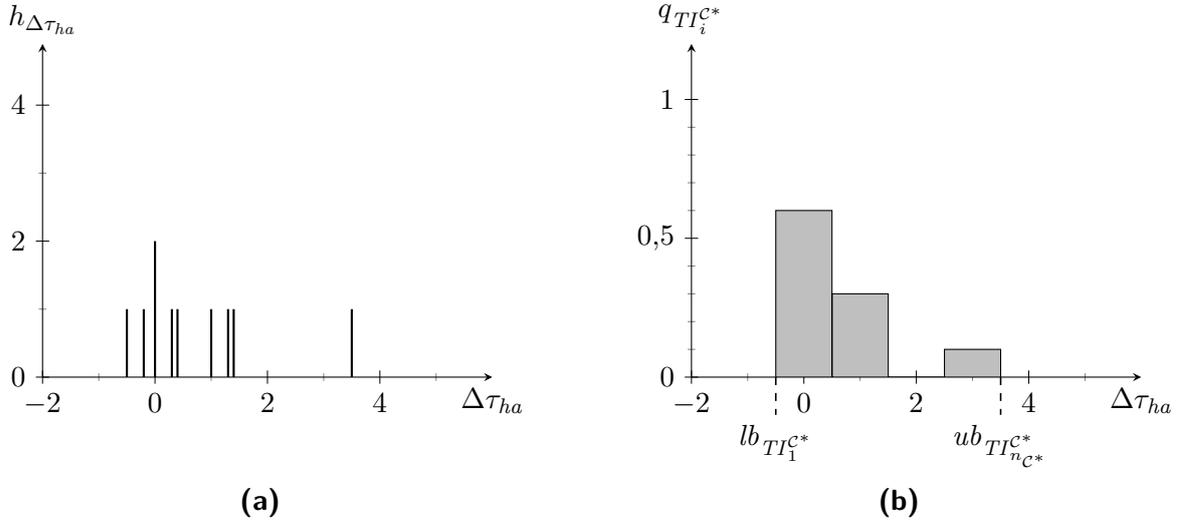


Abbildung 5.3: Aufbau einer Häufigkeitsverteilung der Bearbeitungszeitabweichungen $\Delta\tau_{ha}$ für die Aufträge eines Clusters $ha \in C^*$, dargestellt als Histogramm: (a) Absolute Häufigkeitsverteilung ohne die Einteilung in Klassen (b) Relative Häufigkeitsverteilung mit Einteilung in Klassen

Intervall I_{C^*} in n_{C^*} ³⁰ gleich große Teilintervalle bzw. Klassen $TI_i^{C^*}$ mit $1 \leq i \leq n_{C^*}$ unterteilt. Jede Klasse hat somit die Breite³¹ $kbr_{C^*} = (\Delta\tau_{C^*}^{\max} + \epsilon) - \Delta\tau_{C^*}^{\min} / n_{C^*}$. Die i -te Klasse wird durch die untere Grenze

$$lb_{TI_i^{C^*}} = \Delta\tau_{C^*}^{\min} + ((i - 1) \cdot kbr_{C^*})$$

und durch die obere Grenze

$$ub_{TI_i^{C^*}} = \Delta\tau_{C^*}^{\min} + i \cdot kbr_{C^*}$$

beschrieben und ergibt somit das rechtsoffene Intervall $TI_i^{C^*} = [lb_{TI_i^{C^*}}, ub_{TI_i^{C^*}})$. Jeder Auftrag $ha \in C^*$ gehört somit genau zu der Klasse $TI_i^{C^*}$, in dessen Intervall die Bearbeitungszeitabweichung $\Delta\tau_{ha} \in TI_i^{C^*}$ liegt.

Für die Erstellung der relativen Häufigkeiten $q_{TI_i^{C^*}}$ einer Klasse $TI_i^{C^*}$ wird schließlich die Anzahl der Aufträge $ha \in C^*$, die aufgrund ihrer Bearbeitungszeitabweichungen $\Delta\tau_{ha}$ in dieser Klasse liegen, ermittelt und ihre Anzahl durch die Gesamtanzahl an Werten des Clusters C^* dividiert:

$$q_{TI_i^{C^*}} = \frac{1}{|C^*|} \sum_{ha \in C^* | lb_{TI_i^{C^*}} \leq \Delta\tau_{ha} < ub_{TI_i^{C^*}}} 1$$

³⁰Die Anzahl der Klassen kann somit für jedes Cluster separat gesteuert werden. Dies ist bspw. dann sinnvoll, wenn sich die Anzahl der in einem Cluster enthaltenen Elemente stark von denen anderer Cluster unterscheidet.

³¹Zu der maximalen Bearbeitungszeitabweichung $\Delta\tau_{C^*}^{\max}$ wird der infinitesimale Wert ϵ addiert, um bei der Erstellung nicht überlappender Klassen den Wert $\Delta\tau_{C^*}^{\max}$ in der Klasse $TI_{n_{C^*}}^{C^*}$ zu berücksichtigen.

Die relativen Häufigkeiten werden schließlich dazu verwendet, die empirische Wahrscheinlichkeitsfunktion $\hat{p}_{\mathcal{C}^*}(ha)$ für Cluster \mathcal{C}^* zu definieren:

$$\hat{p}_{\mathcal{C}^*}(ha) = \begin{cases} q_{TI_i^{\mathcal{C}^*}}, & \text{wenn } lb_{TI_i^{\mathcal{C}^*}} \leq \Delta\tau_{ha} < ub_{TI_i^{\mathcal{C}^*}} \\ 0 & \text{in allen anderen Fällen} \end{cases} \quad (5.8)$$

Des Weiteren wird mit $\mathcal{H}_{\mathcal{C}^*}$ die relative Häufigkeitsverteilung bezeichnet.

In Abbildung 5.3(b) ist ein Beispiel für eine relative Häufigkeitsverteilung aufgeführt. Hier wurden die in Abbildung 5.3(a) dargestellten absoluten Häufigkeiten zu vier Klassen zusammengefasst (wobei die dritte Klasse keine Werte enthält) und es wurden die relativen Häufigkeiten gebildet.

Die Erstellung der Häufigkeitsverteilung für jedes Cluster $\mathcal{C}^* \in \mathcal{C}^*$ und deren Ausgabe in Form der empirischen Wahrscheinlichkeitsfunktion aus Gleichung (5.8) ist in Algorithmus 5.3 aufgeführt. Zunächst werden die minimalen ($\Delta\tau_{\mathcal{C}^*}^{\min}$) und maximalen ($\Delta\tau_{\mathcal{C}^*}^{\max}$) Bearbeitungszeitabweichungen für jedes Cluster ermittelt sowie die Klassenbreite pro Cluster festgelegt (Zeilen 7–14). Anschließend erfolgt die Ermittlung der relativen Häufigkeiten $q_{TI_i^{\mathcal{C}^*}}$ für jede Klasse $TI_i^{\mathcal{C}^*}$ (Zeilen 15–20).

5.2.2 Klassifikatorentwicklung für unbekannte Aufträge

Die in den vorherigen Abschnitten beschriebene Aufbereitung der Informationen über die Bearbeitungszeiten der historischen Abfertigungsaufträge kann je nach Umfang der verwendeten Menge \mathcal{A}^H einige Zeit in Anspruch nehmen. Diese Zeit steht jedoch im Rahmen der durchzuführenden Ablaufplanung nicht zur Verfügung, da es das Ziel ist, den Ablaufplan möglichst schnell zu erstellen. Eine vollständige Clusteranalyse für jeden Ablaufplanungsschritt ist daher nicht zielführend und wird im Folgenden durch einen auf Basis der ermittelten Clustermenge \mathcal{C}^* trainierten Klassifikator ersetzt. Eine Aktualisierung der Clustermenge \mathcal{C}^* findet dann in längeren Zeitabschnitten durch die erneute Ausführung einer Clusteranalyse statt.³² Der trainierte Klassifikator ermöglicht eine schnelle Zuordnung eines einzuplanenden Auftrags a mit $a \in \mathcal{A}$ zu einem der Cluster \mathcal{C}^* mit $\mathcal{C}^* \in \mathcal{C}^*$ und damit zu den Informationen über die Intervallgrenzen $I_{\mathcal{C}^*}$ und den Häufigkeitsverteilungen $\mathcal{H}_{\mathcal{C}^*}$ bzw. den ermittelten empirischen Wahrscheinlichkeitsfunktionen $\hat{p}_{\mathcal{C}^*}(ha)$ der Bearbeitungszeitabweichungen der in dem Cluster enthaltenen historischen Aufträge. Im Folgenden wird daher zunächst in Abschnitt 5.2.2.1 erläutert, wie die für den Klassifikator notwendigen Trainingsdaten erstellt werden, mit denen anschließend in Abschnitt 5.2.2.2 der Klassifikator trainiert werden kann.

5.2.2.1 Erstellung von Trainingsdaten

Grundlage für die Erstellung der Trainingsdaten sind die in den vorherigen Abschnitten ermittelten Cluster \mathcal{C}^* und die einem Cluster zugewiesenen Aufträge $ha \in \mathcal{C}^*$. Für die

³²Dies kann bspw. durch eine parallele Rechnerinstanz, die von der eigentlichen Ablaufplanung unabhängig ist, durchgeführt werden.

Algorithmus 5.3: Ermittlung der Häufigkeitsverteilungen der Bearbeitungszeitabweichungen

Input : \mathcal{C}^* – Menge der ermittelten Cluster
 $n_{\mathcal{C}^*}$ – Die Anzahl der zu erzeugenden Klassen für jedes Cluster \mathcal{C}^*

Output : $\hat{p}_{\mathcal{C}^*}(ha)$ – Ausgabe der empirischen Wahrscheinlichkeitsfunktionen

```

1 foreach  $\mathcal{C}^* \in \mathcal{C}^*$  do
2    $kbr_{\mathcal{C}^*} \leftarrow 0$ ;
3    $\Delta\tau_{\mathcal{C}^*}^{\min} \leftarrow +\infty$ ;
4    $\Delta\tau_{\mathcal{C}^*}^{\max} \leftarrow -\infty$ ;
5   for  $i \leftarrow 1$  to  $n_{\mathcal{C}^*}$  do
6      $q_{TI_i^{\mathcal{C}^*}} \leftarrow 0$ ;
   /* Festlegen der Klassenbreiten für jedes Cluster */
7 foreach  $\mathcal{C}^* \in \mathcal{C}^*$  do
8   foreach  $ha \in \mathcal{C}^*$  do
9      $\Delta\tau_{ha} \leftarrow |\tau_{ha} - \hat{\tau}_{ha}| / \hat{\tau}_{ha}$ ;
10    if  $\Delta\tau_{ha} < \Delta\tau_{\mathcal{C}^*}^{\min}$  then
11       $\Delta\tau_{\mathcal{C}^*}^{\min} \leftarrow \Delta\tau_{ha}$ ;
12    if  $\Delta\tau_{ha} > \Delta\tau_{\mathcal{C}^*}^{\max}$  then
13       $\Delta\tau_{\mathcal{C}^*}^{\max} \leftarrow \Delta\tau_{ha}$ ;
14     $kbr_{\mathcal{C}^*} \leftarrow (\Delta\tau_{\mathcal{C}^*}^{\max} + \epsilon) - \Delta\tau_{\mathcal{C}^*}^{\min} / n_{\mathcal{C}^*}$ ;
   /* Ermitteln der relativen Häufigkeiten jeder Klasse */
15 foreach  $\mathcal{C}^* \in \mathcal{C}^*$  do
16   foreach  $ha \in \mathcal{C}^*$  do
17      $\Delta\tau_{ha} \leftarrow |\tau_{ha} - \hat{\tau}_{ha}| / \hat{\tau}_{ha}$ ;
18     for  $i \leftarrow 1$  to  $n_{\mathcal{C}^*}$  do
19       if  $lb_{TI_i^{\mathcal{C}^*}} \leq \Delta\tau_{ha} < ub_{TI_i^{\mathcal{C}^*}}$  then  $q_{TI_i^{\mathcal{C}^*}} \leftarrow q_{TI_i^{\mathcal{C}^*}} + 1$ ;
20      $q_{TI_i^{\mathcal{C}^*}} \leftarrow \frac{1}{|\mathcal{C}^*|} q_{TI_i^{\mathcal{C}^*}}$ ;
21 return  $\hat{p}_{\mathcal{C}^*}(ha)$ ;

```

Klassifizierung wird eine Menge \mathcal{T} von Trainingsdaten benötigt, wobei jedes Trainingsdatum durch ein Eingabe-Ausgabe-Paar definiert ist, bei dem die Eingabe durch ein Klassenlabel (die Ausgabe) beschriftet ist. In dem vorliegenden Fall entspricht eine Eingabe einem historischen Frachtbearbeitungsauftrag, der durch seinen Eigenschaftsvektor charakterisiert wird, und die Ausgabe einem der verkleinerten Cluster, die in Abschnitt 5.2.1.4 ermittelt wurden.

Ein Trainingsdatum kann somit als geordnetes Paar $td = (\vec{e}_{ha}, \mathcal{C}^*)$ mit $ha \in \mathcal{C}^*$ und $\mathcal{C}^* \in \mathcal{C}^*$ dargestellt werden. Für jeden Auftrag eines Clusters wird ein Trainingsdatum erstellt, welche dann zusammen die Trainingsmenge $\mathcal{T}_{\mathcal{C}^*}$ für dieses Cluster bilden. Dieses

Algorithmus 5.4: Erstellung von Trainingsdaten für den Klassifikator

```

Input   :  $\mathcal{C}^*$    – Menge der ermittelten Cluster
Output  :  $\mathcal{T}$       – Menge der erzeugten Trainingsdaten

1  $\mathcal{T} \leftarrow \text{null};$  // Erstellen einer leeren Trainingsmenge
2 foreach  $\mathcal{C}^* \in \mathcal{C}^*$  do
3    $\mathcal{T}_{\mathcal{C}^*} \leftarrow \text{null};$  // Erstellen einer leeren Trainingsmenge
   // für das Cluster
4   foreach  $ha \in \mathcal{C}^*$  do
5      $\mathcal{T}_{\mathcal{C}^*} \leftarrow \mathcal{T}_{\mathcal{C}^*} \cup \{(\vec{e}_{ha}, \mathcal{C}^*)\};$ 
6    $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_{\mathcal{C}^*};$ 
7 trainedClassifier = false;
8 return  $\mathcal{T};$ 

```

Vorgehen wird für jedes vorhandene Cluster wiederholt und die erzeugten Trainingsmengen werden schließlich zur Trainingsmenge \mathcal{T} für die Klassifizierung zusammengefasst:

$$\mathcal{T} = \bigcup_{\mathcal{C}^* \in \mathcal{C}^*} \mathcal{T}_{\mathcal{C}^*}$$

Das Vorgehen zur Erzeugung der Trainingsdaten ist in Algorithmus 5.4 dargestellt. Hier wird in Zeile 5 für jeden Auftrag eines Clusters ein Trainingsdatum erzeugt und zu der Trainingsmenge des Clusters hinzugefügt. Die Trainingsmengen für jedes Cluster werden schließlich in Zeile 6 zur Gesamttrainingsmenge \mathcal{T} zusammengefasst. Mit der Neuerstellung der Trainingsdaten muss ebenfalls der Klassifikator erneut trainiert werden. Dies geschieht über die Variable *trainedClassifier*, die in Zeile 7 auf *false* gesetzt wird, was dazu führt, dass in Algorithmus 5.5 (s. Seite 103) das Training des Klassifikators erneut ausgeführt wird. Nachfolgend wird zunächst die Herleitung dieses Algorithmus dargelegt.

5.2.2.2 Erstellung des Klassifikators

Auf Basis der im vorherigen Abschnitt erzeugten Trainingsdaten kann im Folgenden ein Klassifikator trainiert werden, der eine Klassifizierung ausführt und die Klassengrenzen zwischen den Clustern erlernt. Als Klassifikator kommt ein naiver Bayes-Klassifikator zum Einsatz, der als generatives Modell eine Wahrscheinlichkeitsverteilung über die Zugehörigkeit eines Auftrags zu allen Clustern berechnet und somit eine Bewertung dieser Zugehörigkeit in Form von Zugehörigkeitswahrscheinlichkeiten ermöglicht.

Sei X_p mit $1 \leq p \leq d$ ein Vektor von Zufallsvariablen, der die beobachteten Eigenschaftswerte angibt, und sei Y eine diskrete Zufallsvariable, die das Cluster bezeichnet. Der Klassifikator wird trainiert, indem zwei Mengen von Parametern aus den Trainingsdaten geschätzt werden. Die erste Parametermenge sind die bedingten Wahrscheinlichkeiten, die für jede Eigenschaft p und für jedes Klassenlabel \mathcal{C}_i^* ermittelt und

mit $P(X_p = e_p | Y = C_i^*)$ bezeichnet werden.³³ Die zweite Parametermenge sind die A-priori-Klassenwahrscheinlichkeiten $P(Y = C_i^*)$ für jedes Klassenlabel C_i^* . Die einzelnen Eigenschaften (außer Eigenschaft 1 und 2) sind aufgrund der in Abschnitt 5.2.1.2 durchgeführten Normalisierung stetig bzw. quasistetig,³⁴ mit einem Wertebereich von $[0, 1]$. Die Berechnung der ersten Parametermenge wird für die diskreten Eigenschaften anders durchgeführt als für die (quasi)stetigen Eigenschaften. Im Folgenden wird zunächst die erste Parametermenge für diskrete und anschließend für (quasi)stetige Eigenschaften berechnet.

Die Bestimmung der bedingten Wahrscheinlichkeiten für diskrete Eigenschaften (in diesem Fall gilt dies lediglich für die Eigenschaften 1 und 2) ist durch Abzählen möglich, da diese Eigenschaften nur eine endliche Anzahl J möglicher Ausprägungen annehmen können. Die Berechnung der relativen Auftretenshäufigkeit für eine bestimmte Ausprägung j einer diskreten Eigenschaft (bezeichnet als e_p^j) unter der Bedingung, dass das Cluster C_i^* gegeben ist, wird über Gleichung (5.9) ermittelt:

$$\widehat{P}(X_p = e_p^j | Y = C_i^*) = \frac{|\{td \mid td \in \mathcal{T} \wedge e_{td,p} = e_p^j \wedge C_{td}^* = C_i^*\}| + \alpha}{|\{td \mid td \in \mathcal{T} \wedge C_{td}^* = C_i^*\}| + \alpha J} \quad (5.9)$$

Hierbei bezeichne $e_{td,p}$ den Wert der ersten bzw. zweiten Eigenschaft des Eingabevektors, der als Eingabe für Trainingsdatum td dient, und C_{td}^* das zugeordnete Cluster als Ausgabe in diesem Trainingsdatum. Der Parameter $\alpha > 0$ dient als Glättungsparameter, um Nullwahrscheinlichkeiten zu verhindern, die auftreten würden, wenn eine bestimmte Eigenschaftsausprägung in den Trainingsdaten nicht vorhanden ist und der Zähler in Gleichung (5.9) somit null wäre.³⁵ Hierfür werden mit dem Parameter α gleichverteilt über alle möglichen Ausprägungen von X_p künstliche Beispiele erzeugt.³⁶

Für die (quasi)stetigen Eigenschaften kann keine Zählfunktion und damit nicht Gleichung (5.9) angewendet werden, da die möglichen Ausprägungen der Eigenschaften nicht abzählbar sind. Alternativ kann jedoch für jedes diskrete Klassenlabel $C^* \in \mathcal{C}^*$ als Verteilung für jede (quasi)stetige Eigenschaft p eine Normalverteilung³⁷ angenommen werden, für die der Erwartungswert und die Varianz aus den Trainingsdaten geschätzt werden:³⁸

$$\begin{aligned} \mu_{p,i} &= E[X_p | Y = C_i^*] \\ \sigma_{p,i}^2 &= E[(X_p - \mu_{p,i})^2 | Y = C_i^*] \end{aligned}$$

³³Präziser müsste $P(X_p = e_p | Y = \text{id}(C_i^*))$ mit $\text{id}: \mathcal{C}^* \rightarrow \mathcal{I}$ geschrieben werden, wobei \mathcal{I} die Indexmenge der Menge \mathcal{C}^* bezeichnet. Aus Gründen der Übersichtlichkeit wird im Folgenden jedoch die kürzere Schreibweise für alle (bedingten) Wahrscheinlichkeiten gewählt.

³⁴Aufgrund der hohen Anzahl möglicher Ausprägungen werden die diskreten Eigenschaften 5 und 9 im Folgenden so behandelt, als seien ihre Wertebereiche stetig.

³⁵Vgl. [Mit05], S. 5.

³⁶Dieses Konzept ist auch unter dem Namen *Additive Smoothing* bekannt.

³⁷Eine Alternative für die Schätzung der (quasi)stetigen Eigenschaften bietet die Verwendung von nicht-parametrischen Dichteschätzern anstatt einer einzigen Normalverteilung. John & Langley (vgl. [JL95]) beschreiben z. B. die Anwendung eines Kerndichteschätzers.

³⁸Vgl. [Mit05], S. 6.

Der Erwartungswert und die Varianz können aus den Trainingsdaten z. B. über einen Maximum-Likelihood-Schätzer ermittelt werden. Für den Erwartungswert einer Eigenschaft p und eines Klassenlabels \mathcal{C}_i^* ergibt sich Gleichung (5.10) und für die korrigierte Stichprobenvarianz die Gleichung (5.11):³⁹

$$\hat{\mu}_{p,i} = \frac{1}{\sum_{td \in \mathcal{T}} \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*)} \sum_{td \in \mathcal{T}} e_{td,p} \cdot \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*) \quad (5.10)$$

$$\hat{\sigma}_{p,i}^2 = \frac{1}{\left(\sum_{td \in \mathcal{T}} \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*)\right) - 1} \sum_{td \in \mathcal{T}} (e_{td,p} - \hat{\mu}_{p,i}) \cdot \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*) \quad (5.11)$$

mit der Indikatorfunktion

$$\delta(x) = \begin{cases} 1, & \text{wenn } x \text{ wahr ist} \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

Zur Berechnung der bedingten Wahrscheinlichkeiten der (quasi)stetigen Eigenschaften werden die geschätzten Parameter in die Dichtefunktion der Normalverteilung $\mathcal{N}(\hat{\mu}_{p,i}, \hat{\sigma}_{p,i}^2)$ eingesetzt.⁴⁰

$$\hat{P}(X_p = e_p | Y = \mathcal{C}_i^*) = f(e_p | \mathcal{C}_i^*) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{p,i}^2}} \exp\left\{-\frac{(e_p - \hat{\mu}_{p,i})^2}{2\hat{\sigma}_{p,i}^2}\right\} \quad (5.12)$$

Neben den bereits geschätzten bedingten Wahrscheinlichkeiten müssen die A-priori-Klassenwahrscheinlichkeiten $P(Y = \mathcal{C}_i^*)$ für jedes Klassenlabel \mathcal{C}_i^* aus den Trainingsdaten geschätzt werden:⁴¹

$$\hat{P}(Y = \mathcal{C}_i^*) = \frac{|\{td | td \in \mathcal{T} \wedge \mathcal{C}_{td}^* = \mathcal{C}_i^*\}| + \alpha}{|T| + \alpha|\mathcal{C}^*|}$$

Der Glättungsparameter α wird erneut verwendet, um Nullwahrscheinlichkeiten zu verhindern, wenn eine Klasse in den Trainingsdaten nicht vorhanden ist.

Zur Klassierung eines neuen Auftrags a , der durch den Eigenschaftsvektor \vec{e}_a beschrieben wird, werden die zuvor ermittelten Parametermengen der bedingten und A-priori-Klassenwahrscheinlichkeiten verwendet und die Klassierung wird in zwei Phasen⁴²

³⁹Vgl. [Mit05], S. 6 f.; [WFH11], S. 94.

⁴⁰Hierbei handelt es sich um eine vereinfachende Schreibweise. Die Wahrscheinlichkeit dafür, dass X_p im Falle (quasi)stetiger Werte genau einen bestimmten Wert annimmt, ist null. Stattdessen wird davon ausgegangen, dass der Wert der Eigenschaft p in einem Intervall $[e_p - \epsilon/2, e_p + \epsilon/2]$ liegt: $P(e_p - \epsilon/2 \leq X_p \leq e_p + \epsilon/2) = \int_{e_p - \epsilon/2}^{e_p + \epsilon/2} f(e_p) de_p$ für $\lim_{\epsilon \rightarrow 0} P(e_p - \epsilon/2 \leq X_p \leq e_p + \epsilon/2)/\epsilon = f(e_p)$. Somit gilt $P(X_p = e_p) \approx f(e_p) \times \epsilon$. Im Zähler der Gleichung (5.13) tritt ϵ für jedes Cluster auf, sodass diese sich gegenseitig aufheben. Somit kann Gleichung (5.12) für die Wahrscheinlichkeiten der (quasi)stetigen Eigenschaften verwendet werden (siehe [JL95], S. 339; [WFH11], S. 96).

⁴¹Vgl. [Mit05], S. 6.

⁴²Vgl. [Bis06], S. 43 f.

durchgeführt. In der ersten Phase, der Inferenzphase, wird die A-posteriori-Wahrscheinlichkeitsverteilung über alle Cluster $\mathcal{C}^* \in \mathcal{C}^*$ ermittelt,⁴³ die angibt, mit welcher Wahrscheinlichkeit der Auftrag a zu einem Cluster gehört:⁴⁴

$$P(Y = \mathcal{C}_i^* | X_p = e_{a,p}) = \frac{P(Y = \mathcal{C}_i^*) \prod_{p=1}^d P(X_p = e_{a,p} | Y = \mathcal{C}_i^*)}{\sum_{j=1}^k P(Y = \mathcal{C}_j^*) \prod_{p=1}^d P(X_p = e_{a,p} | Y = \mathcal{C}_j^*)} \quad (5.13)$$

In der anschließenden Entscheidungsphase wird das Cluster mit der höchsten Zugehörigkeitswahrscheinlichkeit bzw. maximalen A-posteriori-Wahrscheinlichkeit ausgewählt. Zusammenfassend kann somit die Funktion `classify` mit `classify: \mathcal{A} \rightarrow \mathcal{C}^*` definiert werden, die für die Eingabe eines zu klassierenden Auftrags a jenes Cluster ausgibt, welches die höchste Zugehörigkeitswahrscheinlichkeit besitzt:⁴⁵

$$\text{classify}(a) = \arg \max_{\mathcal{C}_i^* \in \mathcal{C}^*} \left\{ P(Y = \mathcal{C}_i^*) \prod_{p=1}^d P(X_p = e_{a,p} | Y = \mathcal{C}_i^*) \right\} \quad (5.14)$$

Die für das Cluster ermittelte Intervallgrenze $I_{\mathcal{C}^*}$ sowie die empirische Wahrscheinlichkeitsfunktion $\hat{p}_{\mathcal{C}^*}(ha)$ bzw. relative Häufigkeitsverteilung $\mathcal{H}_{\mathcal{C}^*}$ werden in der anschließenden Ablaufplanung verwendet, um für den klassierten Auftrag a Informationen über die möglichen Bearbeitungszeiten zu generieren.

Das Vorgehen zur Klassierung eines einzuplanenden Auftrags a ist in Algorithmus 5.5 dargestellt. Zunächst wird in Zeile 3 überprüft, ob bereits ein trainierter Klassifikator vorliegt. Ist dies nicht der Fall, wird in Zeile 4 die Funktion `TrainClassifier` aufgerufen, um die entsprechenden Wahrscheinlichkeiten zu ermitteln. Die Funktion berechnet in den Zeilen 14 und 21 die bedingten Wahrscheinlichkeiten für die Eigenschaften 1 und 2 sowie in den Zeilen 9 und 16 die A-priori-Klassenwahrscheinlichkeiten. Der Erwartungswert und die Varianz für die (quasi)stetigen Eigenschaften werden in den Zeilen 12 und 19 bzw. den Zeilen 25 und 28 berechnet. Ein erneutes Training des Klassifikators wird durch Zeile 5 in Algorithmus 5.5 solange unterbunden, bis neue Trainingsdaten generiert wurden.⁴⁶ Schließlich wird Gleichung (5.14) in den Zeilen 6–13 ausgeführt und das Cluster mit der höchsten Zugehörigkeitswahrscheinlichkeit für Auftrag a ermittelt und zurückgegeben.

⁴³Bei dem naiven Bayes-Klassifikator, der hier verwendet wird, wird angenommen, dass die Eigenschaften X_p mit einem gegebenen Y bedingt unabhängig voneinander sind.

⁴⁴Vgl. [Mit05], S. 4.

⁴⁵Vgl. [Mit05], S. 4; [Mit97], S. 157.

⁴⁶Vergleiche Zeile 7 in Algorithmus 5.4 auf Seite 99.

Algorithmus 5.5: Klassifikator für Frachtabfertigungsaufträge

Input : a – Zu klassierender Auftrag
 \mathcal{T} – Menge der erstellten Trainingsdaten
 \mathcal{C}^* – Menge der ermittelten Cluster
Output : bestCluster – Ausgabe des Clusters mit der höchsten Zugehörigkeitswahrscheinlichkeit für den zu klassierenden Auftrag a

```

1 bestAPosterioriProb ← 0;
2 bestCluster ← null;
3 if trainedClassifier = false then
4   TrainClassifier( $\mathcal{T}, \mathcal{C}^*$ );
5   trainedClassifier = true;
6 foreach  $\mathcal{C}_i^* \in \mathcal{C}^*$  do
7   prod ← 1;
8   for  $p \leftarrow 1$  to  $d$  do
9     prod ← prod · P( $e_{a,p} | \mathcal{C}_i^*$ );
10  aPosterioriProb ← P( $\mathcal{C}_i^*$ ) · prod;
11  if aPosterioriProb ≥ bestAPosterioriProb then
12    bestAPosterioriProb ← aPosterioriProb;
13    bestCluster ←  $\mathcal{C}_i^*$ ;
14 return bestCluster;
```

Funktion TrainClassifier($\mathcal{T}, \mathcal{C}^*$)		
Input	: \mathcal{T} \mathcal{C}^*	– Menge der erstellten Trainingsdaten – Menge der ermittelten Cluster
Output	: $\hat{P}(e_p^j \mathcal{C}_i^*)$ $\hat{P}(\mathcal{C}_i^*)$ $\hat{\mu}_{p,i}$ $\hat{\sigma}_{p,i}^2$	– Bedingte Wahrscheinlichkeiten für diskrete Eigenschaften (hier: Eigenschaften 1 und 2) – A-priori-Klassenwahrscheinlichkeiten – Erwartungswerte der (quasi)stetigen Eigenschaften – Varianzen der (quasi)stetigen Eigenschaften

```

1 for i ← 1 to k do
2   for p ← 1 to d do
3     if p ≠ 1  $\mathcal{E}\mathcal{E}$  p ≠ 2 then  $\hat{\mu}_{p,i} \leftarrow 0$ ;  $\hat{\sigma}_{p,i}^2 \leftarrow 0$ ;
4     else
5       for j ← 1 to J do  $\hat{P}(e_p^j | \mathcal{C}_i^*) \leftarrow 0$ ;
6    $\hat{P}(\mathcal{C}_i^*) \leftarrow 0$ ; freqClass $_{\mathcal{C}_i^*} \leftarrow 0$ ;
7 foreach td ∈  $\mathcal{T}$  do
8   for i ← 1 to k do
9     freqClass $_{\mathcal{C}_i^*} \leftarrow$  freqClass $_{\mathcal{C}_i^*} + 1 \cdot \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*)$ ;
10    for p ← 1 to d do
11      if p ≠ 1  $\mathcal{E}\mathcal{E}$  p ≠ 2 then
12         $\hat{\mu}_{p,i} \leftarrow \hat{\mu}_{p,i} + e_{td,p} \cdot \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*)$ ;
13      else
14         $\hat{P}(e_p^j | \mathcal{C}_i^*) \leftarrow \hat{P}(e_p^j | \mathcal{C}_i^*) + 1 \cdot \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*) \cdot \delta(e_{td,p} = e_p^j)$ ;
15 for i ← 1 to k do
16    $\hat{P}(\mathcal{C}_i^*) \leftarrow$  freqClass $_{\mathcal{C}_i^*} + \alpha / |T| + \alpha |\mathcal{C}^*|$ ;
17   for p ← 1 to d do
18     if p ≠ 1  $\mathcal{E}\mathcal{E}$  p ≠ 2 then
19        $\hat{\mu}_{p,i} \leftarrow \hat{\mu}_{p,i} / \text{freqClass}_{\mathcal{C}_i^*}$ ;
20     else
21       for j ← 1 to J do  $\hat{P}(e_p^j | \mathcal{C}_i^*) \leftarrow \hat{P}(e_p^j | \mathcal{C}_i^*) + \alpha / \text{freqClass}_{\mathcal{C}_i^*} + \alpha J$ ;
22 foreach td ∈  $\mathcal{T}$  do
23   for i ← 1 to k do
24     for p ← 1 to d do
25       if p ≠ 1  $\mathcal{E}\mathcal{E}$  p ≠ 2 then  $\hat{\sigma}_{p,i}^2 \leftarrow \hat{\sigma}_{p,i}^2 + (e_{td,p} - \hat{\mu}_{p,i}) \cdot \delta(\mathcal{C}_{td}^* = \mathcal{C}_i^*)$ ;
26 for i ← 1 to k do
27   for p ← 1 to d do
28     if p ≠ 1  $\mathcal{E}\mathcal{E}$  p ≠ 2 then  $\hat{\sigma}_{p,i}^2 \leftarrow \hat{\sigma}_{p,i}^2 / \text{freqClass}_{\mathcal{C}_i^*} - 1$ ;

```

5.3 Multikriterielle Ablaufplanung der Abfertigungsprozesse

Aufbauend auf der in den vorherigen Abschnitten aufbereiteten Informationen über historische Frachtabfertigungsaufträge (vgl. Abschnitt 5.2) und unter Verwendung des erstellten Klassifikators für neue Abfertigungsaufträge wird in den folgenden Abschnitten eine proaktive Ablaufplanung entwickelt. Mit Hilfe des Konzepts soll die automatisierte Erstellung von Ablaufplänen ermöglicht werden, die Informationen über Unsicherheiten der Bearbeitungszeiten von Frachtabfertigungsaufträgen systematisch berücksichtigt, wodurch Ablaufpläne entworfen werden können, die im laufenden Betrieb weniger anfällig gegenüber Störungen sind.

Im Folgenden wird das Grundproblem der proaktiven Ablaufplanung zunächst in Abschnitt 5.3.1 als Mehrzieloptimierungsmodell mathematisch formuliert. Dieses Modell wird anschließend in Abschnitt 5.3.2 in eine rollierende Planung integriert, die dem offenen Planungshorizont Rechnung trägt. Des Weiteren wird das Zielsystem des Grundproblems um die Betrachtung eines Stabilitätskriteriums erweitert. Aufgrund der Komplexität des Problems wird eine evolutionäre Heuristik in Abschnitt 5.3.3 entwickelt. Aufbauend auf dem erweiterten Grundmodell wird dieses in den Abschnitten 5.3.4 und 5.3.5 um die Betrachtung weiterer Einschränkungen in Form des Personaleinsatzes und der Förderkapazitäten erweitert.

5.3.1 Formalisierung des Ablaufplanungsproblems

Im Folgenden soll das Problem der proaktiven Ablaufplanung als Optimierungsmodell mathematisch formalisiert werden. Hierzu werden in Abschnitt 5.3.1.1 zunächst die einzuplanenden Aufträge $a \in \mathcal{A}$ klassiert. Anschließend wird in Abschnitt 5.3.1.2 auf das zu verwendende Zielsystem näher eingegangen und schließlich in Abschnitt 5.3.1.3 das Optimierungsmodell aufgestellt.

5.3.1.1 Klassierung der zu bearbeitenden Aufträge

Grundlage der Optimierung bilden die zu einem Zeitpunkt bekannten und einzuplanenden Aufträge der Menge \mathcal{A} .⁴⁷ Bevor diese auf die zur Verfügung stehenden Maschinen eingeplant werden können, sind zunächst mit Hilfe des in Abschnitt 5.2.2.2 trainierten Klassifikators für jeden Auftrag eine Klassierung durchzuführen, um diese einem der Cluster zuzuordnen. Mit Hilfe der für die Cluster hinterlegten Intervallgrenzen der Bearbeitungszeitabweichungen können anschließend die Bearbeitungszeitintervalle der einzuplanenden Aufträge berechnet werden.

Zur Klassierung wird für jeden Auftrag $a \in \mathcal{A}$ mit Hilfe von Gleichung (5.14) eine Klassierung durchgeführt. Ergebnis dieser Klassierung sind ein Cluster \mathcal{C}^* sowie das für dieses Cluster hinterlegte Intervall $I_{\mathcal{C}^*}$ sowie die relative Häufigkeitsverteilung $\mathcal{H}_{\mathcal{C}^*}$

⁴⁷Vgl. Abschnitt 2.2.2.2. In Abschnitt 5.3.1 wird zunächst noch von einem statischen System ausgegangen. Diese Einschränkung wird in Abschnitt 5.3.2 aufgehoben.

Algorithmus 5.6: Klassierung der einzuplanenden Frachtabfertigungsaufträge und Ermittlung der Bearbeitungszeitintervalle

Input : \mathcal{A} – Zu klassierende Aufträge
 \mathcal{C}^* – Menge der ermittelten Cluster
Output : I_a – Bearbeitungszeitintervall des klassierten Auftrags a

```

1 foreach  $a \in \mathcal{A}$  do
2    $\hat{\tau}_a \leftarrow$  Schätze die Bearbeitungszeit mit Gleichung (2.1);
3   bestCluster  $\leftarrow$  classify( $a$ );
4    $\tau_a^{\min} \leftarrow \hat{\tau}_a + \Delta\tau_{\text{bestCluster}}^{\min} \cdot \hat{\tau}_a$ ;
5    $\tau_a^{\max} \leftarrow \hat{\tau}_a + \Delta\tau_{\text{bestCluster}}^{\max} \cdot \hat{\tau}_a$ ;
6    $I_a \leftarrow [\tau_a^{\min}, \tau_a^{\max}]$ ;
7 return  $I_a$ ;

```

der Bearbeitungszeitabweichungen. Die minimale Bearbeitungszeit eines Auftrags a lässt sich über Gleichung (5.15) berechnen. Die über z. B. Gleichung (2.1) geschätzte Bearbeitungszeit $\hat{\tau}_a$ wird hierzu mit dem minimalen Wert $\Delta\tau_{\mathcal{C}^*}^{\min}$ des Intervalls multipliziert und das Ergebnis wird zu der geschätzten Bearbeitungszeit addiert.

$$\tau_a^{\min} = \hat{\tau}_a + \Delta\tau_{\mathcal{C}^*}^{\min} \cdot \hat{\tau}_a \quad (5.15)$$

Analog wird die maximale Bearbeitungszeit eines Auftrags a ermittelt:

$$\tau_a^{\max} = \hat{\tau}_a + \Delta\tau_{\mathcal{C}^*}^{\max} \cdot \hat{\tau}_a$$

Das Ergebnis der Berechnung ist das Bearbeitungszeitintervall $I_a = [\tau_a^{\min}, \tau_a^{\max}]$ mit $0 < \tau_a^{\min} \leq \tau_a^{\max}$ für jeden Auftrag $a \in \mathcal{A}$.

Für die Ermittlung einer Bearbeitungszeit auf Basis der Häufigkeitsverteilung $\mathcal{H}_{\mathcal{C}^*}$ wird zunächst eine diskrete Zufallsvariable X definiert, die bezüglich dieser Häufigkeitsverteilung verteilt ist $X \sim \mathcal{H}_{\mathcal{C}^*}$. Das bedeutet: X nimmt mit einer Wahrscheinlichkeit von $q_{TI_i^{\mathcal{C}^*}}$ die mittlere⁴⁸ Bearbeitungszeitabweichung der jeweiligen Klasse $TI_i^{\mathcal{C}^*}$ an:

$$\Delta\tau_{\mathcal{C}^*} = lb_{TI_i^{\mathcal{C}^*}} + \frac{ub_{TI_i^{\mathcal{C}^*}} - lb_{TI_i^{\mathcal{C}^*}}}{2}$$

Die ermittelte Bearbeitungszeitabweichung kann anschließend verwendet werden, um eine Bearbeitungszeit für Auftrag a zu ermitteln:

$$\tau_a = \hat{\tau}_a + \Delta\tau_{\mathcal{C}^*} \cdot \hat{\tau}_a \quad (5.16)$$

Die Klassierung und Ermittlung der Bearbeitungszeitintervalle aller einzuplanenden Aufträge ist in Algorithmus 5.6 dargestellt. Zunächst wird in Zeile 2 die Bearbeitungszeit des Auftrags a geschätzt. Der Auftrag wird anschließend klassiert und das Cluster mit

⁴⁸Die Bearbeitungszeitabweichungen innerhalb einer Klasse seien gleichverteilt.

der höchsten Zugehörigkeitswahrscheinlichkeit ermittelt (Zeile 3). Aus den hinterlegten Intervallgrenzen der Bearbeitungszeitabweichungen wird schließlich in den Zeilen 4 und 5 die minimale und maximale Bearbeitungszeit des Auftrags ermittelt.

5.3.1.2 Zielsystem

Mit der proaktiven Ablaufplanung wird das Ziel verfolgt, die einzuplanenden Aufträge so auf die vorhandenen Maschinen zu verteilen, dass möglichst viele der Aufträge ihren Fälligkeitstermin einhalten, d. h. die Anzahl säumiger Aufträge minimiert wird. Zudem soll berücksichtigt werden, dass die für einen Auftrag benötigte Bearbeitungszeit nicht mit Sicherheit bekannt ist und daher als unsicher anzunehmen ist, d. h. die Robustheit einer Lösung soll maximiert werden.⁴⁹ Somit ergeben sich zwei Zielfunktionen, die im Folgenden näher erläutert werden.

Die erste Zielfunktion, die in dem Zielsystem berücksichtigt werden soll, ist die Minimierung der erwarteten gewichteten Anzahl säumiger Aufträge. Für die Formalisierung der Zielfunktion wird im Folgenden das Konzept von Szenarien eingeführt. Ein Szenario $sz = \{\tau_{a_1}^{sz}, \tau_{a_2}^{sz}, \dots, \tau_{a_n}^{sz}\}$ ist eine mögliche Realisierung der Bearbeitungszeiten der Aufträge. Mit $\tau_{a_j}^{sz} \in I_{a_j} = [\tau_{a_j}^{\min}, \tau_{a_j}^{\max}]$ wird die Bearbeitungszeit des Auftrags a_j in Szenario sz bezeichnet. Die Menge aller Szenarien \mathcal{SZ} entspricht dem kartesischen Produkt der $|\mathcal{A}|$ Intervalle I_a und ist somit unendlich. Mit Hilfe dieser Menge wird die Unsicherheit der Bearbeitungszeiten abgebildet.

Ein Auftrag a in Szenario sz ist somit säumig, wenn sein Fertigstellungszeitpunkt in diesem Szenario, wie in Gleichung (2.2) auf Seite 23 beschrieben, größer als der Fälligkeitstermin für diesen Auftrag ist. Erweitert um die Betrachtung von Szenarien bezeichne hierfür $U(ap, sz)$ die gewichtete Anzahl säumiger Aufträge, wenn der Ablaufplan ap unter Szenario sz ausgeführt wird. Die Indikatorfunktion $U_a^{sz}(ap)$ gibt an, ob der Auftrag a im Ablaufplan ap unter Szenario sz säumig ist. In diesem Fall ist der Fertigstellungszeitpunkt $C_a^{sz}(ap)$ des Auftrags a im Ablaufplan ap für Szenario sz größer als der Fälligkeitstermin für diesen Auftrag:

$$U(ap, sz) = \sum_{a \in \mathcal{A}} w_a U_a^{sz}(ap) \quad (5.17)$$

und

$$U_a^{sz}(ap) = \begin{cases} 1, & \text{wenn } C_a^{sz}(ap) > d_a \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

Zu dem Zeitpunkt, zu dem die Ablaufplanung erfolgt, sind die durch säumige Aufträge entstehenden Kosten die einzigen, auf die durch die Planung noch eingewirkt werden kann. Über die terminorientierte Zielfunktion⁵⁰ wird das Ziel verfolgt, eine möglichst hohe

⁴⁹Vgl. Abschnitt 2.2.2.2.

⁵⁰Im Hinblick auf die für den Kunden höheren Transportkosten für den Luftfrachtverkehr im Vergleich zu anderen Transportmöglichkeiten, wie z. B. Schiffsgüterverkehr, ist die Geschwindigkeit und Zuverlässigkeit ein wichtiger Aspekt und damit die Termintreue ein entscheidender Grund für die Wahl des Kunden.

Termintreue für den Kunden einzuhalten. Eine leichte Überschreitung des festgelegten Fälligkeitstermins kann in der Praxis dazu führen, dass eine Sendung den Flug, auf den sie gebucht ist, nicht erreicht.⁵¹ Dementsprechend wird in der Zielfunktion bereits eine geringe Überschreitung mit den maximalen Kosten⁵² belegt. Die Berechnung der erwarteten gewichteten Anzahl säumiger Aufträge für einen gegebenen Ablaufplan ap erfolgt mit Hilfe von Gleichung (5.18):

$$c^{\text{sched}}(ap) = \frac{\sum_{sz \in \mathcal{SZ}} U(ap, sz)}{|\mathcal{SZ}|} \quad (5.18)$$

Die erste Zielfunktion ergibt sich somit, indem das Minimum für Gleichung (5.18) für alle Ablaufpläne $ap \in \mathcal{AP}$, mit \mathcal{AP} als Menge aller zulässigen Ablaufpläne, ermittelt wird:

$$\min_{ap \in \mathcal{AP}} c^{\text{sched}}(ap) \quad (5.19)$$

Aufgrund der bestehenden Unsicherheit hinsichtlich der Bearbeitungsdauer für einen Auftrag sind der unter Berücksichtigung der zuvor beschriebenen Zielfunktion ermittelte Ablaufplan und dessen Zielfunktionswert nicht mit Sicherheit zu realisieren. Daher wird mit der zweiten Zielfunktion die Eigenschaft eines Ablaufplans bewertet, für möglichst viele mögliche Umweltsituationen ein gutes Ergebnis zu liefern, oder anders ausgedrückt: Die Robustheit eines Ablaufplans wird quantifiziert. Wie in Kapitel 4 bereits angesprochen, soll für diese Zielfunktion ein Regret-basiertes Zielkriterium verwendet werden, bei dem die Möglichkeit besteht, den Grad der Robustheit durch Parameter zu beeinflussen und damit, wie konservativ eine Lösung ist.

Zur Berechnung des absoluten Regrets für ein Szenario sz wird der beste Zielfunktionswert für dieses Szenario benötigt.⁵³ Bezeichne hierfür U_{sz}^* die minimale Anzahl säumiger Aufträge unter Szenario sz und ap_{sz}^* den optimalen Ablaufplan, der zu diesem Ergebnis führt:

$$U_{sz}^* = U(ap_{sz}^*, sz) = \min_{ap \in \mathcal{AP}} U(ap, sz) \quad (5.20)$$

Für einen gegebenen Ablaufplan $ap \in \mathcal{AP}$ lässt sich der absolute Regret für Szenario sz in Bezug auf die Gesamtanzahl zu planender Aufträge anschließend folgendermaßen berechnen:

$$R(ap, sz) = \frac{U(ap, sz) - U_{sz}^*}{|\mathcal{A}|} \quad (5.21)$$

Der maximale absolute Regret für einen Ablaufplan ap ergibt sich, indem das Maximum für Gleichung (5.21) für alle $sz \in \mathcal{SZ}$ ermittelt wird:

$$R^{\text{max}}(ap) = \max_{sz \in \mathcal{SZ}} R(ap, sz)$$

⁵¹Vgl. Abschnitt 2.2.

⁵²Statt einheitlicher Strafkosten für jeden säumigen Auftrag könnten auch die tatsächlich entstehenden Kosten Verwendung finden, sofern diese zum Zeitpunkt der Planung durch bspw. Verzugskosten in den Beförderungsverträgen bereits bekannt sind.

⁵³Vgl. [XCLQ13], S. 3.

Das Szenario, das zur Maximierung des absoluten Regrets für einen bestimmten Ablaufplan ap führt, wird als Worst-Case-Szenario sz_{ap}^{worst} bezeichnet:

$$sz_{ap}^{\text{worst}} = \arg \max_{sz \in \mathcal{SZ}} R(ap, sz)$$

Des Weiteren werden für die Zielfunktion zwei Parameter festgelegt, mit denen der Grad der Robustheit beeinflusst werden kann:⁵⁴

- ω : Dieser Parameter bezeichnet einen Schwellenwert für den in Gleichung (5.21) angegebenen absoluten Regret, der für jede Realisierung der Bearbeitungszeiten bzw. jedes Szenario durch den Ablaufplan ap nicht überschritten werden darf. Somit gilt:

$$R(ap, sz) \leq R^{\max}(ap) \leq \omega \quad \forall sz \in \mathcal{SZ} \quad (5.22)$$

- φ : Mit diesem Parameter wird ein Schwellenwert für den in Gleichung (5.21) berechneten absoluten Regret festgelegt, der für möglichst viele Realisierungen der Bearbeitungszeiten bzw. Szenarien nicht überschritten werden soll. Es gilt $\varphi \leq \omega$ und für einen gegebenen Ablaufplan ap :

$$ar_{\varphi}^{\text{sched}}(ap) = |\{sz \mid sz \in \mathcal{SZ} \wedge R(ap, sz) \leq \varphi\}|$$

Der vorherige Zielfunktionswert kann ebenfalls als relative Größe $r_{\varphi}^{\text{sched}}$ angegeben werden:⁵⁵

$$r_{\varphi}^{\text{sched}}(ap) = \frac{|\{sz \mid sz \in \mathcal{SZ} \wedge R(ap, sz) \leq \varphi\}|}{|\mathcal{SZ}|} \quad (5.23)$$

Die zweite Zielfunktion ergibt sich somit, indem das Maximum für Gleichung (5.23) für alle $ap \in \mathcal{AP}$ ermittelt wird:

$$\max_{ap \in \mathcal{AP}} r_{\varphi}^{\text{sched}}(ap) \quad (5.24)$$

Die Definition der zuvor genannten Parameter ermöglicht es, die Robustheit der gefundenen Lösung zu steuern. Über den Parameter ω wird bestimmt, welchen absoluten Regret ein Ablaufplan für jedes Szenario maximal besitzen darf, und dies entspricht somit einer garantierten oberen Grenze für den absoluten Regret. Eine Erhöhung dieses Parameters (für einen konstanten Wert des Parameters φ) kann dazu führen, dass mehr Ablaufpläne gefunden werden, die den Schwellenwert φ für eine größere Anzahl an Szenarien einhalten. Eine Verringerung des Parameters verringert den Verlust im Worst Case, reduziert aber gleichzeitig die Möglichkeit, Lösungen zu finden, die zwar einen höheren maximalen absoluten Regret besitzen, aber dafür einen geringeren absoluten Regret für eine größere Anzahl von Szenarien.

⁵⁴Vgl. [Roy10].

⁵⁵Vgl. ebd., S. 634.

5.3.1.3 Modellformulierung

Auf Basis der für jeden Auftrag ermittelten Intervallgrenzen der Bearbeitungszeitabweichungen in Abschnitt 5.3.1.1 und dem beschriebenen Zielsystem in Abschnitt 5.3.1.2 kann das Mehrzieloptimierungsmodell (5.25)⁵⁶ für eine statische Umgebung, bei der die Menge \mathcal{A} vollständig bekannt ist, formalisiert werden. Ergebnis des Modells sind die Ablaufpläne (repräsentiert über die Entscheidungsvariable $x_{a,a',m}$), die hinsichtlich des verwendeten Zielsystems pareto-optimal sind.

Mengen:

\mathcal{A}	Menge der einzuplanenden Aufträge a
$\mathcal{A}_a^{\text{prec}}$	Menge aller Aufträge, die vor Auftrag a eingeplant werden können
$\mathcal{A}_a^{\text{succ}}$	Menge aller Aufträge, die nach Auftrag a eingeplant werden können
\mathcal{E}	Menge von Vorrangbeziehungen zwischen den Aufträgen
\mathcal{M}	Menge der zur Verfügung stehenden Maschinen m
\mathcal{M}_a	Menge von Maschinen, auf denen der Auftrag a ausgeführt werden kann
\mathcal{SZ}	Menge aller Szenarien sz , die eine mögliche Realisierung der Bearbeitungszeiten der Aufträge angeben

Parameter:

d_a	Fälligkeitstermin von Auftrag $a \in \mathcal{A}$
φ	Schwellenwert, der für möglichst viele Szenarien $sz \in \mathcal{SZ}$ unterschritten werden sollte
M	Big M
ω	Schwellenwert, der für kein Szenario $sz \in \mathcal{SZ}$ überschritten werden darf
ρ	Pseudoauftrag, dessen Bearbeitungszeit, Bereitstellungszeit sowie Gewicht 0 ist. Es gilt: $\rho \in \mathcal{A}$ sowie $\forall a \in \mathcal{A}: \rho \in \mathcal{A}_a^{\text{prec}} \wedge \rho \in \mathcal{A}_a^{\text{succ}}$
r_a	Bereitstellungszeit von Auftrag $a \in \mathcal{A}$
τ_a^{sz}	Bearbeitungszeit des Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
U_{sz}^*	Minimale Anzahl säumiger Aufträge unter Szenario $sz \in \mathcal{SZ}$
w_a	Priorität von Auftrag $a \in \mathcal{A}$

Entscheidungsvariablen:

C_a^{sz}	Fertigstellungszeitpunkt des Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
q	Erwartete gewichtete Anzahl säumiger Aufträge für alle Szenarien
U_a^{sz}	Binär: 1, wenn Auftrag $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$ säumig ist; 0 in allen anderen Fällen
v	Erwartete Robustheit für alle Szenarien
$x_{a,a',m}$	Binär: 1, wenn Auftrag $a \in \mathcal{A}$ direkt vor Auftrag $a' \in \mathcal{A}$ auf Maschine $m \in \mathcal{M}$ ausgeführt wird; 0 in allen anderen Fällen
z_{sz}	Binär: 1, wenn der absolute Regret säumiger Aufträge in Szenario sz kleiner als φ ist; 0 in allen anderen Fällen

⁵⁶Die Grundlage dieses Modells bildet eine Netzwerkformulierung, die ein kapazitiv beschränktes Tourenplanungsproblem (engl. *Capacitated Vehicle Routing Problem*) beschreibt. Hierbei stellen die Kunden die Aufträge, die Fahrzeuge die Maschinen und eine Tour einen Ablaufplan für eine Maschine dar (vgl. z. B. [UM10], S. 788 oder [CM07], S. 221).

Zielfunktionen:

$$\min c^{\text{sched}} = q \quad (5.25a)$$

$$\max r_{\varphi}^{\text{sched}} = v \quad (5.25b)$$

u. B. d. N.:

$$\frac{\sum_{sz \in \mathcal{SZ}} \sum_{a \in \mathcal{A}} w_a U_a^{sz}}{|\mathcal{SZ}|} \leq q \quad (5.25c)$$

$$\frac{\sum_{sz \in \mathcal{SZ}} z_{sz}}{|\mathcal{SZ}|} = v \quad (5.25d)$$

$$\frac{(\sum_{a \in \mathcal{A}} w_a U_a^{sz}) - U_{sz}^*}{|\mathcal{A}|} \leq \omega(1 - z_{sz}) + \varphi z_{sz}, \quad \forall sz \in \mathcal{SZ} \quad (5.25e)$$

$$\sum_{a \in \mathcal{A}: a \neq \rho} x_{a,\rho,m} \leq 1, \quad \forall m \in \mathcal{M} \quad (5.25f)$$

$$\sum_{a' \in \mathcal{A}: a' \neq \rho} x_{\rho,a',m} \leq 1, \quad \forall m \in \mathcal{M} \quad (5.25g)$$

$$\sum_{m \in \mathcal{M}_{a'}} \sum_{a \in \mathcal{A}_a^{\text{prec}}: a \neq a'} x_{a,a',m} = 1, \quad \forall a' \in \mathcal{A}: a' \neq \rho \quad (5.25h)$$

$$\sum_{m \in \mathcal{M}_a} \sum_{a' \in \mathcal{A}_{a'}^{\text{succ}}: a' \neq a} x_{a,a',m} = 1, \quad \forall a \in \mathcal{A}: a \neq \rho \quad (5.25i)$$

$$\sum_{a \in \mathcal{A}_{a'}^{\text{prec}}: a \neq a'} x_{a,a',m} - \sum_{a \in \mathcal{A}_{a'}^{\text{succ}}: a \neq a'} x_{a',a,m} = 0, \quad \forall a' \in \mathcal{A}: a' \neq \rho; \quad \forall m \in \mathcal{M} \quad (5.25j)$$

$$C_a^{sz} - C_{a'}^{sz} + (M - r_{a'})x_{a,a',m} \leq M - (r_{a'} + \tau_{a'}^{sz}), \quad \forall a \in \mathcal{A}; \forall a' \in \mathcal{A}: \quad (5.25k)$$

$$a' \neq \rho, a \neq a'; \quad \forall m \in \mathcal{M}; \forall sz \in \mathcal{SZ}$$

$$C_{a_i}^{sz} + \tau_{a_j}^{sz} \leq C_{a_j}^{sz}, \quad \forall (a_i, a_j) \in \mathcal{E}; \quad \forall sz \in \mathcal{SZ} \quad (5.25l)$$

$$C_a^{sz} - MU_a^{sz} \leq d_a, \quad \forall a \in \mathcal{A}; \forall sz \in \mathcal{SZ} \quad (5.25m)$$

$$x_{a,a',m} \in \{0, 1\}, \quad \forall a \in \mathcal{A}; \forall a' \in \mathcal{A}; \quad \forall m \in \mathcal{M} \quad (5.25n)$$

$$U_a^{sz} \in \{0, 1\}, \quad \forall a \in \mathcal{A}; \forall sz \in \mathcal{SZ} \quad (5.25o)$$

$$z_{sz} \in \{0, 1\}, \quad \forall sz \in \mathcal{SZ} \quad (5.25p)$$

$$C_a^{sz} \geq 0, \quad \forall a \in \mathcal{A}; \forall sz \in \mathcal{SZ} \quad (5.25q)$$

$$q \geq 0 \quad (5.25r)$$

$$v \geq 0 \quad (5.25s)$$

Die Zielfunktion (5.25a) in Kombination mit Restriktion (5.25c) minimiert die erwartete gewichtete Anzahl säumiger Aufträge. Die Zielfunktion (5.25b) in Kombination mit den Restriktionen (5.25d) und (5.25e) maximiert die Robustheit der Lösung. Hierzu wird mit Hilfe der Entscheidungsvariable z_{sz} entschieden, ob die absolute Abweichung der Anzahl säumiger Aufträge (modelliert über die Entscheidungsvariable U_a^{sz}) im Vergleich zum Optimum für das Szenario (dargestellt durch U_{sz}^*) den Schwellenwert ω oder φ in Szenario sz einhält. Wenn $z_{sz} = 0$, wird über die Restriktion (5.25e) lediglich die Einhaltung des Schwellenwerts ω gefordert, für $z_{sz} = 1$ jedoch die Einhaltung des Schwellenwerts φ , der wegen $\varphi \leq \omega$ einer stärkeren Einschränkung entspricht. Die erwartete Robustheit wird schließlich über Gleichung (5.25d) ermittelt. Die Nebenbedingungen (5.25f) und (5.25g) stellen sicher, dass für jede Maschine höchstens ein Auftrag direkt vor bzw. direkt nach dem Pseudoauftrag ρ eingeplant wird.⁵⁷ Die Gleichungen (5.25h) und (5.25i) sorgen dafür, dass jeder Auftrag auf genau einer passenden Maschine eingeplant wird und hierbei die Vorrangbeziehungen berücksichtigt werden. Restriktion (5.25j) stellt sicher, dass jeder Auftrag (ausgenommen dem Pseudoauftrag) genau einen Vorgänger und einen Nachfolger besitzt. Der Fertigstellungszeitpunkt C_a^{sz} eines Auftrags a in Szenario sz wird mit Hilfe der Restriktion (5.25k) ermittelt. Über die Ungleichung (5.25l) wird sichergestellt, dass ein Auftrag, für den eine Vorrangbeziehung besteht, erst nach der Fertigstellung des Vorgängerauftrags starten kann.⁵⁸ Die Einhaltung des für einen Auftrag bestehenden Fälligkeitstermins wird über die Ungleichung (5.25m) beschrieben. Liegt der Fertigstellungszeitpunkt C_a^{sz} für einen Auftrag a in Szenario sz später als der entsprechende Fälligkeitstermin d_a für diesen Auftrag, so wird die Säumigkeit für diesen Auftrag über die Entscheidungsvariable U_a^{sz} modelliert. Die binäre Eigenschaft der Entscheidungsvariablen wird über die Nebenbedingungen (5.25n)–(5.25p) sowie die Nichtnegativität der Entscheidungsvariablen über die Restriktionen (5.25q)–(5.25s) sichergestellt.

Das Modell (5.25) enthält aufgrund der unendlichen Anzahl von Szenarien $|\mathcal{SZ}| = \infty$ eine unendliche Anzahl von Restriktionen. Aus diesem Grund und da mehrere Zielfunktionen vorhanden sind, kann das Modell nicht direkt durch einen Solver gelöst werden. In Abschnitt 5.3.3.3 wird näher darauf eingegangen, wie diese Menge reduziert und somit eine Lösung mit Hilfe einer evolutionären Metaheuristik berechnet werden kann. Zuvor wird im nächsten Abschnitt das beschriebene Grundmodell (5.25) in eine rollierende Planung integriert und um eine weitere Zielfunktion zur Beurteilung der Stabilität von Ablaufplänen erweitert.

5.3.2 Aufbau einer rollierenden Ablaufplanung

Das im vorherigen Abschnitt aufgebaute Modell ist für ein statisches Planungsumfeld verwendbar, in dem die Informationen über die einzuplanenden Aufträge \mathcal{A} zu Beginn der Planung vollständig bekannt sind. Für eine realistische Betrachtung ist jedoch, wie in Abschnitt 2.2.2.2 beschrieben, von einer dynamischen Planungsumgebung auszugehen,

⁵⁷Sind auf einer Maschine keine Aufträge eingeplant, so besitzt der Pseudoauftrag für diese Maschine keinen Vorgänger oder Nachfolger.

⁵⁸Vgl. Abschnitt 2.2.2.2.

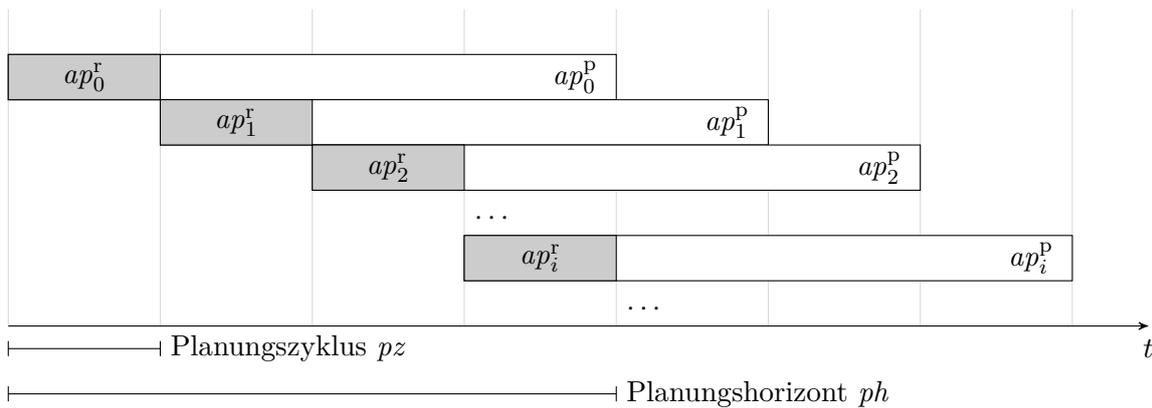


Abbildung 5.4: Rollierende Ablaufplanung

bei der die Menge \mathcal{A} unbegrenzt ist. Zu jedem Planungszeitpunkt t_i ist jeweils nur eine Teilmenge $\mathcal{A}_{t_i} \subset \mathcal{A}$ von Aufträgen bekannt.

Im Folgenden wird daher das aufgebaute Modell um die Betrachtung der dynamischen Planungsumgebung erweitert. Hierzu wird in Abschnitt 5.3.2.1 zunächst der Ablauf einer rollierenden Ablaufplanung erläutert und auf die Besonderheiten im Zusammenhang mit den verschiedenen Auftragsmengen näher eingegangen. Anschließend wird das beschriebene Zielsystem um das Zielkriterium der Stabilität in Abschnitt 5.3.2.2 erweitert und schließlich in Abschnitt 5.3.2.3 Modell (5.25) den Anforderungen entsprechend angepasst.

5.3.2.1 Ablauf der rollierenden Ablaufplanung

Als Grundlage für die Planung in der dynamischen Umgebung soll, wie in Abschnitt 4.2 erläutert, eine rollierende Ablaufplanung dienen. Bei dieser wird periodisch⁵⁹ das Ablaufplanungsproblem für die bekannten Aufträge gelöst. Das Vorgehen ist in Abbildung 5.4 skizziert. Zu einem Zeitpunkt t_i wird die Ablaufplanung mit allen zu dem Zeitpunkt planbaren Aufträgen aus $\mathcal{A}_{t_i}^{\text{Pb}} = \mathcal{A}_{t_i}^{\text{N}} \cup \mathcal{A}_{t_i}^{\text{P}}$ ausgeführt.⁶⁰ Das Ende des so erstellten Ablaufplans ap_i^{P} stellt den Planungshorizont ph dar, der in Abhängigkeit des tatsächlich eintretenden Szenarios zum Zeitpunkt t_i nicht genau bestimmt werden kann. Mit dem Planungszyklus pz wird der zeitliche Abstand festgelegt, der zwischen zwei Planzeitpunkten liegt. Die nächste Planung erfolgt somit zum Zeitpunkt $t_{i+1} = t_i + pz$. Die im Zeitraum $[t_i, t_{i+1}]$ eingetroffenen Aufträge werden durch die ausschließlich rollierende Planung erst zum Zeitpunkt t_{i+1} betrachtet.⁶¹ Die im Ablaufplan ap_i^{P} getroffenen Entscheidungen können im Ablaufplan ap_{i+1}^{P} , der zum Zeitpunkt t_{i+1} erstellt wird, für alle Aufträge der Menge $\mathcal{A}_{t_{i+1}}^{\text{P}}$ revidiert werden.⁶² Die Zuordnungen für Aufträge der Mengen $\mathcal{A}_{t_{i+1}}^{\text{F}}$ und $\mathcal{A}_{t_{i+1}}^{\text{R}}$ hingegen sind fix bzw. werden aktuell ausgeführt, ferner sind

⁵⁹Eine Erweiterung um eine ereignisorientiert angestoßene Ablaufplanung (vgl. Abschnitt 2.2.3) erfolgt in Abschnitt 5.4.

⁶⁰Vgl. Abschnitt 2.2.2.2.

⁶¹Vgl. [CU92], S. 156.

⁶²Vgl. Abschnitt 2.2.2.2.

Aufträge der Menge $\mathcal{A}_{t_{i+1}}^C$ bereits fertiggestellt. Von dem initial erstellten Ablaufplan ap_i^P wird somit nur ein Teil ap_i^F ausgeführt, die weiteren Entscheidungen werden in den nachfolgenden Zyklen ggf. mehrfach revidiert.

Die Planung der Aufträge zum Zeitpunkt t_i wird nicht nur durch die Aufträge der Menge $\mathcal{A}_{t_i}^{Pb}$ bestimmt, sondern unterliegt zusätzlichen Restriktionen durch alle noch nicht beendeten Aufträge des vorherigen Planungszyklus. Diese Aufträge werden entweder auf einer Maschine ausgeführt oder sind auf eine Maschine fixiert und beeinflussen somit die zur Verfügung stehenden Kapazitäten in der Art, dass die Aufträge des aktuellen Planungszeitraums nur nach diesen ausgeführt werden können.

Des Weiteren weicht für Aufträge der Menge $\mathcal{A}_{t_i}^R$ die in Optimierungsmodell (5.30) in Abschnitt 5.3.2.3 zu betrachtende Szenariomenge von der in Abschnitt 5.3.1.2 beschriebenen ab. Für bereits gestartete Aufträge umfasst die mögliche Bearbeitungszeit nicht mehr das vollständige Intervall I_a , sondern nur das mögliche Restbearbeitungszeitintervall I_a^{Rest} :

$$I_a^{\text{Rest}} = \left[\max(\tau_a^{\min} - (t_i - s_a), 0), \max(\tau_a^{\max} - (t_i - s_a), 0) \right] \quad (5.26)$$

Dieses errechnet sich, indem von der minimal τ_a^{\min} und maximal τ_a^{\max} möglichen Bearbeitungszeit aus dem Intervall I_a der Zeitraum $(t_i - s_a)$ (= Differenz aus dem aktuellem Zeitpunkt t_i und dem tatsächlichen Startzeitpunkt s_a des Auftrags) subtrahiert wird. Die Bearbeitungszeit eines Auftrags $a \in \mathcal{A}_{t_i}^R$ in Szenario sz ergibt sich somit aus $\tau_a^{sz} \in I_a^{\text{Rest}}$.

Analog muss für Aufträge der Menge $\mathcal{A}_{t_i}^R$ die Ermittlung von Bearbeitungszeiten auf Basis der jeweiligen Häufigkeitsverteilung \mathcal{H}_{C^*} angepasst werden. Hierzu wird Gleichung (5.16) entsprechend geändert:

$$\tau_a^{\text{Rest}} = \max((\hat{\tau}_a + \Delta\tau_{C^*} \cdot \hat{\tau}_a) - (t_i - s_a), 0) \quad (5.27)$$

Für den Fall, dass das Restbearbeitungszeitintervall I_a^{Rest} gleich $[0, 0]$ oder τ_a^{Rest} gleich 0 ist, entspricht die Bearbeitungszeit dieses Auftrags nicht dem aufgrund der Klassierung zu erwartenden Bearbeitungszeitintervall bzw. der zu erwartenden Bearbeitungszeit. Dieser Fall wird als Ereignis gewertet und in Abschnitt 5.4 behandelt.

5.3.2.2 Zielkriterium „Stabilität“

Zu einem beliebigen Planungszeitpunkt t_i können lediglich für Aufträge aus der Menge $\mathcal{A}_{t_i}^P$, wie im vorherigen Abschnitt beschrieben, die getroffenen Entscheidungen aus einem früheren Ablaufplan revidiert werden. Dabei sollte allerdings berücksichtigt werden, dass auf Basis des früheren Ablaufplans bereits Sekundärplanungen erfolgt sein können, die bspw. auf den berechneten Endterminen der Aufträge beruhen.⁶³ Eine häufige Änderung dieser Termine bzw. eine hohe Nervosität des erstellten Ablaufplans kann somit zu weiteren Kosten für diese Sekundärplanungen führen. Daher soll im Folgenden das in Abschnitt 5.3.1.2 beschriebene Zielsystem um eine weitere Zielfunktion erweitert werden, die die Stabilität eines Ablaufplanes bewertet und damit berücksichtigt.

⁶³Vgl. Abschnitt 2.2.2.2.

Für Aufträge aus der Menge $\mathcal{A}_{t_i}^P$ sind lediglich Ablaufreihenfolgen auf den Maschinen sowie Start- und Endtermine in Abhängigkeit des eintretenden Szenarios festgelegt. Es wurden jedoch noch keine Vorbereitungen einer Maschine in Form tatsächlicher Frachtbewegungen für diesen Auftrag vorgenommen. Daher sollte das Stabilitätskriterium lediglich die Änderung in den Start- und Endterminen dieser Aufträge von dem vorherigen Ablaufplan ap_{i-1} zum jetzigen ap_i für ein Szenario sz bewerten. Eine Änderung der Maschinenzuordnung, sofern die neue Maschine diesen Auftrag bearbeiten kann, wird, wenn dies zu keiner Änderung der Start- und Endtermine führt, durch das Stabilitätskriterium nicht betrachtet.⁶⁴

Als Stabilitätskriterium wird daher die von Jensen⁶⁵ beschriebene Stabilitätsmetrik verwendet. Bezeichne hierfür $\text{overlap}(ap_{i-1}, ap_i, a, sz)$ die Überlappung der Bearbeitungszeit des Auftrags $a \in \mathcal{A}_{t_i}^P$ in Szenario $sz \in \mathcal{SZ}$, der bereits in Ablaufplan ap_{i-1} eingeplant wurde und in dem aktuellen Ablaufplan ap_i erneut eingeplant wird:

$$\text{overlap}(ap_{i-1}, ap_i, a, sz) = \max\left(\min(C_a^{sz}(ap_{i-1}), C_a^{sz}(ap_i)) - \max(C_a^{sz}(ap_{i-1}) - \tau_a^{sz}, C_a^{sz}(ap_i) - \tau_a^{sz}), 0\right)$$

Unter Verwendung dieser Funktion kann anschließend die Überlappung zweier Ablaufpläne ap_{i-1} und ap_i für alle Aufträge $a \in \mathcal{A}_{t_i}^P$ für Szenario $sz \in \mathcal{SZ}$ mit Hilfe der Funktion $\text{overlapS}(ap_{i-1}, ap_i, sz)$ berechnet werden:

$$\text{overlapS}(ap_{i-1}, ap_i, sz) = \frac{\sum_{a \in \mathcal{A}_{t_i}^P} \text{overlap}(ap_{i-1}, ap_i, a, sz)}{\sum_{a \in \mathcal{A}_{t_i}^P} \tau_a^{sz}}$$

Die erwartete Stabilität für einen gegebenen Ablaufplan ap_i unter Berücksichtigung des bereits gestarteten Ablaufplans ap_{i-1} berechnet sich somit folgendermaßen:

$$s^{\text{sched}}(ap_{i-1}, ap_i) = \frac{\sum_{sz \in \mathcal{SZ}} \text{overlapS}(ap_{i-1}, ap_i, sz)}{|\mathcal{SZ}|} \quad (5.28)$$

Die dritte Zielfunktion ergibt sich somit, indem das Maximum für Gleichung (5.28) für alle $ap \in \mathcal{AP}_i$ ⁶⁶ und den bereits bestehenden Vorplan ap_{i-1} ermittelt wird:

$$\max_{ap \in \mathcal{AP}_i} s^{\text{sched}}(ap_{i-1}, ap) \quad (5.29)$$

Eine Visualisierung dieser Berechnung findet sich in Abbildung 5.5. Zum Zeitpunkt $t_0 = 0$ wird der initiale Ablaufplan ap_0 für ein beliebiges, aber festes Szenario $sz \in \mathcal{SZ}$ erstellt (siehe Abbildung 5.5(a)). Hierbei werden acht Aufträge auf zwei Maschinen verteilt. Zum Zeitpunkt $t_1 = 45$ wird ein weiterer Ablaufplan ap_1 erstellt (siehe Abbildung 5.5(b)),

⁶⁴Die Belastungsänderungen, die durch einen Wechsel von Maschinen auf die Transportsysteme ausgeübt werden, werden in Abschnitt 5.3.5 betrachtet und bei der Erstellung der Ablaufpläne berücksichtigt.

⁶⁵Siehe [Jen01], S. 79 f.

⁶⁶Mit \mathcal{AP}_i wird die Menge aller zulässigen Ablaufpläne zum Zeitpunkt t_i bezeichnet.

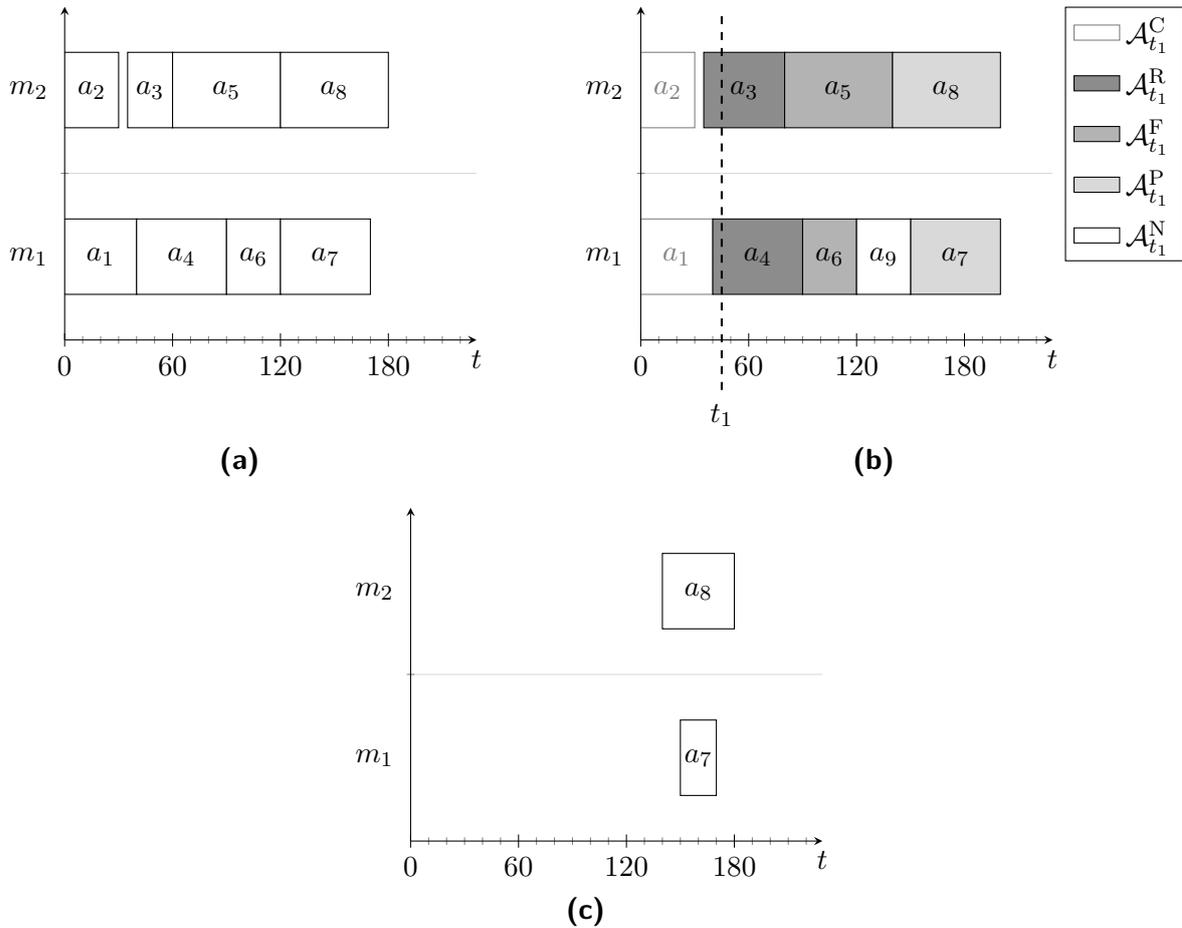


Abbildung 5.5: Visualisierung der Berechnung für die Zielfunktion „Stabilität“ unter Berücksichtigung eines beliebigen, aber festen Szenarios $sz \in \mathcal{SZ}$ (in Anlehnung an: [Jen01], S. 80): (a) Zum Zeitpunkt t_0 initial erstellter Ablaufplan ap_0 (b) Zum Zeitpunkt t_1 erstellter Ablaufplan ap_1 (c) Überlappung der Aufträge aus $\mathcal{A}_{t_1}^P$

der den neu hinzugekommenen Auftrag a_9 berücksichtigt und vor Auftrag a_7 einplant. Die Aufträge a_5 und a_8 wurden zeitlich nach hinten verschoben, da a_3 bereits eine längere Bearbeitungszeit besitzt als für das Szenario $sz \in \mathcal{SZ}$ initial angenommen. Eine Visualisierung der Funktion $\text{overlap}(ap_0, ap_1, a, sz)$ für die betroffenen Aufträge $a \in \mathcal{A}_{t_1}^P = \{a_7, a_8\}$ ist in Abbildung 5.5(c) zu finden. Das Stabilitätskriterium wird für die Menge $\mathcal{A}_{t_1}^P$ aus dem Quotienten der 20 bzw. 40 Zeiteinheiten umfassenden Überlappung für a_7 bzw. a_8 und deren Bearbeitungszeiten berechnet:

$$\text{overlapS}(ap_0, ap_1, sz) = \frac{20 + 40}{50 + 60} = \frac{6}{11}$$

Zwischen dem Stabilitätskriterium und den anderen beiden in Abschnitt 5.3.1.2 beschriebenen Zielkriterien besteht ein Trade-off. Auf der einen Seite geht eine hohe

Stabilität mit einer strikteren Vorgabe für die Ablaufplanung einher, da für bereits eingeplante Aufträge weniger Freiheiten bestehen, diese umzuplanen. Hierdurch kann sowohl die Zielfunktion (5.19) als auch (5.24) negativ beeinflusst werden. Auf der anderen Seite ermöglicht eine geringere Stabilität mehr Freiheiten bei der Umplanung von bereits eingeplanten Aufträgen und führt somit zu evtl. besseren Zielfunktionswerten für die beiden anderen Zielfunktionen. Eine geringere Stabilität kann jedoch zu den zuvor beschriebenen Problemen mit der Plannervosität führen.

5.3.2.3 Erweiterung der Modellformulierung

Basierend auf dem im vorherigen Abschnitt beschriebenen Stabilitätskriterium kann das Mehrzieloptimierungsmodell (5.30) für eine dynamische Umgebung formalisiert werden. Das Modell stellt hierbei eine Erweiterung des Optimierungsmodell (5.25) dar. Aus diesem werden die beiden Zielfunktionen, die dargestellten Restriktionen sowie die ermittelte Zuordnungslösung $x'_{a,a',m}$ übernommen.

Die Zuordnungslösung $x'_{a,a',m}$ aus Ablaufplan ap_{i-1} , der zum Zeitpunkt t_{i-1} erstellt worden ist, muss für die Verwendung in Modell (5.30) aufbereitet werden. Hierzu sind in einem ersten Schritt zunächst die Zuordnungslösungen für alle bereits beendeten Aufträge sowie alle Nachfolgerbeziehungen des Pseudoauftrags zu entfernen:

$$\begin{aligned} x'_{a,a',m} &= 0 \quad , \forall a \in \mathcal{A}_{t_i}^C; \forall a' \in \mathcal{A}_{t_{i-1}}; \forall m \in \mathcal{M} \\ x'_{a,a',m} &= 0 \quad , \forall a' \in \mathcal{A}_{t_i}^C; \forall a \in \mathcal{A}_{t_{i-1}}; \forall m \in \mathcal{M} \\ x'_{\rho,a',m} &= 0 \quad , \forall a' \in \mathcal{A}_{t_{i-1}}; \forall m \in \mathcal{M} \end{aligned}$$

Anschließend sind für den Pseudoauftrag die neuen Nachfolger festzulegen. Nachfolger sind die Aufträge, die aufgrund des vorherigen Schrittes keine Vorgängerbeziehung mehr besitzen und die ein Element der Menge $(\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ sind. Aufgrund von Restriktion (5.25i) kann dies für jede Maschine auf höchstens einen Auftrag zutreffen. Dieser Schritt ist notwendig, damit in Modell (5.30) kein Auftrag der Menge $\mathcal{A}_{t_i}^{\text{Pb}}$ zwischen Pseudoauftrag und einem laufendem oder fixiertem Auftrag eingeplant werden kann.

$$x'_{\rho,a',m} = 1 \quad , \forall a' \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C), \forall m \in \mathcal{M}: \sum_{a \in \mathcal{A}: a \neq a'} x_{a,a',m} - \sum_{a \in \mathcal{A}: a \neq a'} x_{a',a,m} = -1$$

Schließlich sind die Vorgängerbeziehungen des Pseudoauftrags zu lösen, da die Aufträge der Menge $\mathcal{A}_{t_i}^{\text{Pb}}$ eingeplant werden sollen und diese somit mögliche Vorgänger des Pseudoauftrags sind:

$$x'_{a,\rho,m} = 0 \quad , \forall a \in \mathcal{A}_{t_{i-1}}; \forall m \in \mathcal{M}$$

Mit dieser modifizierten Zuordnungslösung kann das nachfolgende Mehrzieloptimierungsmodell (5.30) für eine dynamische Umgebung formalisiert werden. Das Ergebnis des Modells bilden die Ablaufpläne (repräsentiert über die Entscheidungsvariable $x_{a,a',m}$), die hinsichtlich der drei Zielfunktionen⁶⁷ pareto-optimal sind.

⁶⁷Siehe Zielfunktionen (5.19), (5.24) sowie (5.29).

Mengen:

\mathcal{A}	Menge der zum i -ten Planungszeitpunkt t_i zu berücksichtigenden Aufträge a , die eine Auswirkung auf die aktuelle Planung haben. Die Menge ergibt sich folgendermaßen: $\mathcal{A} = \mathcal{A}_{t_i}^N \cup (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$
$\mathcal{A}_{t_{i-1}}$	Menge der zum Zeitpunkt t_{i-1} bekannten Aufträge a
$\mathcal{A}_{t_i}^C$	Menge der zum Zeitpunkt t_i bekannten Aufträge a , die bereits fertiggestellt wurden
$\mathcal{A}_{t_i}^F$	Menge der zum Zeitpunkt t_i bekannten Aufträge a , für die an den Arbeitsstationen bereits eine Vorbereitung stattgefunden hat
$\mathcal{A}_{t_i}^P$	Menge der zum Zeitpunkt t_i bekannten Aufträge a , die lediglich auf eine Arbeitsstation eingeplant sind
$\mathcal{A}_{t_i}^R$	Menge der zum Zeitpunkt t_i bekannten Aufträge a , die bereits auf den Arbeitsstationen gestartet wurden
\mathcal{M}	Menge der zur Verfügung stehenden Maschinen m
\mathcal{SZ}	Menge aller Szenarien sz , die eine mögliche Realisierung der Bearbeitungszeiten der Aufträge angeben

Parameter:

M	Big M
ρ	Pseudoauftrag, dessen Bearbeitungszeit, Bereitstellungszeit sowie Gewicht 0 ist. Es gilt: $\rho \in \mathcal{A}$
r_a	Bereitstellungszeit von Auftrag $a \in \mathcal{A}$
s_a	Tatsächlicher Startzeitpunkt des Auftrags $a \in (\mathcal{A}_{t_i}^C \cup \mathcal{A}_{t_i}^R)$. Für $a \in (\mathcal{A} \setminus \mathcal{A}_{t_i}^R)$ gilt: $s_a = 0$
τ_a^{sz}	Bearbeitungszeit des Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
$x'_{a,a',m}$	Die Zuordnungen aus Ablaufplan ap_{i-1} , der zum Zeitpunkt t_{i-1} erstellt worden ist. Binär: 1, wenn Auftrag $a \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ direkt vor Auftrag $a' \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ auf Maschine $m \in \mathcal{M}$ zugewiesen worden ist; 0 in allen anderen Fällen

Entscheidungsvariablen:

C_a^{sz}	Fertigstellungszeitpunkt des Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
C'^{sz}_a	Fertigstellungszeitpunkt des Auftrags $a \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ in Szenario $sz \in \mathcal{SZ}$, wenn die Zuordnungslösung $x'_{a,a',m}$ verwendet wird
o_a^{sz}	Überlappung der Bearbeitungszeit des Auftrags $a \in \mathcal{A}_{t_i}^P$ in Szenario $sz \in \mathcal{SZ}$, der bereits in Ablaufplan ap_{i-1} eingeplant wurde und in dem aktuellen Ablaufplan ap_i erneut eingeplant wird
p	Erwartete Stabilität für alle Szenarien
so^{sz}	Überlappung der Ablaufpläne ap_{i-1} und ap_i für alle Aufträge $a \in \mathcal{A}_{t_i}^P$ für Szenario $sz \in \mathcal{SZ}$
$x_{a,a',m}$	Binär: 1, wenn Auftrag $a \in \mathcal{A}$ direkt vor Auftrag $a' \in \mathcal{A}$ auf Maschine $m \in \mathcal{M}$ ausgeführt wird; 0 in allen anderen Fällen

Zielfunktionen:

$$(5.25a) \text{ und } (5.25b) \\ \max s^{\text{sched}} = p \quad (5.30a)$$

u. B. d. N.:

$$(5.25c) - (5.25s) \\ \frac{\sum_{sz \in \mathcal{SZ}} so^{sz}}{|\mathcal{SZ}|} = p \quad (5.30b)$$

$$\frac{\sum_{a \in \mathcal{A}_{t_i}^P} o_a^{sz}}{\sum_{a \in \mathcal{A}_{t_i}^P} \tau_a^{sz}} = so^{sz}, \quad \forall sz \in \mathcal{SZ} \quad (5.30c)$$

$$\max(\min(C_a^{sz}, C_a'^{sz}) - \max(C_a^{sz} - \tau_a^{sz}, C_a'^{sz} - \tau_a^{sz}), 0) = o_a^{sz}, \quad \forall a \in \mathcal{A}_{t_i}^P; \forall sz \in \mathcal{SZ} \quad (5.30d)$$

$$x_{a,a',m} = x'_{a,a',m}, \quad \forall a \in (\mathcal{A}_{t_i}^R \cup \mathcal{A}_{t_i}^F); \\ \forall a' \in (\mathcal{A}_{t_i}^R \cup \mathcal{A}_{t_i}^F); \\ \forall m \in \mathcal{M} \quad (5.30e)$$

$$C_a'^{sz} - C_a'^{sz} + (M - \max(r_{a'}, s_{a'}))x'_{a,a',m} \leq M - (\max(r_{a'}, s_{a'}) + \tau_{a'}^{sz}), \quad \forall a \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C); \\ \forall a' \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C); \\ a' \neq \rho, a \neq a'; \\ \forall m \in \mathcal{M}; \forall sz \in \mathcal{SZ} \quad (5.30f)$$

$$C_{a_i}^{sz} + \tau_{a_j}^{sz} \leq C_{a_j}^{sz}, \quad \forall (a_i, a_j) \in \mathcal{E}; \\ \forall sz \in \mathcal{SZ} \quad (5.30g)$$

$$so^{sz} \in [0, 1], \quad \forall sz \in \mathcal{SZ} \quad (5.30h)$$

$$o_a^{sz} \geq 0, \quad \forall a \in \mathcal{A}_{t_i}^P; \forall sz \in \mathcal{SZ} \quad (5.30i)$$

$$p \geq 0 \quad (5.30j)$$

Die Zielfunktion (5.30a) in Kombination mit Restriktion (5.30b) maximiert die erwartete Stabilität des zu erstellenden Ablaufplans. Mit Hilfe der Gleichung (5.30d) wird die überlappende Bearbeitungszeit des Auftrags a in Szenario sz berechnet. Diese Berechnung wird für alle Aufträge durchgeführt, die bereits zu einem früheren Zeitpunkt eingeplant worden sind, für die aber bisher keine Vorbereitungen an der Arbeitsstation stattgefunden haben und die somit zu der Menge $\mathcal{A}_{t_i}^P$ gehören. Mit Gleichung (5.30c) wird das Stabilitätskriterium für den gesamten Ablaufplan für jedes Szenario sz berechnet. Über die Restriktion (5.30e) wird sichergestellt, dass Aufträge aus den beiden Mengen $\mathcal{A}_{t_i}^R$ und $\mathcal{A}_{t_i}^F$ nicht umgeplant werden können. Hierfür werden die Zuordnungen aus der modifizierten vorherigen Zuordnungslösung $x'_{a,a',m}$ in der aktuellen $x_{a,a',m}$ fixiert. Restriktion (5.30f) berechnet den Fertigstellungszeitpunkt $C_a'^{sz}$ aller Aufträge aus $(\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ des vorherigen Planungszeitraums unter Berücksichtigung der aktuellen Szenariomenge sowie der modifizierten Zuordnungslösung aus dem zuvor erstellten Ablaufplan ap_{i-1} . Des Weiteren

werden über Restriktion (5.30g) evtl. bestehende Vorrangbeziehungen zwischen diesen Aufträgen beachtet. Nebenbedingung (5.30h) stellt sicher, dass die Stabilität eines Ablaufplans für ein Szenario sz einen Wert im Intervall $[0, 1]$ besitzt. Die Nichtnegativität der Entscheidungsvariablen wird über die Restriktionen (5.30i) und (5.30j) sichergestellt.

5.3.3 Evolutionäre Heuristik für die Ablaufplanung

Das im vorherigen Abschnitt aufgestellte Modell (5.30) kann wie auch schon das Modell (5.25) nicht direkt durch einen Solver gelöst werden. Zum einen ermöglicht das verwendete Zielsystem aufgrund der beschriebenen konfliktären Zielfunktionen keine eindeutige Identifizierung einer Lösung.⁶⁸ Zum anderen bestehen, wie bereits angesprochen, aufgrund der unendlichen Anzahl von Szenarien der Szenariomenge ($|\mathcal{SZ}| = \infty$) unendlich viele Restriktionen in dem Modell. Des Weiteren stellt das in dem Modell betrachtete Ablaufplanungsproblem bereits für den deterministischen und statischen Fall ein NP-schweres kombinatorisches Optimierungsproblem dar, für das bei größeren Probleminstanzen in der zur Verfügung stehenden Zeit mit Hilfe des Modells keine Lösung ermittelt werden kann.⁶⁹

Aus den zuvor genannten Gründen wird in diesem Abschnitt eine evolutionäre Metaheuristik (NSGA-II) verwendet, um die Ablaufpläne auf Basis einer rollierenden Planung zu erstellen. Der Ablauf dieser Heuristik, dargestellt in Abbildung 5.6, orientiert sich am Standardablauf⁷⁰ für genetische Algorithmen, der um problemspezifische Anpassungen ergänzt wird und aus den folgenden Schritten besteht:

1. Initialisierung: Erstellung einer endlichen Szenariomenge \mathcal{SZ} und Generierung einer Initialpopulation \mathcal{P}_0 möglicher Ablaufpläne
2. Evaluation: Jeder Ablaufplan aus \mathcal{P}_0 wird auf Basis des Zielsystems bewertet
3. Setze $i = 0$
4. Selektion: Auswahl der Individuen aus \mathcal{P}_i , die in die Reproduktion eingehen
5. Reproduktion:
 - a) Rekombination: Aus der Kombination von Merkmalen der zuvor selektierten Ablaufpläne werden neue Ablaufpläne erzeugt, die die Nachfolpopulation \mathcal{Q}_i bilden
 - b) Mutation: Die Ablaufpläne aus \mathcal{Q}_i werden zufälligen Änderungen unterworfen und mit dem Verfahren aus Schritt 2 evaluiert

⁶⁸Eine Zusammenfassung der Zielfunktionen durch Gewichte und somit die Beschränkung auf eine eindeutige Lösung soll hier vermieden werden, da die Informationen über die möglichen pareto-optimalen Pläne eine wertvolle Entscheidungshilfe für den Entscheidungsträger darstellen.

⁶⁹Vgl. [KC93], S. 160 f., mit Verweis auf [GJ79], S. 238, für den deterministischen Fall und [AD08] für den Ein-Maschinen-Fall unter szenariobasierter Unsicherheit der Bearbeitungszeiten.

⁷⁰Vgl. [Tal09], S. 205.

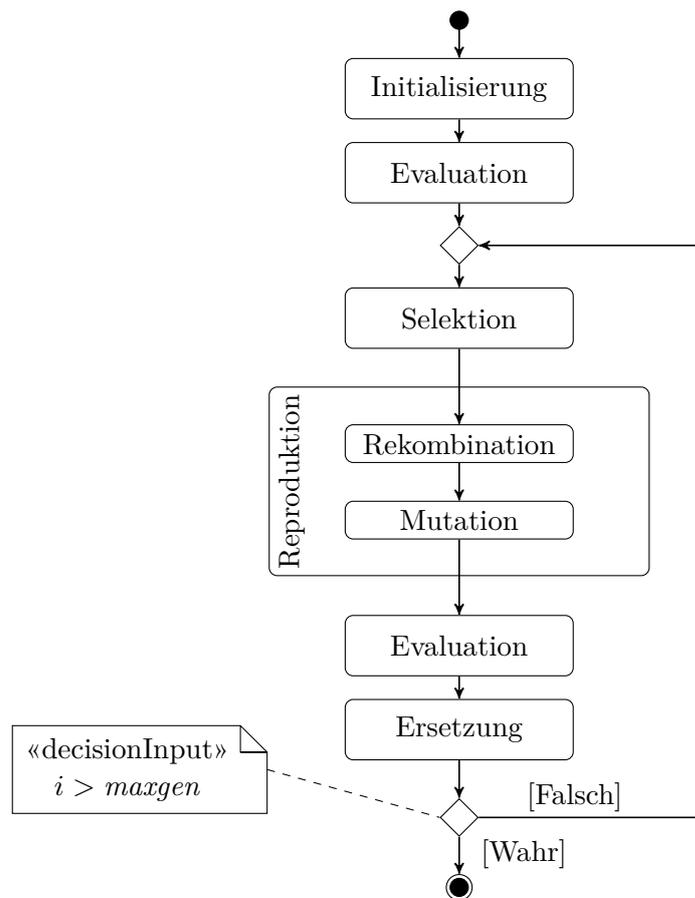


Abbildung 5.6: Ablauf des genetischen Algorithmus (in Anlehnung an: [Bal11], S. 6816)

6. Ersetzung: Aus $\mathcal{R}_i = \mathcal{P}_i \cup \mathcal{Q}_i$ werden die Ablaufpläne bestimmt, die in die Population \mathcal{P}_{i+1} für die nächste Generation übernommen werden.
7. Terminierung: Setze $i = i + 1$. Ein Abbruchkriterium entscheidet darüber, ob die Heuristik beendet wird. Ist dies nicht der Fall, wird Schritt 4 erneut ausgeführt.
8. Ausgabe der Ergebnisse in Form der nicht-dominierten Ablaufpläne

Bevor diese Schritte in den Abschnitten 5.3.3.2–5.3.3.7 näher ausgeführt werden, wird im folgenden Abschnitt die Kodierung und Dekodierung der Ablaufpläne beschrieben, die als grundlegende Repräsentation eines Ablaufplans in allen weiteren Schritten der Heuristik verwendet wird.

5.3.3.1 Kodierung und Dekodierung der Ablaufpläne

Für die Anwendung eines genetischen Algorithmus ist es notwendig, Lösungen (in diesem Fall also die Ablaufpläne) in Chromosomen zu kodieren. Ein Chromosom entspricht hierbei einem Ablaufplan und die einzelnen Gene einer Zuordnung eines Auftrags zu einer Maschine. Die Gesamtheit der Chromosomen ergibt den Kodierungsraum, auf

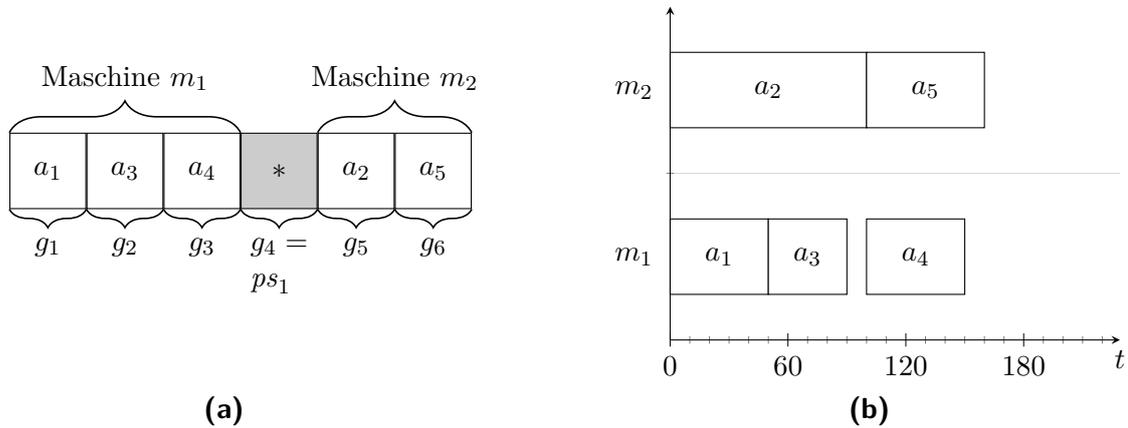


Abbildung 5.7: Dekodierung eines Individuums zu einem Ablaufplan: (a) Kodierter Ablaufplan (Genotyp) (b) Dekodierter Ablaufplan (Phänotyp)

dem die genetischen Operatoren ausgeführt werden. Der Kodierungsraum wird durch die Dekodierung in den Lösungsraum überführt, auf dem die Evaluation stattfindet. Bei der Kodierung sind die in Abschnitt 3.2.2.1 beschriebenen Anforderungen nach der Zulässigkeit, der Gültigkeit sowie der Eindeutigkeit der Zuordnung zwischen einem Chromosom (Genotyp) und der dekodierten Lösung (Phänotyp) zu beachten. Des Weiteren müssen alle zulässigen Ablaufpläne durch die Kodierung darstellbar sein.

Im Folgenden wird ein Kodierungsansatz nach Hou et al.⁷¹ verwendet, der für jede Maschine eine Liste von auszuführenden Aufträgen vorsieht. Die Reihenfolge innerhalb der Listen entspricht dabei der Ausführungsreihenfolge auf der entsprechenden Maschine.⁷² Dieser Ansatz wird in Kombination mit Partitionierungssymbolen nach Cheng et al.⁷³ verwendet, um eine Zusammenfassung der einzelnen Listen zu einer Kodierungskette zu erreichen.

Bezeichne $\vec{g}_\eta = (g_1, g_2, \dots, g_{n+l-1})$ den Vektor der Gene bzw. das Chromosom des Individuums η , wobei $n = |\mathcal{A}_{t_i}^{\text{Pb}}|$ der Anzahl aller planbaren Aufträge und $l = |\mathcal{M}|$ der Anzahl der vorhandenen Maschinen entspricht. Jedes Gen g_j mit $1 \leq j \leq (n+l-1)$ entspricht entweder einem Auftrag $g_j \in \mathcal{A}_{t_i}^{\text{Pb}}$ oder einem von $l-1$ Partitionierungssymbolen $g_j \in \{*\}$. Hierbei bezeichne $\text{pos}(ps_i)$ die Position des i -ten Partitionierungssymbols, wobei $\text{pos}(ps_0) = 0$ und $\text{pos}(ps_l) = (n+l)$ gelte. Die Partitionierungssymbole unterteilen die Aufträge und weisen sie somit den Maschinen zu, die die Aufträge ausführen können. Ein Auftrag (kodiert als g_j) ist dabei der Maschine m_i mit $1 \leq i \leq l$ zugewiesen, wenn für ihn $\text{pos}(ps_{i-1}) < j < \text{pos}(ps_i)$ gilt. Abbildung 5.7(a) stellt ein kodiertes Individuum grafisch dar.

Zur Dekodierung eines Individuums η werden die Gene von Position $j = 1$ beginnend nach rechts bis zum letzten Gen an Position $j = (n+l-1)$ durchlaufen. Für jedes Gen g_j wird der durch dieses Gen kodierte Auftrag der nach obiger Ungleichung

⁷¹[HAR94].

⁷²Siehe ebd., S. 115.

⁷³[CGT95].

berechneten Maschine zugeordnet. Bei der Zuordnung sind der Vorranggraph \mathcal{G} , mögliche Bereitstellungszeiten r_a der Aufträge, das gewählte Szenario sz der Bearbeitungszeiten (vgl. Abschnitt 5.3.3.3) sowie der bereits in der Ausführung befindliche Ablaufplan zu berücksichtigen. Abbildung 5.7 visualisiert die Dekodierung eines Individuums zu einem Ablaufplan. Das Individuum besteht aus fünf Aufträgen und zwei Maschinen, die in kodierter Form vorliegen (siehe Abbildung 5.7(a)). Des Weiteren seien der Vorranggraph $\mathcal{G} = (\mathcal{A}, \{(a_2, a_4), (a_2, a_5)\})$ und das Szenario $sz = \{50, 100, 40, 50, 60\}$ gegeben. Es wird angenommen, dass kein vorheriger Ablaufplan zu berücksichtigen ist und $r_a = 0$ für alle Aufträge gilt. Der entsprechende dekodierte Ablaufplan ist in Abbildung 5.7(b) aufgeführt.

5.3.3.2 Initialisierung

Die Zielfunktionswerte (5.18), (5.23) und (5.28) können aufgrund der unendlichen Szenariomenge nicht direkt evaluiert werden. Aus diesem Grund soll auf Basis der für jeden Auftrag ermittelten Clusterzugehörigkeit (vgl. Abschnitt 5.3.1.1) und der für ein Cluster hinterlegten Häufigkeitsverteilung eine endliche Menge von Szenarien erzeugt werden. Das Vorgehen ist dabei folgendes:

1. Für jeden Auftrag $a \in \mathcal{A}$ wird eine Bearbeitungszeit τ_a unter Verwendung von Gleichung (5.16) bzw. für bereits gestartete Aufträge unter Verwendung von Gleichung (5.27) erstellt.
2. Es wird ein Szenario $sz_i = \{\tau_{a_1}^{sz_i}, \tau_{a_2}^{sz_i}, \dots, \tau_{a_n}^{sz_i}\}$ erzeugt und für dieses ein *optimaler*⁷⁴ Ablaufplan $ap_{sz_i}^*$ ermittelt (vgl. Gleichung (5.20)). Anschließend wird das Szenario der Szenariomenge \mathcal{SZ} hinzugefügt.
3. Die Schritte 1 und 2 werden solange wiederholt, bis eine angegebene Anzahl⁷⁵ von Szenarien erzeugt worden ist.

Neben den so erzeugten Szenarien werden vier Spezialszenarien generiert und ebenfalls der Szenariomenge \mathcal{SZ} hinzugefügt. Als erstes wird das Szenario erzeugt, in dem jeder Auftrag $a \in \mathcal{A}$ seine längste Bearbeitungszeit bzw. im Falle der bereits gestarteten Aufträge seine längste Restbearbeitungszeit (siehe Gleichung (5.26)) benötigt:

$$sz^{ub} = \{\tau_a \mid (\tau_a = \max(I_a) \wedge a \notin \mathcal{A}_t^R) \vee (\tau_a = \max(I_a^{\text{Rest}}) \wedge a \in \mathcal{A}_t^R)\} \quad (5.31)$$

⁷⁴Dies erfolgt hier aufgrund der Komplexität des Problems ebenfalls mit einem genetischen Algorithmus. Dieser optimiert jedoch ausschließlich die erste Zielfunktion (vgl. Gleichung (5.19)). Die verwendete Kodierung und die genetischen Operatoren sind dabei die gleichen wie für den hier beschriebenen NSGA-II-Algorithmus. Aufgrund der Verwendung einer Heuristik müsste der Begriff *optimal* präziser mit *dem besten durch den GA ermittelten Ablaufplan* beschrieben werden, da der gefundene Plan nicht notwendigerweise dem nachweisbar optimalen Ergebnis entspricht. Aus pragmatischen Gründen wird hier und in den folgenden Ausführungen im Zusammenhang mit der evolutionären Heuristik jedoch der Terminus „optimal“ weiter verwendet.

⁷⁵Diese Anzahl richtet sich nach den zur Verfügung stehen Ressourcen zur Durchführung der Evaluation und kann in Testläufen ermittelt werden.

Für dieses Szenario wird anschließend der szenariooptimale Ablaufplan $ap_{sz^{ub}}^*$ ermittelt und sz^{ub} der Szenariomenge \mathcal{SZ} hinzugefügt.

Ein weiteres Szenario wird für den Fall erzeugt, dass jeder Auftrag $a \in \mathcal{A}$ seine kürzeste Bearbeitungszeit bzw. im Falle der bereits gestarteten Aufträge seine kürzeste Restbearbeitungszeit (siehe Gleichung (5.26)) benötigt:

$$sz^{lb} = \{\tau_a \mid (\tau_a = \min(I_a) \wedge a \notin \mathcal{A}_{t_i}^R) \vee (\tau_a = \min(I_a^{\text{Rest}}) \wedge a \in \mathcal{A}_{t_i}^R)\}$$

Für dieses Szenario wird der Ablaufplan $ap_{sz^{lb}}^*$ erstellt und sz^{lb} der Szenariomenge \mathcal{SZ} hinzugefügt.

Für das dritte Spezialszenario wird der jeweilige Erwartungswert der Zufallsvariable $X \sim \mathcal{H}_{C^*}$ ermittelt und als Bearbeitungszeitabweichung $\Delta\tau_{C^*}$ in Gleichung (5.16) bzw. für bereits gestartete Aufträge in Gleichung (5.27) verwendet. Das Szenario wird anschließend folgendermaßen berechnet:

$$sz^{\text{ex}} = \{\tau_a \mid (\tau_a = \tau_a \wedge a \notin \mathcal{A}_{t_i}^R) \vee (\tau_a = \tau_a^{\text{Rest}} \wedge a \in \mathcal{A}_{t_i}^R)\}$$

Es wird erneut der szenariooptimale Ablaufplan $ap_{sz^{\text{ex}}}^*$ generiert und sz^{ex} der Szenariomenge \mathcal{SZ} hinzugefügt.

Schließlich wird als letztes Spezialszenario für die Bearbeitungszeiten der Aufträge die über eine Formel⁷⁶ abgeschätzte Bearbeitungszeit verwendet:

$$sz^{\text{co}} = \{\tau_a \mid (\tau_a = \hat{\tau}_a \wedge a \notin \mathcal{A}_{t_i}^R) \vee (\tau_a = \hat{\tau}_a - (t_i - s_a) \wedge a \in \mathcal{A}_{t_i}^R)\}$$

Der entsprechende szenariooptimale Ablaufplan wird mit $ap_{sz^{\text{co}}}^*$ bezeichnet und sz^{co} ebenfalls der Szenariomenge \mathcal{SZ} hinzugefügt. Die so erstellte Szenariomenge \mathcal{SZ} wird für die Evaluation aller Individuen in allen Generationen⁷⁷ verwendet.

Auf Basis der im vorherigen Abschnitt beschriebenen Kodierung von Ablaufplänen kann im Folgenden eine Population \mathcal{P}_0 , d. h. eine Liste⁷⁸ von Ablaufplänen, erstellt werden, die als Initialpopulation in dem genetischen Algorithmus verwendet wird. Bei der Erstellung eines Individuums ist darauf zu achten, dass der sich hieraus ergebene Ablaufplan zulässig und gültig ist, d. h., jeder Auftrag der Menge $\mathcal{A}_{t_i}^{\text{Pb}}$ muss in dem Individuum kodiert werden und insbesondere die Vorrangbeziehungen zwischen den Aufträgen müssen eingehalten werden.

Bezeichne hierfür $\mathcal{A}_a^{\text{prec}} = \{b \mid b \in \mathcal{A}_{t_i}^{\text{Pb}} \wedge a \neq b \wedge (b, a) \in \mathcal{E}\}$ die Menge der direkten Vorgänger für Auftrag a , d. h. Aufträge, die aufgrund der in dem Vorranggraphen \mathcal{G} angegebenen Vorrangbeziehungen \mathcal{E} direkt vor Auftrag a ausgeführt werden müssen. Sei weiterhin $\mathcal{A}_{t_i}^{\text{open}}$ die Menge der noch nicht kodierten Aufträge, die zu Beginn der Menge $\mathcal{A}_{t_i}^{\text{Pb}}$ entspricht, und sei $\mathcal{A}_{t_i}^{\text{close}}$ die Menge der bereits kodierten Aufträge, die initial

⁷⁶Vgl. beispielsweise Gleichung (2.1) auf Seite 18.

⁷⁷Dies soll zum einen sicherstellen, dass alle Individuen auf Basis der gleichen Szenarien evaluiert werden, und zum anderen beschränkt es den nötigen Aufwand für die Evaluation, da für jedes Szenario sz_i ein optimaler Ablaufplan $ap_{sz_i}^*$ ermittelt werden muss.

⁷⁸Es wird eine Populationsliste statt -menge verwendet, da Duplikate eines Individuums erlaubt sein sollen. Die Notation für Mengen wird jedoch beibehalten.

leer ist. Für jede Maschine m wird zusätzlich eine initial leere Liste \mathcal{L}_m angelegt. Ein Individuum η wird anschließend durch die folgenden Schritte erzeugt:

1. Setze $\mathcal{A}_{t_i}^{\text{open}} = \mathcal{A}_{t_i}^{\text{Pb}}$ und $\mathcal{A}_{t_i}^{\text{close}} = \emptyset$
2. Wähle zufällig einen noch nicht kodierten Auftrag, dessen gesamte Vorgänger bereits kodiert wurden: $a \in \mathcal{A}_{t_i}^{\text{open}} : \mathcal{A}_a^{\text{prec}} \subseteq \mathcal{A}_{t_i}^{\text{close}}$
3. Entferne den gerade ausgewählten Auftrag a aus der Menge der noch nicht kodierten Aufträge $\mathcal{A}_{t_i}^{\text{open}} = \mathcal{A}_{t_i}^{\text{open}} \setminus a$ und füge diesen der Menge der bereits kodierten Aufträge hinzu: $\mathcal{A}_{t_i}^{\text{close}} = \mathcal{A}_{t_i}^{\text{close}} \cup a$
4. Erzeuge das Gen g_j , indem $g_j = a$ gesetzt wird.
5. Für die Wahl der Maschine m wird für ein Individuum η alternierend eine der folgenden Strategien ausgewählt:
 - a) Wähle zufällig eine Maschine m aus der Menge \mathcal{M}_a .
 - b) Wähle unter der Annahme der Bearbeitungszeiten des Szenarios sz^{ub} die Maschine m aus der Menge \mathcal{M}_a , die am frühesten verfügbar ist.
 - c) Wähle unter der Annahme der Bearbeitungszeiten des Szenarios sz^{lb} die Maschine m aus der Menge \mathcal{M}_a , die am frühesten verfügbar ist.
 - d) Unter der Annahme der Bearbeitungszeiten des Szenarios sz^{ex} wähle die Maschine m aus der Menge \mathcal{M}_a , die am frühesten verfügbar ist.

Das Gen g_j wird dem Ende der Liste \mathcal{L}_m hinzugefügt.

6. Wiederhole die Schritte 2 bis 5, solange gilt: $\mathcal{A}_{t_i}^{\text{open}} \neq \emptyset$
7. Fasse die erstellten Listen $\mathcal{L}_{m_1}, \dots, \mathcal{L}_{m_l}$ in der Reihenfolge zu dem Chromosom \vec{g}_η des Individuums η zusammen, indem die Listen durch $(l - 1)$ Partitionierungssymbole voneinander getrennt werden.

Das zuvor beschriebene Vorgehen wird anschließend $(\text{popsize} - 8)$ -mal durchgeführt, um die initiale Population zu erstellen. Jedes erstellte Individuum η_k mit $1 \leq k \leq \text{popsize}$ wird anschließend der Liste der Initialpopulation \mathcal{P}_0 hinzugefügt. Weitere vier Individuen ergeben sich aus den vier für die Spezialszenarien ermittelten und als Individuen kodierten Ablaufplänen. Die verbleibenden vier Individuen werden durch eine Änderung des zweiten Schrittes des zuvor beschriebenen Verfahrens erstellt. Anstatt einer zufälligen Auswahl des als nächstes einzuplanenden Auftrags wird der Auftrag mit dem frühesten Fälligkeitstermin (EDF-Regel) und der kleinsten Identifikationsnummer ausgewählt, dessen Vorgänger bereits eingeplant worden sind. Diese Auswahl ist deterministisch und in Kombination mit den vier Strategien aus Schritt 5 ergeben sich vier zusätzliche Lösungen.

Das gesamte Vorgehen zur Erstellung der Initialpopulation ist in Algorithmus 5.7 aufgeführt. Ein Individuum η_k wird in den Zeilen 8–25 erstellt. Die Suche nach einem Auftrag, der als nächstes eingeplant werden soll, erfolgt in den Zeilen 11–13, indem zufällig ein Auftrag ausgewählt wird, dessen gesamte Vorgänger bereits eingeplant worden sind.

Algorithmus 5.7: Erstellung der Initialpopulation für den genetischen Algorithmus

Input : $\mathcal{A}_{t_i}^{\text{Pb}}$ – Die Menge der zum Zeitpunkt t_i einzuplanenden Aufträge
 popsize – Zu erstellende Anzahl an Individuen bzw. Ablaufplänen
Output : \mathcal{P}_0 – Die Initialpopulation von Individuen bzw. Ablaufplänen

```

1  $\mathcal{P}_0 \leftarrow \emptyset;$ 
2  $s \leftarrow 1;$ 
3 for  $k \leftarrow 1$  to  $\text{popsize} - 4$  do
4   for  $i \leftarrow 1$  to  $|\mathcal{M}|$  do
5      $\mathcal{L}_{m_i} \leftarrow \emptyset;$ 
6      $\mathcal{A}_{t_i}^{\text{open}} = \mathcal{A}_{t_i}^{\text{Pb}};$ 
7      $\mathcal{A}_{t_i}^{\text{close}} = \emptyset;$ 
8     for  $j \leftarrow 1$  to  $|\mathcal{A}_{t_i}^{\text{Pb}}|$  do
9        $a \leftarrow \text{null};$ 
10      if  $k < \text{popsize} - 8$  then
11         $a \leftarrow \text{rnd}(\mathcal{A}_{t_i}^{\text{open}});$ 
12        while  $\mathcal{A}_a^{\text{prec}} \not\subseteq \mathcal{A}_{t_i}^{\text{close}}$  do
13           $a \leftarrow \text{rnd}(\mathcal{A}_{t_i}^{\text{open}});$ 
14        else
15           $a \leftarrow \text{edf}(\mathcal{A}_{t_i}^{\text{open}});$ 
16          while  $\mathcal{A}_a^{\text{prec}} \not\subseteq \mathcal{A}_{t_i}^{\text{close}}$  do
17             $a \leftarrow \text{edf}(\mathcal{A}_{t_i}^{\text{open}});$ 
18           $\mathcal{A}_{t_i}^{\text{open}} \leftarrow \mathcal{A}_{t_i}^{\text{open}} \setminus a;$ 
19           $\mathcal{A}_{t_i}^{\text{close}} \leftarrow \mathcal{A}_{t_i}^{\text{close}} \cup a;$ 
20           $g_j \leftarrow a;$ 
21           $m \leftarrow \text{strat}_s(\mathcal{M}_a);$  // Auswahl einer Maschine nach Strategie  $s$ 
22           $\mathcal{L}_m \leftarrow \mathcal{L}_m \cup g_j;$ 
23      for  $i \leftarrow 1$  to  $|\mathcal{M}| - 1$  do
24         $\vec{g}_{\eta_k} \leftarrow \vec{g}_{\eta_k} \cup \mathcal{L}_i \cup \{*\};$ 
25       $\vec{g}_{\eta_k} \leftarrow \vec{g}_{\eta_k} \cup \mathcal{L}_{m_i};$ 
26       $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \eta_k;$ 
27       $s \leftarrow s + 1;$ 
28      if  $s > 4$  then
29         $s \leftarrow 1;$ 
30  $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup ap_{sz^{\text{ub}}}^* \cup ap_{sz^{\text{lb}}}^* \cup ap_{sz^{\text{ex}}}^* \cup ap_{sz^{\text{co}}}^*;$ 
31 return  $\mathcal{P}_0;$ 

```

Alternativ wird in den Zeilen 15–17 der Auftrag mit dem frühesten Fälligkeitstermin ausgewählt, dessen gesamte Vorgänger bereits eingeplant worden sind. Für den ausgewählten Auftrag wird anschließend in Zeile 21 eine entsprechende Maschine nach der aktuell aktiven Strategie s ausgewählt. Die Erstellung der Population der Größe $popsize - 4$ erfolgt in den Zeilen 3–29. Schließlich werden die für die Spezialszenarien erzeugten optimalen Ablaufpläne in Zeile 30 der Initialpopulation hinzugefügt.

5.3.3.3 Evaluation der Ablaufpläne auf Basis des Zielsystems

Nach der Erstellung der Initialpopulation und nach jeder weiteren Iteration sind die erstellten Individuen auf Basis des in Abschnitt 5.3.1.2 erstellten und in Abschnitt 5.3.2.2 erweiterten Zielsystems zu evaluieren. Hierzu ist in einem ersten Schritt der durch ein Individuum η kodierte Ablaufplan zu dekodieren und mit dem sich bereits in der Ausführung befindlichen Ablaufplan ap_{i-1} zu dem neuen Plan ap_i^η zu kombinieren. In einem zweiten Schritt wird Plan ap_i^η hinsichtlich der drei Zielfunktionen evaluiert und das Individuum somit bewertet. Die Durchführung der beiden Schritte wird im Folgenden näher erläutert.

Der erste Schritt wird durchgeführt, indem die in Abschnitt 5.3.3.1 beschriebene Prozedur zur Dekodierung auf ein Individuum η angewendet wird. Der Auftrag a , der in einem Gen g_j kodiert ist, darf allerdings auf der entsprechenden Maschine erst nach den dort evtl. bereits gestarteten oder fixierten Aufträgen eingeplant werden. Der so erstellte temporäre Ablaufplan ap_i^η unter Verwendung des Individuums η ist aufgrund der verwendeten Kodierung, die die Vorrangbeziehungen der Aufträge beachtet, hinsichtlich der in Modell (5.30) aufgestellten Restriktionen (bis auf Restriktion (5.25e) aus Modell (5.25)) bereits zulässig. Die Zulässigkeit der Restriktion (5.25e) kann jedoch erst im Lösungsraum bei der Evaluierung von r^{sched} erfolgen.

Die Bewertung des Ablaufplans ap_i^η stellt den zweiten Schritt dar. Hierzu ist es zunächst notwendig zu testen, ob der Plan die Restriktion (5.25e) einhält. Hierzu ist zu überprüfen, ob für alle Szenarien $sz \in \mathcal{SZ}$ und für das Worst-Case-Szenario sz_η^{worst} die Ungleichung (5.22) gilt. Das Worst-Case-Szenario entspricht für dieses Problem jedoch nicht notwendigerweise einem Extrempunktszenario.⁷⁹ Dies führt dazu, dass die Szenariomenge \mathcal{SZ} nicht auf eine endliche Anzahl zu betrachtender Szenarien reduziert werden kann, um das Worst-Case-Szenario zu finden.⁸⁰

Ein Beispiel⁸¹ ist in Abbildung 5.8 gegeben, bei dem drei Aufträge auf zwei Maschinen einzuplanen sind. Die Bearbeitungszeitintervalle der Aufträge sind dabei als $I_{a_1} = [50, 50]$, $I_{a_2} = [20, 130]$ und $I_{a_3} = [35, 35]$ mit den Fälligkeitsterminen $d_1 = 200$, $d_2 = 90$ und $d_3 = 40$ definiert. Somit ist nur die Bearbeitungszeit des Auftrags a_2 unsicher und es

⁷⁹Ein Extrempunktszenario ist ein Szenario, bei dem alle Bearbeitungszeiten τ_a der im Ablaufplan ap_i^η eingeplanten Aufträge a entweder ihre kürzeste τ_a^{\min} oder längste τ_a^{\max} Bearbeitungszeit des zugehörigen Bearbeitungszeitintervalls I_a bzw. I_a^{Rest} annehmen.

⁸⁰Vergleiche beispielsweise die Ablaufplanungsprobleme in [XCLQ13] oder [XLC14], die in Abschnitt 3.2.4 zusammengefasst wurden, in denen das Worst-Case-Szenario einem Extrempunktszenario entspricht.

⁸¹Vergleiche das Beispiel von Kasperski, der das Problem anhand eines Beispiels für das Problem $1||\sum U_i$ mit einer *minmax regret*-Zielfunktion illustriert (vgl. [Kas04], S. 287 f.; [Kas08], S. 182).

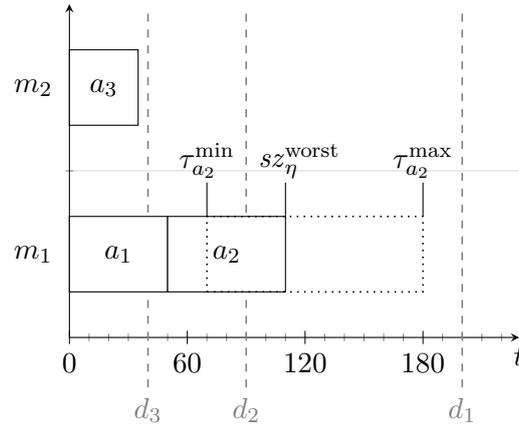


Abbildung 5.8: Visualisierung eines Worst-Case-Szenarios bei Anwendung des absoluten Regret-Kriteriums, das kein Extrempunktszenario ist (Beispiel in Anlehnung an: [Kas08], S. 182)

ergeben sich die zwei Extrempunktszenarien $sz_1 = \{\tau_{a_1}^{sz_1} = 50, \tau_{a_2}^{sz_1} = 20, \tau_{a_3}^{sz_1} = 35\}$ und $sz_2 = \{\tau_{a_1}^{sz_2} = 50, \tau_{a_2}^{sz_2} = 130, \tau_{a_3}^{sz_2} = 35\}$. Für diese beiden Szenarien und den in der Abbildung 5.8 dargestellten Ablaufplan ap gilt $R(ap, sz_1) = 0$ und $R(ap, sz_2) = 0$. Für das Szenario $sz_3 = \{\tau_{a_1}^{sz_3} = 50, \tau_{a_2}^{sz_3} = 60, \tau_{a_3}^{sz_3} = 35\}$ hingegen ist $R(ap, sz_3) = 1/3$, da $U_{sz_3}^* = 0$ gilt, wenn Auftrag a_2 vor a_1 auf Maschine m_1 eingeplant wird. Somit stellt sz_3 eines der Worst-Case-Szenarien sz_η^{worst} für den durch das Individuum η kodierten Ablaufplan ap_i^η dar.

Aus dem zuvor erläuterten Grund wird im Folgenden eine alternative Vorgehensweise beschrieben, die ein möglichst ungünstiges Szenario sz_η^{dis} für den Ablaufplan ap_i^η ermittelt. Hierzu wird für Szenario sz^{ub} (siehe Gleichung (5.31)) die Anzahl⁸² $U(ap_i^\eta, sz^{\text{ub}})$ säumiger Aufträge bei Verwendung des Individuums η berechnet. Anschließend wird unter Verwendung des Modells (5.32) das Szenario sz_η^{dis} ermittelt, das unter Beibehaltung des Ablaufplans ap_i^η zur gleichen Anzahl säumiger Aufträge wie Szenario sz^{ub} führt, dabei aber für jeden Auftrag eine möglichst geringe Bearbeitungszeit τ_a ausnutzt. Ziel dieses Vorgehens ist es, dass der für das Szenario sz_η^{dis} optimale Ablaufplan $ap_{sz_\eta^{\text{dis}}}^*$ eine wesentlich geringere Anzahl säumiger Aufträge aufweist als Ablaufplan ap_i^η unter diesem Szenario und dass somit der Regret maximiert wird.

Mengen:

\mathcal{A}	Menge der zum i -ten Planungszeitpunkt t_i zu berücksichtigenden Aufträge a , die eine Auswirkung auf die aktuelle Planung haben. Die Menge ergibt sich wie folgt: $\mathcal{A} = \mathcal{A}_{t_i}^N \cup (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$
$\mathcal{A}_{t_i}^R$	Menge der zum Zeitpunkt t_i bekannten Aufträge a , die bereits auf den Arbeitsstationen gestartet wurde
\mathcal{E}	Menge von Vorrangbeziehungen zwischen den Aufträgen
\mathcal{M}	Menge der zur Verfügung stehenden Maschinen m

⁸²Der Gewichtungsfaktor w_a aus Gleichung (5.17) wird hier nicht berücksichtigt.

Parameter:

d_a	Fälligkeitstermin von Auftrag $a \in \mathcal{A}$
M	Big M
ρ	Pseudoauftrag, dessen Bearbeitungszeit, Bereitstellungszeit sowie Gewicht 0 ist. Dieser Auftrag entspricht sowohl dem ersten als auch dem letzten Auftrag, der auf einer Maschine ausgeführt wird. Es gilt: $\rho \in \mathcal{A}$
r_a	Bereitstellungszeit von Auftrag $a \in \mathcal{A}$
$U(ap_i^\eta, sz^{\text{ub}})$	Die Anzahl säumiger Aufträge in Ablaufplan ap_i^η unter Szenario sz^{ub}
$x_{a,a',m}$	Die Zuordnungen aus Ablaufplan ap_i^η . Binär: 1, wenn Auftrag $a \in \mathcal{A}$ direkt vor Auftrag $a' \in \mathcal{A}$ auf Maschine $m \in \mathcal{M}$ zugewiesen worden ist; 0 in allen anderen Fällen

Entscheidungsvariablen:

C_a	Fertigstellungszeitpunkt des Auftrags $a \in \mathcal{A}$
τ_a	Bearbeitungszeit des Auftrags $a \in \mathcal{A}$
y_a	Binär: 1, wenn für Auftrag $a \in \mathcal{A}$ die Restriktion (5.32d) relaxiert werden soll; 0 in allen anderen Fällen

Zielfunktion:

$$\min c^{\text{dis}} = \sum_{a \in \mathcal{A}} \tau_a \quad (5.32a)$$

u. B. d. N.:

$$\begin{aligned} r_a + \tau_a = C_a & \quad , \forall a \in \mathcal{A}: \\ & (\exists m \in \mathcal{M}: x_{\rho,a,m} = 1 \wedge \\ & \forall a_i \in \mathcal{A}: (a_i, a) \notin \mathcal{E}) \end{aligned} \quad (5.32b)$$

$$\max(r_a + \tau_a, \max_{\forall (a_i, a) \in \mathcal{E}} (C_{a_i} + \tau_a)),$$

$$\forall a' \in \mathcal{A} \forall m \in \mathcal{M} (x_{a',a,m} C_{a'} + \tau_a) = C_a \quad , \forall a \in \mathcal{A} \quad (5.32c)$$

$$C_a + M y_a > d_a \quad , \forall a \in \mathcal{A} \quad (5.32d)$$

$$|\mathcal{A}| - U(ap_i^\eta, sz^{\text{ub}}) = \sum_{a \in \mathcal{A}} y_a \quad (5.32e)$$

$$C_a \geq 0 \quad , \forall a \in \mathcal{A} \quad (5.32f)$$

$$\tau_a \in I_a \quad , \forall a \in (\mathcal{A} \setminus \mathcal{A}_{t_i}^{\text{R}}) \quad (5.32g)$$

$$\tau_a \in I_a^{\text{Rest}} \quad , \forall a \in \mathcal{A}_{t_i}^{\text{R}} \quad (5.32h)$$

$$y_a \in \{0, 1\} \quad , \forall a \in \mathcal{A} \quad (5.32i)$$

Die Zielfunktion (5.32a) minimiert die Summe der Bearbeitungszeiten über alle Aufträge. Der Fertigstellungszeitpunkt C_a des jeweils ersten Auftrags a auf einer Maschine m , der keine Vorrangbeziehungen zu anderen Aufträgen besitzt, wird mit Hilfe der Restriktion (5.32b) festgelegt. Die Fertigstellungszeitpunkte der anderen Aufträge werden

über Nebenbedingung (5.32c) abgebildet. Für einen Auftrag ergibt sich der Fertigstellungszeitpunkt aus dem Maximum der Summe der eigenen Bereitstellungszeit und der Bearbeitungszeit, der Summe aus dem maximalen Fertigstellungszeitpunkt der über Vorrangbeziehungen verbundenen Aufträge und der eigenen Bearbeitungszeit sowie der Summe des Fertigstellungszeitpunktes des direkten Vorgängerauftrags auf der gleichen Maschine und der eigenen Bearbeitungszeit.⁸³ Ungleichung (5.32d) fordert, dass der Fertigstellungszeitpunkt C_a eines Auftrags a später als sein Fälligkeitstermin d_a liegt, sofern $y_a = 0$. Ist $y_a = 1$, so muss dieser Auftrag nicht säumig sein, da aufgrund des Terms My_a die Bedingung bereits eingehalten wird. Über Gleichung (5.32e) wird festgelegt, wie viele Aufträge nicht säumig sein müssen, um die Anzahl $U(ap_i^\eta, sz^{\text{ub}})$ säumiger Aufträge des Szenarios sz^{ub} zu erzielen. Die binäre Eigenschaft der Entscheidungsvariable y_a wird über die Nebenbedingungen (5.32i) sowie die Nichtnegativität der Entscheidungsvariable C_a über die Restriktionen (5.32f) sichergestellt. Über die Nebenbedingungen (5.32g) und (5.32h) wird schließlich die Bearbeitungszeit für Aufträge aus $(\mathcal{A} \setminus \mathcal{A}_{t_i}^{\text{R}})$ auf das Bearbeitungszeitintervall I_a und für Aufträge aus $\mathcal{A}_{t_i}^{\text{R}}$ auf das Restbearbeitungszeitintervall I_a^{Rest} begrenzt. Das Szenario $sz_\eta^{\text{dis}} = \{\tau_a \mid a \in \mathcal{A}\}$ ergibt sich aus den Entscheidungsvariablen τ_a nach Durchführung der Optimierung mit dem Modell (5.32).

Die während der Initialisierungsphase⁸⁴ erstellte Szenariomenge \mathcal{SZ} wird für die Evaluation eines Individuums η jeweils mit dem für dieses spezifisch erzeugte Szenario sz_η^{dis} zu der Szenariomenge $\mathcal{SZ}_\eta = \mathcal{SZ} \cup \{sz_\eta^{\text{dis}}\}$ kombiniert. Für diese Menge ist zunächst noch der optimale Ablaufplan $ap_{sz_\eta^{\text{dis}}}^*$ für das individuumspezifische Szenario sz_η^{dis} zu bestimmen (vgl. Gleichung (5.20)).⁸⁵ Anschließend können die Zielfunktionswerte berechnet und das Individuum η bewertet bzw. dessen Fitness ermittelt werden. Wird die Ungleichung (5.22) von Individuum η nicht eingehalten, so wird es mit entsprechenden Strafkosten sk_η belegt, die zu einer geringeren Auswahlwahrscheinlichkeit⁸⁶ in der im nächsten Abschnitt zu beschreibenden Selektionsphase führt. Die Strafkosten steigen dabei proportional mit der Überschreitung des Parameters ω an und werden wie folgt berechnet:

$$sk_\eta = \max(0, R^{\text{max}}(ap) - \omega) \quad (5.33)$$

Das gesamte zuvor beschriebene Vorgehen zur Evaluation einer Population \mathcal{P}_i ist in Algorithmus 5.8 aufgeführt.

5.3.3.4 Selektion und Rekombination von Ablaufplänen

Nachdem die Initialpopulation \mathcal{P}_0 erstellt und evaluiert worden ist, startet ein iterativer Prozess, in dem in jeder Iteration i durch die Anwendung genetischer Operatoren neue

⁸³Die Max-Funktion dieser Restriktion wird in dem Solver-Modell durch die Einführung von 0/1-Indikator-Variablen und der Modellierung von weiteren Ungleichungen aufgelöst.

⁸⁴Siehe Abschnitt 5.3.3.2.

⁸⁵Dies geschieht erneut über den in Fußnote 74 auf Seite 123 erläuterten genetischen Algorithmus.

⁸⁶Das Individuum wird nicht vollständig entfernt, da der Prozess, zulässige Lösungen zu finden, zunächst über nicht zulässige Lösungen führen kann.

Algorithmus 5.8: Ablauf der Evaluation einer Population

Input	: \mathcal{P}_i	–	Zu evaluierende Population
	ap_{i-1}	–	Aktuell ausgeführter Ablaufplan
	\mathcal{SZ}	–	Die für die Evaluation zu verwendende Szenariomen- ge
Output	: \mathcal{P}_i	–	Evaluierte Population

```

1 foreach  $\eta \in \mathcal{P}_i$  do
2   Fasse den Ablaufplan  $ap_{i-1}$  mit der durch das Individuum  $\eta$  kodierten Lösung
   zu dem Ablaufplan  $ap_i^\eta$  zusammen;
3   Für das Szenario  $sz^{ub}$  ermittle die Anzahl säumiger Aufträge;
4   Ermittle das Szenario  $sz_\eta^{dis}$  unter Verwendung von Modell (5.32);
5   Ermittle für das Szenario  $sz_\eta^{dis}$  den optimalen Ablaufplan  $ap_{sz_\eta^{dis}}^*$ ;
6   Erstelle die kombinierte Szenariomenge  $\mathcal{SZ}_\eta = \mathcal{SZ} \cup \{sz_\eta^{dis}\}$ ;
7   if  $ap_i^\eta$  nicht zulässig für Ungleichung (5.22) then
8     | Belege  $ap_i^\eta$  mit Strafkosten;
9     Ermitteln des Zielfunktionswerts  $c^{sched}(ap_i^\eta)$  (siehe Gleichung (5.18));
10    Ermitteln des Zielfunktionswerts  $r_\varphi^{sched}(ap_i^\eta)$  (siehe Gleichung (5.23));
11    Ermitteln des Zielfunktionswerts  $s^{sched}(ap_{i-1}, ap_i^\eta)$  (siehe Gleichung (5.28));
12 return  $\mathcal{P}_i$ ;

```

Individuen bzw. Ablaufpläne erzeugt werden. Der erste Operator, der angewendet wird, ist die Rekombination. Bei dieser werden aus der Kombination von je zwei Individuen der Population \mathcal{P}_i zwei neue Individuen erzeugt. Hierbei ist erneut die Zulässigkeit und Gültigkeit der erzeugten Lösungen zu gewährleisten. Für die Rekombination der Individuen wird hier ein von Corrêa et al.⁸⁷ vorgeschlagener Rekombinations-Operator verwendet.⁸⁸

Bevor eine Rekombination ausgeführt werden kann, müssen zwei Individuen aus \mathcal{P}_i ausgewählt werden. Dies geschieht über die zweimalige Anwendung eines binären Turnierselektionsmechanismus. Bei diesem werden zufällig zwei Individuen aus Population \mathcal{P}_i ausgewählt und auf Basis eines Vergleichsoperators miteinander verglichen, wobei das *bessere* Individuum zurückgeliefert wird. Der Vergleichsoperator⁸⁹ wird dem genetischen Algorithmus NSGA-II entnommen, der drei Werte eines Individuums miteinander vergleicht. Dies sind die Strafkosten für die Überschreitung des Parameters ω , die Dominanz bezüglich der Zielfunktionen sowie ein *crowding-distance*-Wert,⁹⁰ der angibt welchen Abstand ein Individuum zu seinen Nachbarn besitzt. Das erste für die Rekombination

⁸⁷Siehe [CFR99], S. 830 f.

⁸⁸Die Autoren nutzen die gleiche Kodierung wie Hou et al. [HAR94], verwenden jedoch einen anderen Rekombinations-Operator, da der ursprüngliche nicht alle zulässigen Lösungen erzeugen kann (siehe [CFR99], S. 829).

⁸⁹Siehe [DPAM02], S. 192.

⁹⁰Siehe ebd., S. 185 f., für eine Erläuterung, wie die *crowding-distance* berechnet werden kann.

ausgewählte Individuum wird als P_1 mit $P_1 \in \mathcal{P}_i$ und das zweite als P_2 mit $P_2 \in \mathcal{P}_i$ bezeichnet. Mit dem Parameter p^{cx} wird festgelegt, mit welcher Wahrscheinlichkeit für die beiden zuvor ausgewählten Individuen das nachfolgende Verfahren durchgeführt wird.

Das Verfahren basiert auf dem Vorranggraphen $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ (vgl. Abschnitt 2.2.2.2), der die Vorrangbeziehungen zwischen den Aufträgen repräsentiert. Für einen Ablaufplan, der sich aus der Dekodierung eines Individuums ergibt, erweitert sich die Menge \mathcal{E} um Vorrangbeziehungen für die Aufträge, die auf der gleichen Maschine eingeplant sind. Diese erweiterte Menge werde als \mathcal{E}_η bezeichnet, die wie folgt definiert ist:⁹¹

$$\begin{aligned} \mathcal{E}_\eta = \mathcal{E} \cup \{ & (a_i, a_j) \mid (a_i, a_j) \notin \mathcal{E}, \\ & \text{assigned}(a_i, \eta) = \text{assigned}(a_j, \eta) \wedge \\ & \text{pos}(a_i, \eta) = \text{pos}(a_j, \eta) - 1 \}, \end{aligned}$$

wobei $\text{assigned}(a_i, \eta)$ die Maschine zurückliefert, auf die Auftrag a_i in dem durch das Individuum η kodierten Ablaufplan eingeplant ist und $\text{pos}(a_i, \eta)$ die Position des Auftrags auf der zugewiesenen Maschine bezeichne. Des Weiteren bezeichne \mathcal{E}_η^+ die transitive Hülle von \mathcal{E}_η .

Zur Durchführung der Rekombination müssen die in P_1 und P_2 kodierten Aufträge zunächst in zwei Submengen unterteilt werden. Sei $\mathcal{S} = \mathcal{A}_{i_i}^{\text{pb}}$, dann werden die folgenden Schritte ausgeführt, solange $S \neq \emptyset$ gilt:⁹²

1. Wähle zufällig einen Auftrag $a \in S$ und eine Zufallszahl $\beta \in \{1, 2\}$.
2. Wenn $\beta = 1$, dann

$$\mathcal{S}_1 = \mathcal{S}_1 \cup \{a\} \cup \{a' \mid a' \in \mathcal{S} \wedge (a', a) \in (\mathcal{E}_{P_1}^+ \cup \mathcal{E}_{P_2}^+)\},$$

sonst

$$\mathcal{S}_2 = \mathcal{S}_2 \cup \{a\} \cup \{a' \mid a' \in \mathcal{S} \wedge (a, a') \in (\mathcal{E}_{P_1}^+ \cup \mathcal{E}_{P_2}^+)\},$$

wobei $\mathcal{E}_{P_1}^+$ bzw. $\mathcal{E}_{P_2}^+$ die transitiven Hüllen der aus den Individuen P_1 bzw. P_2 ermittelten Ablaufplänen bezeichne.

3. Setze $\mathcal{S} = \mathcal{S} \setminus \mathcal{S}_\beta$.

Das erste aus der Rekombination resultierende Individuum bzw. der Ablaufplan wird mit Ch_1 bezeichnet. Dieses wird erzeugt, indem die Aufträge aus \mathcal{S}_1 entsprechend der im Individuum P_1 kodierten Abfolge auf den kodierten Maschinen eingeplant werden. Für die Aufträge aus \mathcal{S}_2 wird folgendermaßen vorgegangen:⁹³

1. Wähle einen Auftrag $a \in \mathcal{S}_2$, der den kleinsten Wert für $\text{pos}(a, P_2)$ besitzt, bzw. zufällig, falls diese Bedingung auf mehrere Aufträge zutrifft.
2. Wähle als Maschine $\text{assigned}(a, P_2)$.

⁹¹Siehe [CFR99], S. 829.

⁹²Siehe ebd., S. 830.

⁹³Siehe ebd.

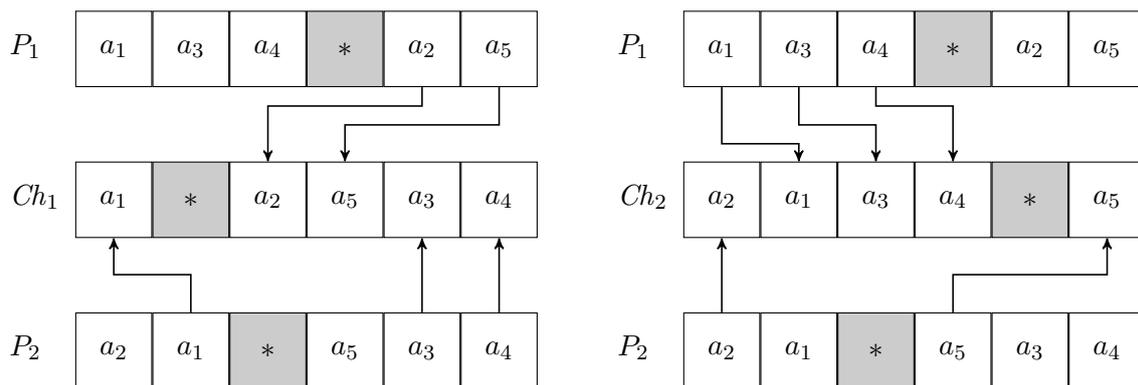


Abbildung 5.9: Rekombinationsoperator des genetischen Algorithmus

Das zweite aus der Rekombination resultierende Individuum bzw. der Ablaufplan wird mit Ch_2 bezeichnet und analog zu dem zuvor beschriebenen Vorgehen erzeugt. Hier werden die Aufträge aus \mathcal{S}_1 allerdings nach der Reihenfolge aus P_2 und die Aufträge aus \mathcal{S}_2 nach dem oben beschriebenen Verfahren wie in P_1 eingeplant. Das beschriebene Vorgehen stellt sicher, dass aus den zwei Ausgangsindividuen, die jeweils eine zulässige Lösung bzgl. der Vorrangbeziehungen darstellen, zwei Individuen resultieren, die die Vorrangbeziehungen ebenfalls einhalten.⁹⁴

In Abbildung 5.9 ist ein Beispiel für den zuvor beschriebenen Rekombinationsoperator aufgeführt. Es sind insgesamt fünf Aufträge auf zwei Maschinen einzuplanen. Der Vorranggraph ist mit $\mathcal{E} = \{(a_2, a_4), (a_2, a_5)\}$ gegeben. Für die ausgewählten Individuen P_1 und P_2 gilt entsprechend $\mathcal{E}_{P_1} = \{(a_2, a_4), (a_2, a_5), (a_1, a_3), (a_3, a_4)\}$ und $\mathcal{E}_{P_2} = \{(a_2, a_4), (a_2, a_5), (a_2, a_1), (a_5, a_3), (a_3, a_4)\}$. Als Submengen wurden durch die Wahl von a_5 mit $\beta = 1$ und a_1 mit $\beta = 2$ die Submengen $\mathcal{S}_1 = \{a_5, a_2\}$ und $\mathcal{S}_2 = \{a_1, a_3, a_4\}$ generiert. Die aus der Rekombination resultierenden Individuen Ch_1 und Ch_2 sind in der Abbildung aufgeführt und die übernommenen Gene durch Pfeile markiert.

Das zuvor beschriebene Vorgehen wird anschließend $(popsize/2)$ -mal durchgeführt, um die Nachfolgepopulation \mathcal{Q}_i zu erstellen. Jedes erstellte Individuum Ch_k mit $1 \leq k \leq popsize$ wird anschließend der Liste der Nachfolgepopulation \mathcal{Q}_i hinzugefügt.

5.3.3.5 Mutationsoperator für Ablaufpläne

Die Mutation von Individuen ist eine weitere Möglichkeit, veränderte Ablaufpläne zu generieren. Sie wird für alle Individuen der im vorherigen Schritt erzeugten Nachfolgegeneration \mathcal{Q}_i durchgeführt. Hierbei ist erneut die Zulässigkeit und Gültigkeit der erzeugten Lösungen zu gewährleisten. Es können zwei Arten von Mutationen durchgeführt werden. Die erste ändert die Auftragsreihenfolge der auf einer Maschine eingeplanten Aufträge. Die zweite Mutation verändert die Maschine, auf der ein Auftrag ausgeführt wird. Für ein Individuum wird immer nur eine der beiden Mutationsarten ausgeführt. Das Verhältnis der ersten zur zweiten Mutationsart wird dabei über den Parameter p^{mut} gesteuert.

⁹⁴Für die entsprechenden Beweise sei auf Corrêa et al. [CFR99] verwiesen.

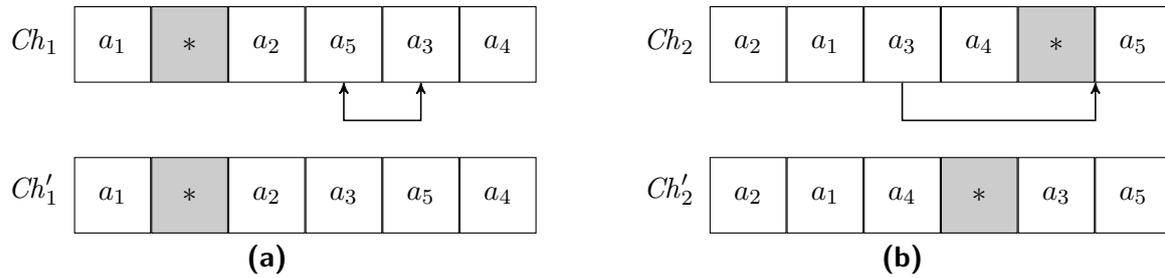


Abbildung 5.10: Mutationsoperatoren des genetischen Algorithmus: (a) Erste Mutationsart
(b) Zweite Mutationsart

Für die folgenden Ausführungen bezeichne \mathcal{E}_{Ch_1} bzw. \mathcal{E}_{Ch_2} die jeweiligen Vorranggraphen der Individuen Ch_1 und Ch_2 .

Die erste Mutationsart wird durchgeführt, indem für jedes Gen g_j mit $g_j \in \mathcal{A}_{t_i}^{\text{Pb}}$ und $1 \leq j \leq (|\mathcal{A}_{t_i}^{\text{Pb}}| + |\mathcal{M}| - 1)$ eines Individuums $\eta \in \mathcal{Q}_i$ mit einer Wahrscheinlichkeit von pa^{mut} der in diesem Gen kodierte Auftrag mit dem in einem anderen Gen $g_k \in \mathcal{A}_{t_i}^{\text{Pb}}$ kodierten Auftrag vertauscht wird. Für die Auswahl von g_k gelte zudem die Zuordnung zur gleichen Maschine $\text{assigned}(g_j, \eta) = \text{assigned}(g_k, \eta)$ sowie eine direkte Vorgängerbeziehung $\text{pos}(g_j, \eta) = \text{pos}(g_k, \eta) - 1$. Die Vertauschung wird allerdings nur durchgeführt, sofern der sich ergebene Vorranggraph $\mathcal{E}_{Ch'_1}$ ein gerichteter azyklischer Graph ist. Eine Visualisierung dieser Mutationsart ist in Abbildung 5.10(a) aufgeführt. In dem Beispiel wird das aus der vorherigen Rekombination erzeugte Individuum Ch_1 an einer Stelle einer Mutation unterzogen. Die Aufträge a_5 und a_3 auf Maschine m_2 werden hierbei getauscht und somit das Individuum Ch'_1 erzeugt.

Die zweite Mutationsart verändert die Maschinenzuordnung. Hierfür wird für ein Gen g_j mit $g_j \in \mathcal{A}_{t_i}^{\text{Pb}}$ und $1 \leq j \leq (|\mathcal{A}_{t_i}^{\text{Pb}}| + |\mathcal{M}| - 1)$ eines Individuums $\eta \in \mathcal{Q}_i$ mit einer Wahrscheinlichkeit von pm^{mut} dieses Gen auf eine andere zulässige Maschine $m \in \mathcal{M}_{g_j}$ mit $m \neq \text{assigned}(g_j, \eta)$ verschoben. Auf dieser Maschine wird es an der ersten möglichen Position eingefügt, sodass der sich ergebene Vorranggraph $\mathcal{E}_{Ch'_2}$ weiterhin ein gerichteter azyklischer Graph ist. Abbildung 5.10(b) zeigt die Anwendung dieser Mutationsart für das Individuum Ch_2 aus der vorherigen Rekombination. Der Auftrag a_3 wird hierbei von Maschine m_1 auf Maschine m_2 verschoben und somit das Individuum Ch'_2 erzeugt.

5.3.3.6 Ersetzungsstrategie für die Ablaufpläne

In diesem Schritt werden die Individuen ausgewählt, die in der nächsten Iteration des Algorithmus verwendet werden sollen. Hierfür wird zunächst eine kombinierte Population $\mathcal{R}_i = \mathcal{P}_i \cup \mathcal{Q}_i$ der Größe $2 \cdot \text{popsize}$ gebildet, die aus der Ausgangs- und Nachfolgepopulation besteht. Aus dieser sind popsize Individuen für die Population \mathcal{P}_{i+1} zu bestimmen.

Als Ersetzungsstrategie wird hier die im NSGA-II-Algorithmus verwendete Strategie eingesetzt.⁹⁵ Diese ermittelt zunächst alle Fronten \mathcal{F}_j nicht-dominierter Individuen

⁹⁵Siehe [DPAM02], S. 185 f.

aus der kombinierten Population \mathcal{R}_i .⁹⁶ Die erste Front \mathcal{F}_1 enthält hierbei alle nicht dominierten bzw. pareto-optimalen⁹⁷ Individuen der Population \mathcal{R}_i . Alle weiteren \mathcal{F}_j mit $j > 1$ enthalten jeweils die Individuen, die ausschließlich von Individuen der vorherigen Front dominiert werden. Die Fronten werden beginnend mit \mathcal{F}_1 der Population \mathcal{P}_{i+1} hinzugefügt, bis zu Front \mathcal{F}_l , die nicht vollständig hinzugefügt werden kann, da hierdurch das Limit von $|\mathcal{P}_{i+1}| = \text{popsize}$ überschritten würde. Für die Individuen aus \mathcal{F}_l wird daher der *crowding-distance*-Wert berechnet. Ein höherer Wert für die Distanz ist mit einer höheren Diversität verbunden. Daher werden die Individuen aus \mathcal{F}_l mit den höchsten *crowding-distance*-Werten verwendet, um die verbleibenden $(\text{popsize} - |\mathcal{P}_{i+1}|)$ -Stellen aufzufüllen.

5.3.3.7 Terminierung und Auswahl eines Ablaufplans durch den Entscheidungsträger

Der iterative Prozess ist nach einer gewissen Anzahl von Iterationen abubrechen, da es kein spezifisches Terminierungskriterium gibt. Hier wird daher die Anzahl an durchlaufenden Generationen i gewählt und eine maximale Anzahl an Iterationen maxgen vorgegeben. Sobald $i > \text{maxgen}$ gilt, wird das Verfahren abgebrochen und die nicht-dominierten Ablaufpläne der Population $\mathcal{P}_{\text{maxgen}}$ werden zurückgeliefert.

Eine Terminierung kann darüber hinaus zu jedem beliebigen Zeitpunkt während des iterativen Prozesses stattfinden und die bis dahin ermittelten nicht-dominierten Ablaufpläne können als Ergebnis zurückgegeben werden. Eine vorzeitige Unterbrechung ist bspw. notwendig, wenn nur eine begrenzte Zeit für die Ablaufplanung zur Verfügung steht und diese Grenze eher erreicht ist als die durch maxgen vorgegebene maximale Anzahl an Iterationen.

Aus der Menge $\mathcal{P}_{\text{maxgen}}$ ist schließlich durch den Entscheidungsträger ein Ablaufplan ap^p zu bestimmen.⁹⁸ Dieser Ablaufplan wird als aktiver Ablaufplan markiert und auf der reaktiven Ebene der Architektur gespeichert und durch die Ablaufsteuerung ausgeführt.

5.3.4 Integration der Personaleinsatzplanung

In den Ausführungen der vorherigen Abschnitte wurden lediglich Maschinen als einschränkende Ressource betrachtet. Im Folgenden soll die Betrachtung auf die Einbeziehung von Personalressourcen erweitert werden. Hierzu wird in Abschnitt 5.3.4.1 zunächst die Modellformulierung erweitert und anschließend in Abschnitt 5.3.4.2 eine Heuristik zur Einplanung des Personals vorgestellt.

⁹⁶Die Ermittlung der Dominanz erfolgt, wie bei der Selektion (vgl. Abschnitt 5.3.3.4), auf Basis des Vergleichsoperators, der die Strafkosten für die Überschreitung des Parameters ω , die Dominanz bezüglich der Zielfunktionen sowie die *crowding-distance* eines Individuums miteinander vergleicht.

⁹⁷Vgl. Ungleichung (2.3) auf Seite 28.

⁹⁸Für die Entscheidung sind durch den Entscheidungsträger Präferenzen bezüglich der Zielfunktionen anzugeben, sodass bspw. auf Basis der aktuellen Situation einer Zielfunktion eine höhere Relevanz zugeordnet wird (vgl. z. B. [ZB12] für A-posteriori-Ansätze).

5.3.4.1 Erweiterung der Modellformulierung

Ausgehend von dem in Abschnitt 2.2.2.3 beschriebenen Problem, das bei der Berücksichtigung von Personalkapazitäten besteht, kann das Mehrzieloptimierungsmodell (5.34)⁹⁹ formalisiert werden. Das Modell stellt hierbei eine Erweiterung der Optimierungsmodelle (5.25) und (5.30) dar. Aus diesen werden die Zielfunktionen und die dargestellten Restriktionen übernommen. Des Weiteren ist die Zuordnungslösung des Personals $y'_{a,a',wo}$ aus dem Ablaufplan ap_{i-1} , der zum Zeitpunkt t_{i-1} erstellt worden ist, für die Verwendung in dem Modell (5.34) aufzubereiten. Hierzu wird analog zur Aufbereitung der Maschinenzuordnung (vgl. Abschnitt 5.3.2.3) vorgegangen.

Mengen:

\mathcal{A}	Menge der zum i -ten Planungszeitpunkt t_i zu berücksichtigenden Aufträge a , die eine Auswirkung auf die aktuelle Planung haben. Die Menge ergibt sich folgendermaßen: $\mathcal{A} = \mathcal{A}_{t_i}^N \cup (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$
$\mathcal{A}_a^{\text{prec}}$	Menge aller Aufträge, die vor Auftrag a eingeplant werden können
$\mathcal{A}_a^{\text{succ}}$	Menge aller Aufträge, die nach Auftrag a eingeplant werden können
$\mathcal{A}^{\text{Bu/Tr}}$	Menge der Aufbau bzw. Transferaufträge. Es gilt: $\mathcal{A}^{\text{Bu/Tr}} \subseteq \mathcal{A}$
$\mathcal{A}_{t_i}^R$	Menge der zum Zeitpunkt t_i bekannten Aufträge a , die bereits auf den Arbeitsstationen gestartet wurden
\mathcal{SZ}	Menge aller Szenarien sz , die eine mögliche Realisierung der Bearbeitungszeiten der Aufträge angeben
\mathcal{WO}	Menge des zur Verfügung stehenden Personals wo
\mathcal{WO}^{Br}	Menge des Personals, das nur für Abbauaufträge eingesetzt werden kann. Es gilt: $\mathcal{WO}^{\text{Br}} \subset \mathcal{WO}$

Parameter:

b_a	Bedarf an Mitarbeitern für Auftrag $a \in \mathcal{A}$
M	Big M
ρ	Pseudoauftrag, dessen Bearbeitungszeit, Bereitstellungszeit sowie Gewicht 0 ist. Es gilt: $\rho \in \mathcal{A}$ sowie $\forall a \in \mathcal{A}: \rho \in \mathcal{A}_a^{\text{prec}} \wedge \rho \in \mathcal{A}_a^{\text{succ}}$
τ_a^{sz}	Bearbeitungszeit des Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
$y'_{a,a',wo}$	Die Mitarbeiterzuordnungen aus dem Ablaufplan ap_{i-1} der zum Zeitpunkt t_{i-1} erstellt worden ist. Binär: 1, wenn Auftrag $a \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ direkt vor Auftrag $a' \in (\mathcal{A}_{t_{i-1}} \setminus \mathcal{A}_{t_i}^C)$ von Mitarbeiter $wo \in \mathcal{WO}$ bearbeitet worden ist; 0 in allen anderen Fällen

Entscheidungsvariablen:

C_a^{sz}	Fertigstellungszeitpunkt des Auftrags $a \in \mathcal{A}$ in Szenario $sz \in \mathcal{SZ}$
$y_{a,a',wo}$	Binär: 1, wenn Auftrag $a \in \mathcal{A}$ direkt vor Auftrag $a' \in \mathcal{A}$ von Mitarbeiter $wo \in \mathcal{WO}$ bearbeitet wird; 0 in allen anderen Fällen

⁹⁹Die Grundlage dieses Modells bildet eine Netzwerkformulierung, die ein Tourenplanungsproblem mit aufgeteilten Auslieferungen (engl. *Split Delivery Vehicle Routing Problem*) beschreibt. Hierbei stellen die Kunden die Aufträge, die Fahrzeuge die Mitarbeiter und eine Tour einen Ablaufplan für einen Mitarbeiter dar (vgl. z. B. [AS08], S. 104f.). Im Gegensatz zu dem klassischen *Vehicle Routing Problem* kann ein Kunde bzw. Auftrag von mehreren Fahrzeugen bzw. Mitarbeitern beliefert bzw. bearbeitet werden.

Zielfunktionen:

(5.25a), (5.25b) und (5.30a)

u. B. d. N.:

(5.25c)–(5.25s) und (5.30b)–(5.30j)

$$\sum_{a \in \mathcal{A}: a \neq \rho} y_{a,\rho,wo} \leq 1, \forall wo \in \mathcal{WO} \quad (5.34a)$$

$$\sum_{a' \in \mathcal{A}: a' \neq \rho} y_{\rho,a',wo} \leq 1, \forall wo \in \mathcal{WO} \quad (5.34b)$$

$$\sum_{wo \in \mathcal{WO}} \sum_{a \in \mathcal{A}_a^{\text{prec}}: a \neq a'} y_{a,a',wo} \geq 1, \forall a' \in \mathcal{A}: a' \neq \rho \quad (5.34c)$$

$$\sum_{wo \in \mathcal{WO}} \sum_{a' \in \mathcal{A}_a^{\text{succ}}: a' \neq a} y_{a,a',wo} \geq 1, \forall a \in \mathcal{A}: a \neq \rho \quad (5.34d)$$

$$\sum_{a \in \mathcal{A}_a^{\text{prec}}: a \neq a'} y_{a,a',wo} - \sum_{a \in \mathcal{A}_a^{\text{succ}}: a \neq a'} y_{a',a,wo} = 0, \forall a' \in \mathcal{A}: a' \neq \rho; \forall wo \in \mathcal{WO} \quad (5.34e)$$

$$C_a^{\text{sz}} - C_{a'}^{\text{sz}} + (M - r_{a'})y_{a,a',wo} \leq M - (r_{a'} + \tau_{a'}^{\text{sz}}), \forall a \in \mathcal{A}; \forall a' \in \mathcal{A}: a' \neq \rho, a \neq a'; \forall wo \in \mathcal{WO}; \forall sz \in \mathcal{SZ} \quad (5.34f)$$

$$\sum_{wo \in \mathcal{WO}} \sum_{a' \in \mathcal{A}} y_{a,a',wo} = b_a, \forall a \in \mathcal{A}: a \neq \rho \quad (5.34g)$$

$$\sum_{a \in \mathcal{A}} y_{a,a',wo} = 0, \forall a' \in \mathcal{A}^{\text{Bu/Tr}}: a' \neq \rho; \forall wo \in \mathcal{WO}^{\text{Br}} \quad (5.34h)$$

$$\sum_{a' \in \mathcal{A}} y_{a,a',wo} = 0, \forall a \in \mathcal{A}^{\text{Bu/Tr}}: a \neq \rho; \forall wo \in \mathcal{WO}^{\text{Br}} \quad (5.34i)$$

$$y_{a,a',wo} = y'_{a,a',wo}, \forall a \in \mathcal{A}_{t_i}^{\text{R}}; \forall a' \in \mathcal{A}_{t_i}^{\text{R}}; \forall wo \in \mathcal{WO} \quad (5.34j)$$

$$y_{a,a',wo} \in \{0, 1\}, \forall a \in \mathcal{A}; \forall a' \in \mathcal{A}; \forall wo \in \mathcal{WO} \quad (5.34k)$$

Die Nebenbedingungen (5.34a) und (5.34b) stellen sicher, dass für jeden Mitarbeiter höchstens ein Auftrag direkt vor bzw. direkt nach dem Pseudoauftrag ρ eingeplant wird.¹⁰⁰ Die (Un-)Gleichungen (5.34c), (5.34d) und (5.34g) sorgen dafür, dass jeder Auftrag von

¹⁰⁰Sind für einen Mitarbeiter keine Aufträge eingeplant, so besitzt der Pseudoauftrag für diesen Mitarbeiter keinen Vorgänger oder Nachfolger.

mindestens einem bzw. genau der benötigten Anzahl Mitarbeiter bearbeitet wird. Die Restriktion (5.34e) stellt sicher, dass jeder Auftrag (ausgenommen des Pseudoauftrags) genauso viele Nachfolger wie Vorgänger besitzt. Der Fertigstellungszeitpunkt C_a^{sz} eines Auftrags a in Szenario sz wird mit Hilfe der Restriktion (5.34f) ermittelt. Über die Gleichungen (5.34h) und (5.34i) wird sichergestellt, dass jeder Aufbau- bzw. Transferauftrag lediglich von Mitarbeitern bearbeitet wird, die hierfür qualifiziert sind. Über die Restriktion (5.34j) wird sichergestellt, dass Aufträge aus der Menge $\mathcal{A}_{l_i}^R$ nicht umgeplant werden können.¹⁰¹ Hierfür werden die Zuordnungen aus der modifizierten vorherigen Zuordnungslösung $y'_{a,a',wo}$ in der aktuellen $y_{a,a',wo}$ fixiert. Die binäre Eigenschaft der Entscheidungsvariable $y_{a,a',wo}$ wird über die Nebenbedingung (5.34k) sichergestellt.

5.3.4.2 Heuristik für die Personaleinsatzplanung

Wie zuvor die Modelle (5.25) und (5.30) kann auch das im vorherigen Abschnitt aufgestellte Modell (5.34) nicht direkt bzw. nur mit großem Aufwand durch einen Solver gelöst werden. Im Folgenden wird daher eine Heuristik zur Einplanung des Personals entwickelt, die in der beschriebenen Metaheuristik (vgl. Abschnitt 5.3.3) verwendet wird. Hierzu wird nach der Dekodierung eines Individuums η zu dem Ablaufplan ap_i^η die nachfolgende Heuristik ausgeführt, bevor die Evaluation auf Basis der drei beschriebenen Zielfunktionen erfolgt.

Zunächst wird für jeden Auftrag die längste Bearbeitungszeit angenommen. Für noch nicht gestartete Aufträge ist dies τ_a^{\max} und für bereits gestartete Aufträge die über Gleichung (5.26) ermittelte maximale Restbearbeitungszeit $\max(I_a^{\text{Rest}})$. Mit Hilfe dieses Vorgehens soll die heuristisch ermittelte Mitarbeiterzuordnung möglichst unabhängig bzgl. schwankender Bearbeitungszeiten innerhalb der Intervalle sein. Die Aufträge werden anschließend in aufsteigender Reihenfolge ihres Starttermins geordnet.

Für den Auftrag mit dem frühesten Startzeitpunkt wird anschließend ein Mitarbeiter mit den nötigen Fähigkeiten ausgewählt, der diesem Auftrag zugeordnet wird. Hierbei sind drei Fälle zu unterscheiden:¹⁰²

1. Alle Mitarbeiter mit den nötigen Fähigkeiten sind bereits vor dem Startzeitpunkt des Auftrags verfügbar. Wähle den Mitarbeiter, der die späteste Verfügbarkeitszeit besitzt, um Leerlaufzeiten zu minimieren.
2. Alle Mitarbeiter mit den nötigen Fähigkeiten sind erst nach dem Startzeitpunkt des Auftrags verfügbar. Wähle den Mitarbeiter, der die früheste Verfügbarkeitszeit besitzt, um Wartezeiten des Auftrags zu minimieren.
3. Einige Mitarbeiter mit den nötigen Fähigkeiten sind bereits vor dem Startzeitpunkt des Auftrags verfügbar, während andere hingegen erst danach zur Verfügung stehen. Verfahre analog zu Fall 1 und ggf. Fall 2.

¹⁰¹Im Gegensatz zu Gleichung (5.30e), mit der sichergestellt wird, dass sowohl laufende als auch fixierte Aufträge nicht auf eine andere Maschine umgeplant werden, gilt dies bei den Mitarbeitern nur für laufende Aufträge. Für fixierte Aufträge können die Mitarbeiter umgeplant werden, da dies im betrachteten Fall nicht mit zusätzlichen Kosten verbunden ist.

¹⁰²Vgl. [ML13], S. 5066.

Das beschriebene Vorgehen wird sooft für den Auftrag wiederholt, bis die nötige Anzahl an Mitarbeitern d_a erreicht ist. Analog werden die weiteren Aufträge behandelt. Die Evaluation des Individuums η kann anschließend unter Berücksichtigung der zugeordneten Mitarbeiter für die drei Zielfunktionen erfolgen.

5.3.5 Integration der Transportsysteme

Transportsysteme stellen nach den Ausführungen in Abschnitt 2.2.2.4 weitere bei der Ablaufplanung zu berücksichtigende Ressourcen dar. In den folgenden Abschnitten wird die Ablaufplanung dahingehend angepasst. In Abschnitt 5.3.5.1 wird zunächst die Modellformulierung erweitert. Hierauf aufbauend wird in Abschnitt 5.3.5.2 die evolutionäre Metaheuristik erweitert, um die Transportsysteme zu berücksichtigen.

5.3.5.1 Erweiterung der Modellformulierung

Zur Erweiterung der Modellformulierung um die Betrachtung der Transportsysteme wird das Transportsystem als Graph modelliert und die durch den Ablaufplan erzeugten Flüsse werden innerhalb dieses Systems als Netzwerkflussproblem (spezifischer: *Multi-commodity flow problem*) dargestellt.

Für das Transportsystem werden zwei identische¹⁰³ Netzwerke $\mathcal{G}^{\text{Br}} = (\mathcal{V}^{\text{Br}}, \mathcal{E}^{\text{Br}})$ für Abbauaufträge (Break-Netzwerk) und $\mathcal{G}^{\text{Bu}} = (\mathcal{V}^{\text{Bu}}, \mathcal{E}^{\text{Bu}})$ für Aufbauaufträge (Build-Netzwerk) verwendet. Bei den Knoten \mathcal{V}^{Br} bzw. \mathcal{V}^{Bu} kann es sich entweder um Maschinen \mathcal{M} , Lagerbereiche \mathcal{L} oder um Transferknoten (z. B. Pufferplätze, Gablungen, etc.) handeln. Die ersten beiden können einen bestimmten Bedarf besitzen bzw. ein bestimmtes Angebot liefern, wohingegen Transferknoten die Fracht lediglich weiterleiten. Der Bedarf bzw. das Angebot an einer Maschine ergibt sich aus den Aufträgen, die auf diese Maschine eingeplant sind. Die Mengen für verschiedene Typen von Ladeeinheiten, die durch einen Auftrag a benötigt \mathcal{K}_a^- bzw. bereitgestellt \mathcal{K}_a^+ werden, ergeben sich folgendermaßen:

- Abbauaufträge:
 - \mathcal{K}_a^- : die Menge besteht aus der dem Auftrag zugeordneten beladenen Flugzeugladeeinheit sowie leeren Gitterboxen
 - \mathcal{K}_a^+ : die Menge besteht aus der abgebauten Flugzeugladeeinheit sowie den beladenen Gitterboxen
- Aufbauaufträge:
 - \mathcal{K}_a^- : die Menge besteht aus der dem Auftrag zugeordneten leeren Flugzeugladeeinheit sowie beladenen Gitterboxen
 - \mathcal{K}_a^+ : die Menge besteht aus der aufgebauten Flugzeugladeeinheit sowie den entladenen Gitterboxen

¹⁰³Für die Formulierung in Modell (5.35) werden zwei Netzwerke verwendet, da Aufbauaufträge zum Teil Fracht benötigen, die zuvor durch Abbauaufträge verfügbar gemacht werden muss. Transferaufträge können als Kombination aus Auf- und Abbauauftrag betrachtet werden.

Die Kanten \mathcal{E}^{Br} bzw. \mathcal{E}^{Bu} in den beiden Netzwerken beschreiben die Verbindungen (z. B. Transferfahrzeuge, Aufzüge, Förderbänder etc.) zwischen den Knoten. Über diese Kanten können verschiedene Typen von Ladeeinheiten $k \in \mathcal{K}$ transportiert werden. Für jede Kante (i, j) der Netzwerke wird eine maximale Transportkapazität $c_{i,j}$ festgelegt sowie unter Berücksichtigung der Ladeeinheit k eine spezifische Transportzeit $t_{i,j,k}$ definiert.¹⁰⁴ Aufbauend auf diesen beiden Netzwerken kann das Modell (5.35) formalisiert werden, das eine Erweiterung der Optimierungsmodelle (5.25), (5.30) und (5.34) darstellt.

Mengen:

\mathcal{A}^{Br}	Menge der Abbauaufträge
\mathcal{A}^{Bu}	Menge der Aufbauaufträge
\mathcal{E}^{Br}	Menge der Transportstrecken im Break-Netzwerk des Transportsystems
\mathcal{E}^{Bu}	Menge der Transportstrecken im Build-Netzwerk des Transportsystems
\mathcal{K}	Menge aller Typen von Ladeeinheiten. Es gilt: $\mathcal{K} = \bigcup_{a \in \mathcal{A}} (\mathcal{K}_a^+ \cup \mathcal{K}_a^-)$
\mathcal{K}_a^+	Menge an Typen von Ladeeinheiten, die von $a \in \mathcal{A}$ angeboten werden
\mathcal{K}_a^-	Menge an Typen von Ladeeinheiten, die von $a \in \mathcal{A}$ benötigt werden
\mathcal{L}	Menge der Knoten im Netzwerk, an denen Ladeeinheiten gelagert werden können. Es gilt: $\mathcal{L} \subset \mathcal{V}^{\text{Br}}$ bzw. $\mathcal{L} \subset \mathcal{V}^{\text{Bu}}$
\mathcal{M}	Menge der zur Verfügung stehenden Maschinen m . Es gilt: $\mathcal{M} \subset \mathcal{V}^{\text{Br}}$ bzw. $\mathcal{M} \subset \mathcal{V}^{\text{Bu}}$
\mathcal{V}^{Br}	Menge der Knoten im Break-Netzwerk des Transportsystems
\mathcal{V}^{Bu}	Menge der Knoten im Build-Netzwerk des Transportsystems

Parameter:

cs_i	Maximale Lagerkapazität im Lagerbereich $i \in \mathcal{L}$. Es gilt, dass $c_{i,j}$ für das Break- und Build-Netzwerk jeweils der Hälfte der im tatsächlichen Transportsystem vorhandenen Lagerkapazität für diesen Lagerbereich entspricht
$ct_{i,j}$	Maximale Transportkapazität auf der Strecke $(i, j) \in \mathcal{E}^{\text{Br}}$ bzw. $(i, j) \in \mathcal{E}^{\text{Bu}}$. Es gilt, dass $c_{i,j}$ für das Break- und Build-Netzwerk jeweils der Hälfte der im tatsächlichen Transportsystem vorhandenen Kapazität für diese Kante entspricht
$t_{i,j,k}$	Transportzeit auf der Strecke $(i, j) \in \mathcal{E}^{\text{Br}}$ bzw. $(i, j) \in \mathcal{E}^{\text{Bu}}$ für eine Ladeeinheit vom Typ $k \in \mathcal{K}$
$\zeta_{i,k}$	Binär: 1, wenn Ladeeinheit vom Typ $k \in \mathcal{K}$ im Lagerbereich $i \in \mathcal{L}$ gelagert werden kann oder wenn sie den Knoten $i \in \mathcal{V}^{\text{Br}}$ bzw. $i \in \mathcal{V}^{\text{Bu}}$ passieren kann; 0 in allen anderen Fällen

Entscheidungsvariablen:

$b_{i,k}$	Bedarf ($b_{i,k}$ negativ) bzw. Angebot ($b_{i,k}$ positiv) an Ladeeinheiten vom Typ $k \in \mathcal{K}$ an Knoten $i \in \mathcal{V}^{\text{Br}}$ bzw. $i \in \mathcal{V}^{\text{Bu}}$. Gilt $b_{i,k} = 0$, so handelt es sich um einen Knoten, an dem die Ladeeinheit weitergesendet wird
$f_{i,j,k}$	Binär: 1, wenn eine Ladeeinheit vom Typ $k \in \mathcal{K}$ auf der Strecke $(i, j) \in \mathcal{E}^{\text{Br}}$ bzw. $(i, j) \in \mathcal{E}^{\text{Bu}}$ transportiert wird; 0 in allen anderen Fällen
r'_a	Zeitbedarf zur Bereitstellung der ersten Ladeeinheit für $a \in \mathcal{A}$

¹⁰⁴Diese Angaben sind abhängig von den technischen Spezifikationen der Fördertechnik.

Zielfunktionen:

$$(5.25a), (5.25b) \text{ und } (5.30a)$$

u. B. d. N.:

$$(5.25c)–(5.25s), (5.30b)–(5.30j) \text{ und } (5.34a)–(5.34k)$$

$$\sum_{(i,j) \in \mathcal{E}^{\text{Br}}} \zeta_{i,k} f_{i,j,k} - \sum_{(j,i) \in \mathcal{E}^{\text{Br}}} \zeta_{i,k} f_{j,i,k} = b_{i,k} \quad , \forall i \in \mathcal{V}^{\text{Br}}; \forall k \in \mathcal{K} \quad (5.35a)$$

$$\sum_{(i,j) \in \mathcal{E}^{\text{Bu}}} \zeta_{i,k} f_{i,j,k} - \sum_{(j,i) \in \mathcal{E}^{\text{Bu}}} \zeta_{i,k} f_{j,i,k} = b_{i,k} \quad , \forall i \in \mathcal{V}^{\text{Bu}}; \forall k \in \mathcal{K} \quad (5.35b)$$

$$\sum_{k \in \mathcal{K}} f_{i,j,k} + \sum_{k \in \mathcal{K}} f_{i',j',k} \leq ct_{i,j} + ct_{i',j'} \quad , \forall (i,j) \in \mathcal{E}^{\text{Br}}; \forall (i',j') \in \mathcal{E}^{\text{Bu}} \quad (5.35c)$$

$$-\sum_{a' \in \mathcal{A}: a' \neq a} |\mathcal{K}_a^-| x_{a,a',m} = \sum_{k \in \mathcal{K}_a^-} b_{m,k} \quad , \forall a \in \mathcal{A}^{\text{Br}}: a \neq \rho; \quad \forall m \in \mathcal{M} \subset \mathcal{V}^{\text{Br}} \quad (5.35d)$$

$$\sum_{a' \in \mathcal{A}: a' \neq a} |\mathcal{K}_a^+| x_{a,a',m} = \sum_{k \in \mathcal{K}_a^+} b_{m,k} \quad , \forall a \in \mathcal{A}^{\text{Br}}: a \neq \rho; \quad \forall m \in \mathcal{M} \subset \mathcal{V}^{\text{Br}} \quad (5.35e)$$

$$\sum_{\forall i \in \mathcal{L} \subset \mathcal{V}^{\text{Br}}} \sum_{k \in \mathcal{K}_a^+} \zeta_{i,k} b_{i,k} = -|\mathcal{K}_a^+| \quad , \forall a \in \mathcal{A}^{\text{Br}}: a \neq \rho \quad (5.35f)$$

$$b_{i',k} = (-1)b_{i,k} \quad , \forall i' \in \mathcal{L} \subset \mathcal{V}^{\text{Bu}} \wedge \quad \forall i \in \mathcal{L} \subset \mathcal{V}^{\text{Br}}: i' = i; \quad \forall k \in \mathcal{K} \quad (5.35g)$$

$$\min_{k \in \mathcal{K}_a^-} \left(\sum_{(i,j) \in (\mathcal{E}^{\text{Br}} \cup \mathcal{E}^{\text{Bu}})} t_{i,j,k} f_{i,j,k} \right) = r'_a \quad , \forall a \in (\mathcal{A}^{\text{Br}} \cup \mathcal{A}^{\text{Bu}}) \quad (5.35h)$$

$$-\sum_{a' \in \mathcal{A}: a' \neq a} |\mathcal{K}_a^-| x_{a,a',m} = \sum_{k \in \mathcal{K}_a^-} b_{m,k} \quad , \forall a \in \mathcal{A}^{\text{Bu}}: a \neq \rho; \quad \forall m \in \mathcal{M} \subset \mathcal{V}^{\text{Bu}} \quad (5.35i)$$

$$\sum_{a' \in \mathcal{A}: a' \neq a} |\mathcal{K}_a^+| x_{a,a',m} = \sum_{k \in \mathcal{K}_a^+} b_{m,k} \quad , \forall a \in \mathcal{A}^{\text{Bu}}: a \neq \rho; \quad \forall m \in \mathcal{M} \subset \mathcal{V}^{\text{Bu}} \quad (5.35j)$$

$$\sum_{\forall i \in \mathcal{L} \subset \mathcal{V}^{\text{Bu}}} \sum_{k \in \mathcal{K}_a^+} \zeta_{i,k} b_{i,k} = -|\mathcal{K}_a^+| \quad , \forall a \in \mathcal{A}^{\text{Bu}}: a \neq \rho \quad (5.35k)$$

$$\sum_{k \in \mathcal{K}} b_{i,k} \leq cs_i \quad , \forall i \in \mathcal{L} \subset \mathcal{V}^{\text{Br}} \quad (5.35l)$$

$$\sum_{k \in \mathcal{K}} b_{i,k} \leq cs_i \quad , \forall i \in \mathcal{L} \subset \mathcal{V}^{\text{Bu}} \quad (5.35m)$$

$$f_{i,j,k} \in \{0, 1\} \quad , \forall (i,j) \in (\mathcal{E}^{\text{Br}} \cup \mathcal{E}^{\text{Bu}}); \forall k \in \mathcal{K} \quad (5.35n)$$

$$b_{i,k} \in \{-1, 0, 1\} \quad , \forall i \in (\mathcal{V}^{\text{Br}} \cup \mathcal{V}^{\text{Bu}}); \forall k \in \mathcal{K} \quad (5.35o)$$

Die Restriktionen (5.35a) und (5.35b) stellen den Flusserhalt in beiden Netzwerken sicher. Hierbei wird über den Parameter $\zeta_{i,k}$ berücksichtigt, dass bspw. Gitterboxen nur auf Wegen im Sendungslager und Flugzeugladeeinheiten nur auf Wegen im ULD-Lager bewegt werden können. Über die Ungleichung (5.35c) wird der maximal mögliche Fluss über eine Kante begrenzt. Der maximale Fluss, der sich aus technischen Restriktionen des realen Transportsystems ergibt, wird jeweils zur Hälfte auf das Break- bzw. Build-Netzwerk aufgeteilt. Basierend auf der Entscheidung, an welcher Arbeitsstation welcher Auftrag ausgeführt wird, ergeben sich verschiedene Bedarfe an den entsprechenden Maschinen. Mit den Nebenbedingungen (5.35d) und (5.35e) werden der Bedarf bzw. das Angebot der Abbauaufträge modelliert. Mit Restriktion (5.35f) wird entschieden, zu welchen Lagerbereichen das Angebot der Abbauaufträge zugewiesen wird. Über den Parameter $\zeta_{i,k}$ wird sichergestellt, dass Gitterboxen im Sendungslager und Flugzeugladeeinheiten im ULD-Lager gelagert werden. Über die Nebenbedingung (5.35g) werden die Nachfrageknoten im Lager des Break-Netzes zu Angebotsknoten im Lager des Build-Netzes transferiert. Mit der Gleichung (5.35h) wird für jeden Auftrag die Ladeeinheit ermittelt, die von diesem Auftrag benötigt wird und den geringsten Zeitbedarf für die Bereitstellung besitzt. Der ermittelte Parameter r'_a wird auf die Bereitstellungszeit r_a des Auftrags aufgeschlagen.¹⁰⁵ Mit den Nebenbedingungen (5.35i) und (5.35j) wird analog zu den Abbauaufträgen der Bedarf bzw. das Angebot für die Aufbauaufträge modelliert. Mit Restriktion (5.35k) wird entschieden, zu welchen Lagerbereichen das Angebot der Aufbauaufträge zugewiesen wird. Über die Ungleichungen (5.35l) und (5.35m) wird die Lagerkapazität im Lagerbereich $i \in \mathcal{L}$ auf einen maximalen Wert begrenzt. Die maximale Lagerkapazität, die sich aus technischen Restriktionen des realen Transportsystems ergibt, wird jeweils zur Hälfte auf das Break- bzw. Build-Netzwerk aufgeteilt. Die binäre Eigenschaft der Flussvariablen wird über die Nebenbedingung (5.35n) sichergestellt. Schließlich wird über Restriktion (5.35o) festgelegt, dass entweder ein Bedarf ($b_{i,k} = -1$) bzw. ein Angebot ($b_{i,k} = +1$) einer Ladeeinheit $k \in \mathcal{K}$ an einem Knoten i vorliegt oder der Knoten lediglich als Transferknoten ($b_{i,k} = 0$) dient.

5.3.5.2 Heuristik für die Materialflusssteuerung

Im Folgenden wird die in Abschnitt 5.3.3 beschriebene evolutionäre Metaheuristik erweitert, um die Überlegungen zu dem Materialfluss und den Transportsystemen als weitere einschränkende Ressource aus dem vorherigen Abschnitt zu integrieren.

Basierend auf der Information, welchen Ausgangs- und Zielpunkt eine Ladeeinheit hat, ergibt sich der Transportweg dieser Einheit im Transportsystem (vgl. Abschnitt 2.2.2.4). Der Transportweg führt hierbei immer über die kürzeste Verbindung. Daher sind für das Transportsystem einmalig für alle Paare von Knoten $v \in \mathcal{V}$ und $v' \in \mathcal{V}$ mit $v \neq v'$ die kürzesten Pfade zu bestimmen und zu speichern.¹⁰⁶ Die Informationen über die kürzesten Pfade werden für die Evaluation eines Ablaufplans verwendet.

¹⁰⁵Dies betrifft die Restriktionen (5.25k), (5.30f) und (5.34f). Es soll sichergestellt werden, dass für den Auftrag genügend Zeit für die Bereitstellung der ersten Ladeeinheit eingeplant wird.

¹⁰⁶Dies kann bspw. mit Hilfe von *All-pairs shortest path*-Algorithmen, wie dem Floyd-Warshall-Algorithmus, erfolgen.

Die erste Änderung an der Metaheuristik betrifft Schritt 5 der Initialisierung des Algorithmus (vgl. Abschnitt 5.3.3.2). Die Maschine für den aktuell betrachteten Auftrag a wird nicht mehr auf Basis der vier beschriebenen Strategien aus der Menge \mathcal{M}_a bestimmt, sondern spezifisch ausgewählt, um somit Initiallösungen zu erzeugen, die bereits eine Reduktion der Förderdistanzen und damit der Transportzeiten der Fracht im Lager berücksichtigen. Hierbei ist zwischen Abbau- und Aufbauaufträgen zu unterscheiden.

Für erstere ist zum Zeitpunkt der Planung bekannt, an welcher Lagerposition im ULD-Lager sich die Flugzeugladeeinheit befindet,¹⁰⁷ die abgebaut werden muss. Daher wird für einen Abbauauftrag $a \in \mathcal{A}^{\text{Br}}$ mit einer hohen Wahrscheinlichkeit die Maschine $m^* \in \mathcal{M}_a$ ausgewählt, die die kürzeste Distanz zu der entsprechenden Lagerposition besitzt, um somit die Transportzeit zu minimieren. Es können jedoch auch weiter entfernte Maschinen $m \in \mathcal{M}_a \setminus \{m^*\}$ ausgewählt werden, wobei die Auswahlwahrscheinlichkeit mit der Entfernung abnimmt. Dies ist notwendig, wenn nicht alle Aufträge, die auf Maschine $m^* \in \mathcal{M}_a$ aufgrund der Distanz eingeplant würden, in der zur Verfügung stehenden Zeit durch die Transportsysteme transportiert werden können.

Für Aufbauaufträge hingegen sind zwei Fälle zu unterscheiden. Im ersten Fall ist zum Zeitpunkt der Planung bekannt, an welchen Lagerpositionen innerhalb des Sendungslagers sich die Gitterboxen mit der Fracht zum Aufbau befinden. Darüber hinaus ist die Position des leeren und dem Auftrag zugeordneten Ladehilfsmittels im ULD-Lager bekannt. Als die dem Aufbauauftrag $a \in \mathcal{A}^{\text{Bu}}$ zuzuordnende Maschine $m^* \in \mathcal{M}_a$ wird mit hoher Wahrscheinlichkeit diejenige Maschine bestimmt, an der die mittlere Entfernung zu allen Ladeeinheiten und -hilfsmitteln am geringsten ist. Analog zu den Abbauaufträgen können auch hier weiter entfernte Maschinen mit entsprechend geringerer Wahrscheinlichkeit ausgewählt werden. Im zweiten Fall ist lediglich die Position des leeren Ladehilfsmittels im ULD-Lager bekannt. Die darüber hinaus erforderlichen Frachtstücke zum Aufbau müssen zuvor durch einen oder mehrere Abbauaufträge verfügbar gemacht werden. Somit ist die Position der Gitterboxen mit der entsprechenden Fracht zum Zeitpunkt der Planung noch nicht bekannt, sondern abhängig von den einzuplanenden Abbauaufträgen und der Einlagerstrategie des Sendungslagers für die Gitterboxen. Hier bietet es sich an, dem Aufbauauftrag $a \in \mathcal{A}^{\text{Bu}}$ die Maschine $m^* \in \mathcal{M}_a$ mit hoher Wahrscheinlichkeit zuzuordnen, deren mittlerer Abstand zu den Maschinen der entsprechenden Abbauaufträge am geringsten ist. Weiter entfernte Maschinen können erneut mit abnehmender Wahrscheinlichkeit ausgewählt werden.

Eine weitere Änderung ergibt sich für die in Abschnitt 5.3.3.5 beschriebene zweite Mutationsart des Mutationsoperators. Die zufällige Auswahl einer Maschine m aus der Menge möglicher Maschinen \mathcal{M}_a für einen Auftrag a wird ergänzt um die Berücksichtigung der Transportentfernungen. Hierbei werden die gleichen Annahmen bzgl. der Auf- und Abbauaufträge wie zuvor bei Schritt 5 der Initialisierung zu Grunde gelegt. Weiter entfernte Maschinen werden ebenfalls mit geringerer Wahrscheinlichkeit ausgewählt als näher liegende.

¹⁰⁷Die Lagerpositionen der benötigten leeren Gitterboxen aus dem Sendungslager sind zu vernachlässigen, da diese durch eine geeignete Betriebsstrategie für das Lager gleichmäßig verteilt werden sollten. Dies ist ebenfalls nötig, da der genaue Bedarf an Gitterboxen für einen Abbauauftrag zum Zeitpunkt der Planung nur geschätzt werden kann.

Nach der Dekodierung eines Individuums η zu dem Ablaufplan ap_i^η können die Kanten der beiden Netzwerke \mathcal{G}^{Br} und \mathcal{G}^{Bu} mit den entsprechenden Flüssen $f_{i,j,k}$ auf Basis des Ausgangs- und Zielpunktes der Fracht und den bekannten kürzesten Wegen belegt werden. Für die Menge \mathcal{K}_a^+ wird angenommen, dass die abgebaute Flugzeugladeeinheit und die beladenen Gitterboxen im Falle der Abbauaufträge sowie die aufgebaute Flugzeugladeeinheit und die entladenen Gitterboxen im Falle der Aufbauaufträge jeweils in die Lagerbereiche \mathcal{L} transportiert werden, denen sie entnommen worden sind.

Nachdem die Flüsse auf den Kanten bestimmt sind, wird für jede Kante über Restriktion (5.35c) ermittelt, ob die entsprechenden Kapazitäten eingehalten werden. Die Summe der Kanten, an denen die Kapazitäten überschritten werden, werden als Strafkosten für das Individuum η berücksichtigt und beeinflussen die Auswahlwahrscheinlichkeit während der Selektionsphase. Die über Gleichung (5.35h) ermittelte Bereitstellungszeit der ersten Ladeeinheit wird auf die Bereitstellungszeit des Auftrags aufgeschlagen. Anschließend kann die Evaluation des Individuums η durchgeführt werden.

5.4 Ablaufsteuerung und Revision von Ablaufplänen

Mit der in den vorherigen Abschnitten beschriebenen proaktiven Ablaufplanung werden Ablaufpläne erstellt, die bekannte Schwankungen der Bearbeitungszeiten berücksichtigen und somit weniger anfällig gegenüber Störungen sind. In Abschnitt 2.2.3 wurde bereits aufgezeigt, dass während der Ausführung des Plans Ereignisse eintreten können, deren Auswirkungen zu groß sind, als dass sie durch den robusten Ablaufplan kompensiert werden könnten. Daher wird im Folgenden eine Ablaufsteuerung konzipiert, die beim Auftreten dieser Ereignisse Anpassungen am Ablaufplan vornehmen kann. Hierzu wird in Abschnitt 5.4.1 zunächst beschrieben, wie die Ausführungsüberwachung des aktiven Ablaufplans erfolgt und wie Ereignisse erkannt werden können. Anschließend wird in Abschnitt 5.4.2 dargelegt, wie die erkannten Ereignisse und ihre Auswirkungen den verschiedenen Auftragsmengen zugeordnet und wie entsprechende Verfahren zur Revision des Ablaufplans eingesetzt werden.

5.4.1 Ausführungsüberwachung des aktiven Ablaufplans

Der durch den Entscheidungsträger ausgewählte Ablaufplan¹⁰⁸ ap_i^p für den Zeitraum t_i bis zum nächsten Planungszeitpunkt t_{i+1} der rollierenden Ablaufplanung wird auf der reaktiven Ebene¹⁰⁹ gespeichert und durch diese sukzessiv abgearbeitet, indem Auftragsfreigaben an die entsprechenden Subsysteme gesendet werden, die für den Materialtransport zuständig sind. Des Weiteren werden die entsprechenden Mitarbeiter über die nächsten zu bearbeitenden Aufträge informiert. Der Planfortschritt wird über die Rückmeldungen der Subsysteme und der mobilen Datenerfassung an den Arbeitsstationen überwacht.¹¹⁰

¹⁰⁸Vergleiche Abschnitt 5.3.3.7.

¹⁰⁹Vergleiche Abschnitt 5.1.

¹¹⁰Die konkrete Ausgestaltung der Auftragsfreigaben und der Überwachung des Planfortschritts ist Teil der technischen Umsetzung eines konkreten Systems und wird hier nicht weiter behandelt.

Ereignisse wie der Planfortschritt werden über die Subsysteme, insbesondere die mobile Datenerfassung an den Arbeitsstationen, erkannt. Für Aufträge wird die Überschreitung der zugehörigen Bearbeitungszeitintervalle als Ereignis zum Zeitpunkt t_{ev} berücksichtigt. Hierfür wird für jeden gestarteten Auftrag $a \in \mathcal{A}^R$ regelmäßig überprüft, ob für das Restbearbeitungszeitintervall dieses Auftrags $I_a^{\text{Rest}} = [0, 0]$ bzw. für die aus der Häufigkeitsverteilung \mathcal{H}_{c^*} ermittelte Restbearbeitungszeit $\tau_a^{\text{Rest}} = 0$ gilt.¹¹¹ Ist dies der Fall, liegt eine Primärstörung für diesen Auftrag vor, da eine Bearbeitungszeit außerhalb des Intervalls bei der initialen proaktiven Ablaufplanung nicht berücksichtigt worden war. Aufgrund der Abhängigkeiten zwischen den Aufträgen können Sekundärstörungen bei nachgelagerten Aufträgen auftreten. Hierbei bestehen Abhängigkeiten zu

- Aufträgen, die auf derselben Maschine wie der verspätete Auftrag eingeplant sind,
- Aufträgen, die aufgrund von Vorrangbeziehungen erst nach dem verspäteten Auftrag ausgeführt werden können,
- Aufträgen, die durch dieselben Mitarbeiter wie der verspätete Auftrag bearbeitet werden.

Die Sekundärstörungen können dabei auf Basis der Abhängigkeiten sowohl bei den direkten als auch bei den indirekten Nachfolgeaufträgen durch die Ausbreitung von Störungen auftreten.

Weitere Ereignisse, die bei der Ausführung des Ablaufplans auftreten und durch die Ausführungsüberwachung berücksichtigt werden müssen, sind Maschinen-, Transportsystem- und Mitarbeiterausfälle sowie die Ankunft von Eilaufträgen und der Abbruch bereits eingeplanter oder fixierter Aufträge. Die zuvor erläuterten Abhängigkeiten sind auch hier zu berücksichtigen. Die Ereignisse können je nach vorhandenen Systemen und Automatisierungsgrad automatisiert erkannt oder manuell der Planungsarchitektur zur Verfügung gestellt werden.

5.4.2 Ereignisorientierte Revision des aktiven Ablaufplans

Wurde durch die Ausführungsüberwachung ein Ereignis festgestellt, ist auf dieses durch geeignete Reparaturstrategien bzw. Umplanungsverfahren zu reagieren. Hierbei ist zwischen den verschiedenen Auftragsmengen zu unterscheiden, da der aktuelle Ausführungsstatus eines Auftrags darüber entscheidet, welche Reparaturverfahren einsetzbar sind. Die konkrete Ausgestaltung dieser Verfahren ist aufgrund der Systemnähe der Ablaufsteuerung von dem konkreten System abhängig und entsprechend zu adaptieren. Daher sind die im Folgenden beschriebenen Vorgehensweisen exemplarisch zu verstehen.

Aufträge der Menge $\mathcal{A}_{t_{ev}}^R$, die zum Zeitpunkt t_{ev} des Ereignisses ev bereits gestartet sind, können ausschließlich durch eine Primärstörung betroffen sein. Die Möglichkeiten für eine Revision innerhalb dieser Auftragsmenge sind begrenzt und umfassen lediglich den Abbruch und die Verschiebung sowie die zeitliche Aussetzung der weiteren Bearbeitung des betroffenen Auftrags. Der Abbruch und die anschließende Verschiebung des Auftrags

¹¹¹Vergleiche Abschnitt 5.3.2.1.

ist im Falle des längerfristigen Ausfalls von Maschinen oder Transportsystemen gegeben, sofern die Möglichkeit besteht, die Fracht von der betroffenen Arbeitsstation auf eine andere Station zu verlegen und dort die Bearbeitung fortzusetzen. Die Aussetzung der Bearbeitung bietet sich bei absehbaren Störungen an. Bei der Überschreitung des für einen Auftrag ermittelten Bearbeitungszeitintervalls sollte von den beiden zuvor genannten Maßnahmen abgesehen und der Auftrag zu Ende bearbeitet werden, da eine Verschiebung nur zu weiteren Verzögerungen führen würde.¹¹²

In jedem Fall bedarf die Störungsdauer einer Quantifizierung als Ausgangsbasis für die Reparaturstrategien, die zur Anwendung bei den anderen Auftragsmengen kommen. Ist dies nicht sofort durchführbar, so wird empfohlen, mit der Anwendung von Umplanungsverfahren abzuwarten, bis eine Quantifizierung möglich bzw. bis die Störung behoben worden ist.¹¹³ Diese Forderung kann im Folgenden abgeschwächt werden, indem die Propagation der Primärstörung vermindert wird. Für alle Aufträge, die dann nicht mehr unmittelbar von der Primärstörung betroffen sind, kann ein Umplanungsverfahren angewendet werden, um den Ablaufplan den Änderungen anzupassen.

Zur Reduzierung der Propagation der Primärstörung werden die bestehenden Abhängigkeiten zu anderen Aufträgen minimiert. Die Vorbereitungen für Aufträge, die auf derselben Maschine wie der von der Störung betroffene Auftrag eingeplant sind und der Menge $\mathcal{A}_{t_{ev}}^F$ angehören, werden – sofern zeitlich möglich – rückgängig gemacht. Anschließend werden diese Aufträge sowie Aufträge der Menge $\mathcal{A}_{t_{ev}}^P$ für eine erneute Ablaufplanung freigegeben. Für Aufträge, zu denen Vorrangbeziehungen¹¹⁴ bestehen und die der Menge $\mathcal{A}_{t_{ev}}^F$ oder $\mathcal{A}_{t_{ev}}^P$ angehören, werden die Einplanungen auf die Maschinen rückgängig gemacht. Die Aufträge werden markiert und bis zur Auflösung der Störung in der Ablaufplanung nicht mehr berücksichtigt.¹¹⁵ Die Abhängigkeiten aufgrund der Mitarbeiter-einplanungen werden aufgelöst, indem alle weiteren Zuordnungen der Mitarbeiter, die für den von der Störung betroffenen Auftrag eingeplant sind, entfernt werden. Diese Mitarbeiter werden bis zur Auflösung der Störung in der Personaleinsatzplanung nicht mehr berücksichtigt, da sie zur Fertigstellung des betroffenen Auftrags nach Auflösung der Störung benötigt werden.

Nach Durchführung der zuvor genannten Anpassungen kann die Ablaufplanung und Personaleinsatzplanung erneut durchgeführt werden. Für die Ablaufplanung können prinzipiell partielle Umplanungsverfahren, wie sie in Abschnitt 3.3.1 dargestellt

¹¹²Hierbei handelt es sich um eine rein reaktive Vorgehensweise. Wird während der Bearbeitung des Auftrags fortlaufend der aufgrund der aktuellen Bearbeitungsgeschwindigkeit und der noch ausstehenden Fracht wahrscheinlich zu erzielende Fertigstellungszeitpunkt ermittelt, so wäre eine antizipative Reaktion möglich. Diese könnte z. B. in der Beschleunigung der Bearbeitung durch Hinzunahme weiterer Mitarbeiter liegen. Auf diese Möglichkeit, die einer Erweiterung der reaktiven Ebene entspräche, wird hier nicht näher eingegangen.

¹¹³Vgl. [QBY06], S. 168.

¹¹⁴Dies können hier lediglich Aufbauaufträge sein, die selbst keine weiteren Vorrangbeziehungen zu anderen Aufträgen besitzen.

¹¹⁵Eine alternative Vorgehensweise zur Auflösung einer Vorrangbeziehung zwischen einem Abbau- und einem Aufbauauftrag besteht in der Änderung des Aufbauauftrags, sodass dieser keine Fracht aus dem Abbauauftrag mehr enthält. Hierfür ist allerdings eine Rückkopplung in das für die Auftragserstellung verantwortliche System notwendig, das hier nicht näher betrachtet wird.

worden sind, verwendet werden. Hierbei sollte in jedem Fall das in Abschnitt 5.3.2.2 beschriebene Stabilitätsmaß als Entscheidungsgröße mit berücksichtigt werden. Alternativ kann die hier vorgestellte proaktive Ablaufplanung erneut aufgerufen werden. Für die Personaleinsatzplanung kann die in Abschnitt 5.3.4.2 beschriebene Heuristik verwendet werden. Sobald eine Quantifizierung der Störungsdauer der Primärstörung möglich oder diese behoben bzw. beendet worden ist, können die zuvor entfernten Aufträge und Mitarbeiter erneut in die Ablaufplanung bzw. Personaleinsatzplanung einbezogen werden.

Aufträge der Menge $\mathcal{A}_{t_{ev}}^F$, die zum Zeitpunkt t_{ev} des Ereignisses ev bereits fixiert worden sind, können – wie zuvor beschrieben – durch Sekundärstörungen betroffen sein. Primärstörungen treten hier ausschließlich durch den Abbruch des fixierten Auftrags auf. Die Möglichkeiten für eine Revision innerhalb dieser Auftragsmenge umfassen, sofern bereits durchgeführte Vorbereitungen rückgängig gemacht werden können, alle zuvor genannten Vorgehen und zusätzlich die Revision durch Umplanungsverfahren. Die Vorbereitungen sollten allerdings nur dann rückgängig gemacht werden, wenn es sich um längerfristige Störungen handelt oder nur ein geringer Prozentsatz der Vorbereitungen abgeschlossen ist. Das zuvor beschriebene Vorgehen zur Reduktion der Propagation der Störung gilt hier analog.

Für Aufträge der Menge $\mathcal{A}_{t_{ev}}^P$ gilt grundsätzlich das zuvor für fixierte Aufträge Geschriebene. Bei ausschließlich eingeplanten Aufträgen besteht jedoch eine größere Flexibilität, da noch keine Arbeitsvorbereitungen – wie die Vorpufferung von Fracht – stattgefunden haben. Daher sollte für diese Aufträge die bestehende Einplanung auf eine Arbeitsstation aufgehoben und der Auftrag erneut für die Ablaufplanung freigegeben werden.

6 Validierung

In diesem Kapitel erfolgt die Validierung¹ der in den vorangegangenen Abschnitten entwickelten Konzepte für ein Ablaufplanungs- und -steuerungssystem für Frachtabfertigungsprozesse in Luftfrachtterminals auf Basis synthetischer sowie realer² Testinstanzen.³ Ziel ist es, die vorgestellten Verfahren hinsichtlich ihrer Effektivität und Effizienz bezüglich der in Kapitel 2 dargelegten Problemstellung und Anforderungen zu bewerten.

Der Aufbau des Kapitels orientiert sich hierbei an den in den Abschnitten 2.2.1 und 2.2.2 identifizierten Problemfeldern und den für deren Lösung in Kapitel 5 vorgestellten Verfahren. In den einzelnen Abschnitten wird jeweils zuerst die Erstellung der Testinstanzen und der Versuchsaufbau der einzelnen Experimente beschrieben, woran sich die Darstellung und Diskussion der Ergebnisse im Hinblick auf die Problemstellung und die gestellten Anforderungen anschließt. In Abschnitt 6.1 wird zunächst die Validierung des Clustering- und Klassifikationsverfahrens für historische Betriebsdaten beschrieben. Schließlich erfolgt in Abschnitt 6.2 die Auswertung der proaktiven Ablaufplanung.

6.1 Validierung der Erstellung des benötigten Informationsstands

In diesem Abschnitt erfolgt die Validierung für die beiden in den Abschnitten 5.2.1 und 5.2.2 beschriebenen Verfahren zur Clusterbildung bzw. Klassifikatorentwicklung für historische bzw. unbekannte Aufträge. Hierzu werden zunächst die verwendeten Testinstanzen beschrieben und anschließend werden der Versuchsaufbau sowie die Auswertung der Ergebnisse für jedes Experiment des entsprechenden Moduls erläutert.

6.1.1 Testinstanzen für die Validierung des Clustering- und Klassifikationsverfahrens

Für die Validierung des Clustering- und Klassifikationsverfahrens werden sieben Testinstanzen verwendet. Die ersten vier Instanzen (R1–R4) sind Realdaten zu historischen

¹Die Validierung erfolgt auf zwei Intel Xeon E5620 CPUs mit 2.40 GHz, 48 GB RAM und Windows Server 2008 R2 64 Bit. Die Implementierung erfolgte in C# unter .NET Framework 4.5 und wurde als 64-Bit-Anwendung kompiliert.

²Bei den hier verwendeten realen Testinstanzen handelt es sich um Echtdateien aus einem Luftfrachtterminal, die im Rahmen dieser Arbeit anonymisiert wurden.

³Auf eine Validierung der rein konzeptionellen Beschreibung der Ablaufsteuerung und die Revision von Ablaufplänen aus Abschnitt 5.4 wird hier aufgrund der Abhängigkeit dieser Verfahren von einem konkreten System verzichtet.

Tabelle 6.1: Anzahl der historischen Aufträge in den realen und synthetischen Testinstanzen unterteilt nach den nicht (quasi)stetigen Eigenschaften

Instanz	Anz. historische Aufträge					
	$e_1 = 0$		$e_1 = 1$		$e_1 = 2$	
	$e_2 = 0$	$e_2 = 1$	$e_2 = 0$	$e_2 = 1$	$e_2 = 0$	$e_2 = 1$
R1	2271	2870	2077	2782	0	0
R2	1771	2588	2556	3085	0	0
R3	2416	2925	1982	2677	0	0
R4	2435	2928	2123	2514	0	0
S1	2000	2000	2000	2000	1000	1000
S2	100	100	100	100	50	50
S3	650	550	230	220	300	50

Aufträgen, die in einem großen Luftfrachtterminal in verschiedenen Monaten erfasst worden sind. Des Weiteren wurden drei synthetische Testinstanzen (S1–S3) generiert, die jeweils zehn Cluster mit Aufträgen enthalten. Für die Aufträge eines Clusters wurden verschiedene Verteilungen für die auftragsbeschreibenden Eigenschaften sowie für die realisierten Bearbeitungszeiten definiert.

Die Aufteilung in Auf- ($e_1 = 0$), Abbau- ($e_1 = 1$) und Transferaufträge ($e_1 = 2$) sowie eine weitere Unterteilung nach dem verwendeten Ladehilfsmittel ($e_2 = 0$ für Luftfrachtcontainer und $e_2 = 1$ für -paletten) ist Tabelle 6.1 zu entnehmen. In den realen Testinstanzen traten zwölf verschiedene Typen von Luftfrachtcontainern und sechs Typen von -paletten auf. Diese reichten von kleinen AKH-Containern mit $3,4 \text{ m}^3$ bis zu PMC-Paletten mit bis zu $21,2 \text{ m}^3$ Ladevolumen.

Weitere statistische Informationen zu den realen Testinstanzen sind in Tabelle 6.2 aufgeführt. Aufgegliedert nach Testinstanzen und Typ des Auftrags sind jeweils das arithmetische Mittel sowie die Populations-Standardabweichung der Bearbeitungszeitabweichung $\Delta\tau$ sowie der (quasi)stetigen Eigenschaften p mit $p \in \{3, \dots, 9\}$ der historischen Aufträge dargestellt.

Bei dem in Abschnitt 6.1.2 zu evaluierenden Clusteringverfahren handelt es sich um ein unüberwachtes Lernverfahren zur Suche nach neuen Clustern in den Daten über historische Aufträge. Aufgrund der unbekanntenen Clusterzugehörigkeit der historischen Aufträge in den realen Testinstanzen werden zur Clustervalidierung und somit zur Bemessung der Qualität des Clusteringergebnisses sogenannte interne Maße⁴ verwendet. Die synthetischen Testinstanzen wurden im Hinblick auf die Evaluation des Clusteringverfahrens konstruiert, da es bei diesen möglich ist, im Vorhinein Cluster zu konstruieren, die dann durch das Clusteringverfahren ermittelt werden sollen. Für die gefundenen Cluster der synthetischen Testinstanzen ist eine Clustervalidierung sowohl auf Basis interner

⁴Bei internen Maßen zur Clustervalidierung werden Kriterien und Eigenschaften verwendet, die aus den Daten selbst gewonnen werden (z. B. Intra- und Inter-Cluster-Distanzen) (vgl. [HBV01], S. 128 f.; [LLX⁺10]; [RAAQ11], S. 28 f.; [ZM14], Kapitel 17).

Tabelle 6.2: Arithmetisches Mittel und Populations-Standardabweichung der Bearbeitungszeitabweichung sowie der (quasi)stetigen Eigenschaften für die realen Testinstanzen

Inst.	Typ	$\Delta\tau$ / %	e_3 / m ³	e_4 / m ³	e_5 / Anz.	e_6 / %	e_7 / %	e_8 / %	e_9 / Anz.
R1	BD ¹	6,4 ±	8,4 ±	6,5 ±	23 ±	54,5 ±	10,2 ±	5,0 ±	4 ±
		11,0	3,7	3,0	49	38,7	10,1	8,1	5
	BU ²	16,8 ±	8,4 ±	7,0 ±	18 ±	58,6 ±	49,8 ±	29,3 ±	3 ±
		21,3	3,6	3,1	36	41,0	22,5	29,6	3
R2	BD ¹	8,1 ±	8,7 ±	7,9 ±	24 ±	41,8 ±	15,9 ±	7,5 ±	4 ±
		10,3	3,7	3,4	47	40,0	13,0	10,6	5
	BU ²	15,3 ±	8,2 ±	6,4 ±	18 ±	56,3 ±	35,0 ±	18,3 ±	3 ±
		17,3	3,7	3,0	32	41,4	18,5	20,8	3
R3	BD ¹	5,9 ±	8,3 ±	7,0 ±	22 ±	46,1 ±	27,1 ±	12,9 ±	3 ±
		8,6	3,7	3,2	40	38,8	17,4	16,2	3
	BU ²	12,8 ±	8,4 ±	7,6 ±	17 ±	57,7 ±	28,5 ±	15,1 ±	4 ±
		14,8	3,7	3,4	27	41,1	15,9	17,0	3
R4	BD ¹	5,7 ±	8,3 ±	6,1 ±	22 ±	61,6 ±	20,3 ±	10,5 ±	3 ±
		6,3	3,7	2,8	39	38,8	14,6	13,8	3
	BU ²	12,4 ±	8,2 ±	7,8 ±	17 ±	56,8 ±	25,8 ±	13,6 ±	4 ±
		12,0	3,7	3,5	28	41,0	14,5	15,7	3

Legende:¹ Break-down = Abbauauftrag² Build-up = Aufbauauftrag

als auch externer Maße⁵ möglich, da die Clusterzugehörigkeit bekannt ist. Hierdurch ist ein Vergleich beider Maße und somit eine Beurteilung der Qualität des internen Maßes möglich.

6.1.2 Analyse des Clusteringverfahrens

Für die Validierung des Clusteringverfahrens aus Abschnitt 5.2.1 werden im Folgenden zwei Experimente durchgeführt. Das erste dient der Bestimmung, welche der drei beschriebenen Initialisierungsmethoden zur initialen Bestimmung der Clusterzentren (hier: Medoiden) verwendet werden sollte. Mit dem zweiten Experiment wird der Parameter k bestimmt, mit dem die Anzahl der zu bildenden Cluster festgelegt werden kann.

⁵Bei externen Maßen zur Clustervalidierung werden Kriterien und Eigenschaften verwendet, die nicht Teil der Daten sind, sondern bspw. in Form von vorherigem Wissen (z. B. Klassenlabel) bekannt sind (vgl. [HBV01], S. 124–128; [RAAQ11], S. 29 f.; [ZM14], Kapitel 17).

6.1.2.1 Auswahl der Initialisierungsmethode

Mit dem folgenden Experiment soll die geeignetste Initialisierungsmethode zur Bestimmung der initialen Clusterzentren ermittelt werden. Zur Auswahl stehen die drei in Abschnitt 5.2.1.3 beschriebenen Verfahren: *Random*, *Most-Middle*, *Farthest-First*. Diese werden für jede der sieben Testinstanzen bei einer festgelegten Clusteranzahl ($k = 10$ für die synthetischen und $k = 8$ für die realen Testinstanzen)⁶ verwendet und der Clusteringalgorithmus wird bis zum Abbruchkriterium bzw. bis zu maximal 50 Iterationen ausgeführt. Für die Initialisierungsmethode *Random* werden 50 unabhängige Durchläufe durchgeführt und der Mittelwert gebildet. Für die Normalisierungsfaktoren der Distanzfunktion wird die in Gleichung (5.1) auf Seite 86 beschriebene Menge \mathcal{NF} und als Gewichtungsfaktor $w_p = 1/9$ mit $1 \leq p \leq 9$ für \mathcal{W} (siehe Gleichung (5.2) auf Seite 86) verwendet.

Als internes Maß zur Clustervalidierung wird der Silhouettenkoeffizient nach Rousseeuw⁷ verwendet. Die Silhouette ist ein Indikator dafür, wie gut ein Objekt geclustert worden ist, indem die durchschnittliche Distanz zu den anderen Objekten des eigenen Clusters mit der durchschnittlichen Distanz zu den Objekten des nächstgelegenen Clusters verglichen wird. Die durchschnittliche Distanz eines historischen Auftrags ha_j zu den anderen Aufträgen des eigenen Clusters \mathcal{C}_{ha_j} wird folgendermaßen berechnet:

$$\text{dist}(\mathcal{C}_{ha_j}, ha_j) = \frac{1}{|\mathcal{C}_{ha_j}|} \sum_{ha \in \mathcal{C}_{ha_j}} \text{dist}(ha_j, ha)$$

Die Distanz zum nächstgelegenen Cluster \mathcal{C}' sei über das Minimum der durchschnittlichen Distanzen zu den Aufträgen in den anderen Clustern definiert:

$$\text{dist}(\mathcal{C}', ha_j) = \min_{\substack{\mathcal{C}_i \in \mathcal{C}: \\ \mathcal{C}_i \neq \mathcal{C}_{ha_j}}} \left(\frac{1}{|\mathcal{C}_i|} \sum_{ha \in \mathcal{C}_i} \text{dist}(ha_j, ha) \right)$$

Der Silhouettenkoeffizient eines Auftrags ha_j ist dann folgendermaßen definiert:

$$\text{sc}(ha_j) = \begin{cases} 1, & \text{wenn } \text{dist}(\mathcal{C}_{ha_j}, ha_j) = 0 \\ \frac{\text{dist}(\mathcal{C}', ha_j) - \text{dist}(\mathcal{C}_{ha_j}, ha_j)}{\max\{\text{dist}(\mathcal{C}_{ha_j}, ha_j), \text{dist}(\mathcal{C}', ha_j)\}} & \text{in allen anderen Fällen} \end{cases}$$

Der Silhouettenkoeffizient ist auf das Intervall $[-1, +1]$ begrenzt. Dabei gibt ein Wert von $+1$ an, dass der historische Auftrag nah an den anderen Aufträgen des eigenen Clusters liegt und weit entfernt von anderen Clustern ist. Ein Wert von -1 kennzeichnet einen historischen Auftrag, der näher an einem anderen als dem eigenen Cluster liegt. Für den Fall, dass der Koeffizient 0 ist, kann keine Aussage darüber getroffen werden,

⁶Es werden 10 bzw. 8 Cluster verwendet, da durch die Kombination aus der binären Eigenschaft 2 und der nominalen Eigenschaft 1 bereits von 6 Clustern im Falle der synthetischen und 4 Clustern (hier sind keine Transferaufträge aufgetreten, vgl. Tabelle 6.1) im Fall der realen Testinstanzen ausgegangen werden kann. Die restlichen vier Cluster dienen zum Auffinden unbekannter Gruppen.

⁷Siehe [Rou87].

ob die Zuordnung zum aktuellen oder zum nächstgelegenen Cluster besser ist, da der Auftrag an der Grenze zwischen den beiden Clustern liegt.⁸ Für eine Testinstanz i kann der Silhouettenkoeffizient schließlich als arithmetisches Mittel der Silhouetten aller historischen Aufträge \mathcal{A}_i^H dieser Instanz berechnet werden:

$$\text{sc}(\mathcal{A}_i^H) = \frac{1}{|\mathcal{A}_i^H|} \sum_{ha \in \mathcal{A}_i^H} \text{sc}(ha) \quad (6.1)$$

Ein Wert nahe +1 entspricht hierbei einer guten Aufteilung der Aufträge über die verschiedenen Cluster und somit einem Clusteringergebnis mit einer starken Strukturierung.⁹

Als externe Maße zur Clustervalidierung werden das Rand-Maß und der Jaccard-Koeffizient verwendet.¹⁰ Diese Maße geben an, wie gut das erzielte Clusteringergebnis des zu testenden Verfahrens im Vergleich zu einem a priori bekannten Clustering ist. Die tatsächliche Clusterzugehörigkeit wird somit als externe Information verwendet.¹¹ Zur Berechnung der Maße bezeichne $ha_j, ha_l \in \mathcal{A}_i^H$ mit $j \neq l$ zwei unterschiedliche historische Aufträge einer Testinstanz i sowie $y_j, y_l \in \{1, \dots, r\}$ die a priori bekannte und $\hat{y}_j, \hat{y}_l \in \{1, \dots, k\}$ die durch den Clusteralgorithmus ermittelte Clusterzugehörigkeit. Im Folgenden wird von $r = k$ ausgegangen. Der Fall $\hat{y}_j = \hat{y}_l$ wird als positives und $\hat{y}_j \neq \hat{y}_l$ als negatives Ereignis gewertet. Unter Berücksichtigung der A-priori-Informationen ergeben sich vier Fälle:¹²

- Richtig positiv: Die beiden Aufträge ha_j und ha_l gehören demselben Cluster an und wurden durch das Verfahren auch demselben Cluster zugeordnet.

$$TP = |\{(ha_j, ha_l) \mid y_j = y_l \wedge \hat{y}_j = \hat{y}_l\}|$$

- Falsch negativ: Die beiden Aufträge ha_j und ha_l gehören demselben Cluster an, wurden durch das Verfahren jedoch unterschiedlichen Clustern zugeordnet.

$$FN = |\{(ha_j, ha_l) \mid y_j = y_l \wedge \hat{y}_j \neq \hat{y}_l\}|$$

- Falsch positiv: Die beiden Aufträge ha_j und ha_l gehören nicht demselben Cluster an, wurden durch das Verfahren jedoch demselben Cluster zugeordnet.

$$FP = |\{(ha_j, ha_l) \mid y_j \neq y_l \wedge \hat{y}_j = \hat{y}_l\}|$$

- Richtig negativ: Die beiden Aufträge ha_j und ha_l gehören nicht demselben Cluster an und wurden durch das Verfahren auch nicht demselben Cluster zugeordnet.

$$TN = |\{(ha_j, ha_l) \mid y_j \neq y_l \wedge \hat{y}_j \neq \hat{y}_l\}|$$

⁸Siehe [Rou87], S. 56.

⁹Vgl. [ZM14], S. 445.

¹⁰Siehe [Ran71] für das Rand-Maß und [Jac12] für den Jaccard-Koeffizient.

¹¹Vgl. [ZM14], S. 425 f.

¹²Siehe ebd., S. 433 f.

Tabelle 6.3: Silhouettenkoeffizient, Rand-Maß und Jaccard-Koeffizient der verschiedenen Testinstanzen in Abhängigkeit der gewählten Initialisierungsmethode

Instanz	Initialisierungsmethode								
	Random			Most-Middle			Farthest-First		
	sc	rand	jacc	sc	rand	jacc	sc	rand	jacc
R1	0,42	—	—	0,43	—	—	0,43	—	—
R2	0,41	—	—	0,35	—	—	0,44	—	—
R3	0,37	—	—	0,32	—	—	0,39	—	—
R4	0,37	—	—	0,33	—	—	0,34	—	—
S1	0,34	0,93	0,53	0,25	0,88	0,37	0,51	0,97	0,78
S2	0,35	0,93	0,56	0,39	0,95	0,60	0,50	0,98	0,81
S3	0,30	0,90	0,50	0,25	0,87	0,32	0,48	0,98	0,88

Das Rand-Maß kann nun als Verhältnis der richtig positiven und negativen zu allen Auftragspaaren wie folgt definiert werden:

$$\text{rand}(\mathcal{A}_i^H) = \frac{TP + TN}{TP + FP + FN + TN}$$

Das Rand-Maß ist auf das Intervall $[0, 1]$ beschränkt. Ein Wert nahe +1 entspricht hierbei einem Clusteringergebnis, das sehr nahe an der a priori bekannten Clusterzuordnung liegt.¹³ Der Jaccard-Koeffizient ist definiert als das Verhältnis der richtig positiven zu allen Auftragspaaren, ohne Berücksichtigung der richtig negativen Auftragspaare:

$$\text{jacc}(\mathcal{A}_i^H) = \frac{TP}{TP + FP + FN}$$

Mit diesem Maß kann der für das Clustering wichtigere Wert TP besser berücksichtigt werden. Der Jaccard-Koeffizient ist auf das Intervall $[0, 1]$ beschränkt. Ein Wert nahe +1 entspricht hierbei einem Clusteringergebnis, das sehr nahe an der a priori bekannten Clusterzuordnung liegt.¹⁴

In Tabelle 6.3 sind die Ergebnisse für die verschiedenen Initialisierungsmethoden aufgeführt. Die zuvor beschriebenen externen Maße, Rand-Maß sowie Jaccard-Koeffizient, werden für die synthetischen und das interne Maß des Silhouettenkoeffizienten für sowohl die synthetischen als auch die realen Testinstanzen angegeben. Es zeigt sich, dass die Initialisierungsmethode Farthest-First in sechs der sieben Testinstanzen die besten Ergebnisse für den Silhouettenkoeffizienten liefert. Die für diesen Koeffizienten ermittelten guten Ergebnisse werden durch die externen Maße bei den synthetischen Testinstanzen bestätigt. Aus diesem Grund wird die Initialisierungsmethode Farthest-First im Folgenden bei der Parametrisierung der Anzahl Cluster verwendet.

¹³Siehe [ZM14], S. 435.

¹⁴Siehe ebd.

6.1.2.2 Parametrisierung der Anzahl Cluster

Nachdem im vorherigen Abschnitt die beste Initialisierungsmethode ermittelt worden ist, wird auf Basis dieser im Folgenden bestimmt, wie viele Cluster jeweils für die sieben Testinstanzen gebildet werden sollten. Hierzu wird ein Parametrisierungsexperiment für den Parameter k des Clusteringalgorithmus durchgeführt. Der Parameter wird dabei im Bereich von 3 bis 30 in Einzelschritten variiert und der Clusteringalgorithmus bis zum Abbruchkriterium bzw. bis zu maximal 50 Iterationen ausgeführt. Als Maß zur Bewertung des Clusteringergebnisses für jede Parametervariation wird erneut der in Gleichung (6.1) beschriebene Silhouettenkoeffizient verwendet, da sich dieser zur Bestimmung der Clusteranzahl eignet.¹⁵ Die Normalisierungs- und Gewichtungsfaktoren werden wie im vorherigen Experiment festgelegt.

Für die synthetischen Testinstanzen (S1–S3) ist die Anzahl der zu bildenden Cluster aufgrund der Konstruktion dieser Instanzen bereits bekannt. Erzielt der Silhouettenkoeffizient bei der jeweils bekannten Clusteranzahl für eine Testinstanz den höchsten Wert, so kann von einer Eignung des internen Maßes für die Ermittlung der Anzahl Cluster ausgegangen werden. Für die realen Testinstanzen kann somit auf die unbekannte Anzahl an Clustern geschlossen werden.

Die Ergebnisse des Parametrisierungsexperiments sind in Abbildung 6.1 dargestellt. Der Verlauf des Silhouettenkoeffizienten für die synthetischen Testinstanzen für die verschiedenen Ausprägungen des Parameters k ist in Abbildung 6.1(a) aufgezeigt. Analog zeigt Abbildung 6.1(b) die Ergebnisse für die realen Testinstanzen. Es zeigt sich, dass für die Instanzen S1–S3 der Silhouettenkoeffizient genau bei der bekannten Anzahl von jeweils zehn Clustern seinen höchsten Wert erzielt und danach wieder abfällt. Für die realen Testinstanzen lässt sich kein so eindeutiges Ergebnis erzielen. Auf Basis des Silhouettenkoeffizienten sollte jedoch für die Testinstanzen R1 und R4 von jeweils vier, für R2 von neun und für R3 von drei Clustern ausgegangen werden.

6.1.3 Analyse des Klassifikationsverfahrens

Die Validierung des in Abschnitt 5.2.2 beschriebenen Klassifikationsverfahrens erfolgt in diesem Abschnitt mittels zweier Experimente. Das erste dient der Quantifizierung der erreichten Klassifikationsgüten für die verschiedenen Testinstanzen anhand von Metriken und somit dem Nachweis der Effektivität des Verfahrens. Des Weiteren wird die Effizienz in Form der benötigten Zeit für eine Klassierung ermittelt. Das zweite Experiment dient zur Detaillierung der im ersten Experiment gewonnenen Ergebnisse, indem für eine ausgewählte Testinstanz die abstrakten Metriken näher untersucht werden.

6.1.3.1 Effektivität und Effizienz des Klassifikationsverfahrens

Im Folgenden wird die Effektivität und Effizienz des eingesetzten Klassifikationsverfahrens untersucht. Hierzu sind für jede reale und synthetische Testinstanz sowohl eine Trainingsmenge zur Erstellung des Klassifikators (vgl. Abschnitt 5.2.2) als auch eine hiervon

¹⁵Vgl. [ZM14], S. 448.

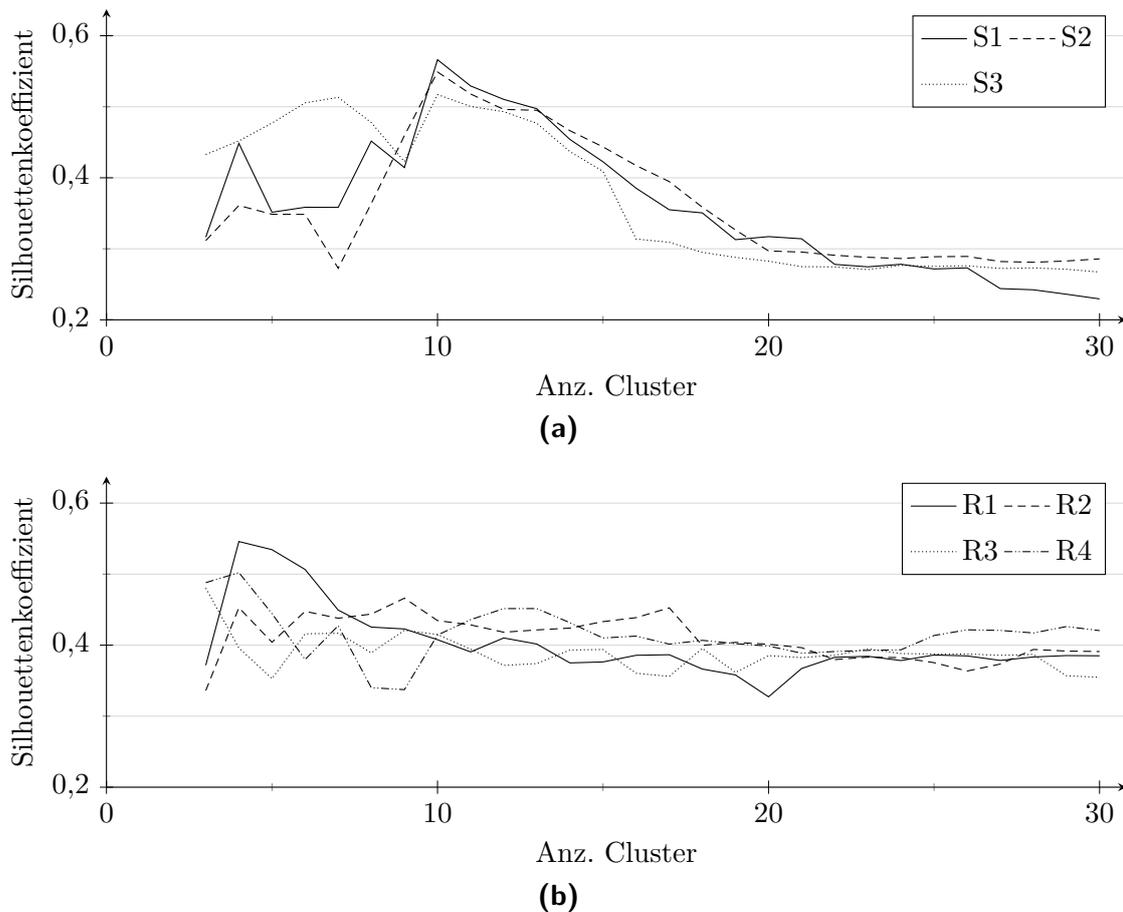


Abbildung 6.1: Verläufe des Silhouettenkoeffizienten für verschiedene Parametervariationen der Anzahl Cluster für die untersuchten Testinstanzen: (a) Synthetische Testinstanzen (b) Reale Testinstanzen

unabhängige Testmenge, die nicht zur Erstellung des Klassifikators verwendet wird, für die Berechnung der Metriken und Bestimmung der Effektivität notwendig.

Die Trainings- und Testmengen werden hier unter Verwendung einer zehnfachen stratifizierten Kreuzvalidierung gewonnen. Hierbei wird die Datenmenge \mathcal{A}_i^H historischer Aufträge einer Testinstanz i in zehn annähernd gleich große Teilmengen $\mathcal{A}_{i,j}^H$ mit $1 \leq j \leq 10$ unterteilt. Anschließend werden zehn Testdurchläufe durchgeführt, bei der jeweils die j -te Teilmenge $\mathcal{A}_{i,j}^H$ als Testmenge \mathcal{R}_j und die restlichen neun Teilmengen als Trainingsmenge \mathcal{T}_j verwendet werden. Die Stratifizierung dient dazu, dass sowohl die Test- als auch die Trainingsmenge eine repräsentative Anzahl an historischen Aufträgen aus jedem Cluster der Testinstanz beinhaltet. Aufgrund der zufälligen Auswahl der zehn Teilmengen können verschiedene Kreuzvalidierungsdurchläufe verschiedene Ergebnisse liefern. Aus diesem Grund wird das zuvor beschriebene Vorgehen der Kreuzvalidierung zehnmal wiederholt und das arithmetische Mittel der im Folgenden erläuterten Einzelergebnisse der Metriken gebildet.¹⁶

¹⁶Vgl. [WFH11], S. 152 ff.

Als Maß für die Effektivität werden die drei Metriken Erfolgsrate, quadratischer Verlust sowie Cohens Kappa verwendet. Die Erfolgsrate dient dabei der Quantifizierung, wie häufig das durch den Klassifikator klassierte Cluster¹⁷ für einen historischen Auftrag dem tatsächlichen Klassenlabel bzw. Cluster \mathcal{C}_{ha}^* dieses Auftrags entspricht. Die Metrik nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung ist folgendermaßen definiert:

$$\text{sr}(\mathcal{A}_i^{\text{H}}) = \frac{1}{|\mathcal{A}_i^{\text{H}}|} \sum_{j=1}^{10} \sum_{ha \in \mathcal{R}_j} \delta(\mathcal{C}_{ha}^* = \text{classify}(ha)),$$

mit der Indikatorfunktion

$$\delta(x) = \begin{cases} 1, & \text{wenn } x \text{ wahr ist} \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

Die Zugehörigkeitswahrscheinlichkeit wird bei dieser Metrik nicht berücksichtigt.

Mit Hilfe einer quadratischen Verlustfunktion kann die Zugehörigkeitswahrscheinlichkeit berücksichtigt und somit eine Aussage darüber getroffen werden, wie sicher die Entscheidungen bei den korrekt klassierten Aufträgen sind. Ein historischer Auftrag ha kann durch den Klassifikator einem der für die Testinstanz i ermittelten Cluster aus \mathcal{C}_i^* zugewiesen werden, wobei für jedes Cluster eine Zugehörigkeitswahrscheinlichkeit ermittelt wird. Diese seien durch den Vektor \vec{p}_{ha} gegeben, der für jedes Cluster $\mathcal{C}_j^* \in \mathcal{C}_i^*$ eine A-posteriori-Wahrscheinlichkeit¹⁸ $p_{ha,j}$ enthält, wobei die einzelnen Wahrscheinlichkeiten in Summe eins ergeben. Die tatsächliche Clusterzugehörigkeit kann in Form eines Zuordnungsvektors \vec{z}_{ha} angegeben werden, der für jedes Cluster $\mathcal{C}_j^* \in \mathcal{C}_i^*$ eine Zuordnung $z_{ha,j} \in \{0, 1\}$ enthält. Hierbei hat $z_{ha,j'}$, mit j' der tatsächlichen Clusterzugehörigkeit, den Wert 1 und alle anderen haben den Wert 0. Die quadratische Verlustfunktion nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung ist dann folgendermaßen definiert:¹⁹

$$\text{ql}(\mathcal{A}_i^{\text{H}}) = \sum_{l=1}^{10} \sum_{ha \in \mathcal{R}_l} \sum_{j=1}^{|\mathcal{C}_i^*|} (p_{ha,j} - z_{ha,j})^2$$

Die Funktionswerte sind auf das Intervall $[0, 2|\mathcal{A}_i^{\text{H}}|]$ beschränkt, wobei der kleinste Wert für den Fall erreicht wird, bei dem für die tatsächlichen Clusterzugehörigkeiten der Aufträge jeweils eine Zugehörigkeitswahrscheinlichkeit von 100 % ermittelt wird. Der maximale Wert wird unter der Bedingung erzielt, bei der für jeden Auftrag eine Zugehörigkeitswahrscheinlichkeit von 100 % für ein von der tatsächlichen Clusterzugehörigkeit verschiedenes Cluster ermittelt wird.

Als weiteres Maß wird Cohens Kappa²⁰ als ein statistisches Maß für die Interrater-Reliabilität der Klassierungen von zwei Ratern verwendet. Der erste Rater entspricht

¹⁷Siehe Gleichung (5.14) auf Seite 102.

¹⁸Siehe Gleichung (5.13) auf Seite 102.

¹⁹Vgl. [WFH11], S. 160 f.

²⁰[Coh60].

dabei den tatsächlichen und der andere den durch den Klassifikator ermittelten Clusterzugehörigkeiten der historischen Aufträge. Sei hierfür eine Kontingenztafel (engl. *confusion matrix*) mit den Ergebnissen des ersten Raters als Zeilen und des zweiten Raters als Spalten gegeben (vgl. Tabelle 6.5). Die Matrix besteht somit aus $|\mathcal{C}_i^*| \times |\mathcal{C}_i^*|$ Zellen. Der Inhalt einer Zelle entspricht den absoluten Häufigkeiten nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung und sei mit $h_{j,l}$ bezeichnet. Zur Berechnung von Cohens Kappa wird als erstes die gemessene Übereinstimmung beider Rater benötigt:²¹

$$p_0 = \frac{1}{|\mathcal{A}_i^H|} \sum_{j=1}^{|\mathcal{C}_i^*|} h_{j,j}$$

Als nächstes wird die erwartete zufällige Übereinstimmung beider Rater berechnet:

$$p_e = \frac{1}{|\mathcal{A}_i^H|^2} \sum_{j=1}^{|\mathcal{C}_i^*|} z_j \cdot s_j ,$$

wobei z_j der Zeilen- und s_j der Spaltensumme der Kontingenztafel entspricht. Cohens Kappa ist nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung schließlich folgendermaßen definiert:

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

Für $\kappa = 1$ gilt eine exakte Übereinstimmung der Zuordnungen beider Rater. Besteht keine Übereinstimmung, die über die erwartete zufällige Übereinstimmung hinausgeht, so ist $\kappa = 0$.

Zur Bestimmung der Effizienz des Verfahrens wird die durchschnittliche Zeit zur Klassierung eines Auftrags gemessen. Bei der späteren Ablaufplanung müssen die Aufträge, die neu in einen Ablaufplan eingeplant werden sollen, zunächst klassiert werden.²² Daher sollte dieser Vorgang effizient durchgeführt werden, um möglichst viel Zeit auf die eigentliche Ablaufplanung verwenden zu können.

Die Ergebnisse für die zuvor beschriebenen Metriken sind in Tabelle 6.4 für eine zehnfache stratifizierte Kreuzvalidierung mit zehnfacher Wiederholung aufgeführt. Es zeigt sich, dass bei den Klassierungen für alle Testinstanzen eine Erfolgsrate von über 90% erzielt wird. Die Werte für den quadratischen Verlust weisen zudem auf hohe Zugehörigkeitswahrscheinlichkeiten bei der Klassierung und somit auf sichere Zuordnungen hin. Die Ergebnisse für Cohens Kappa zeigen eine hohe Übereinstimmung der beiden Rater und somit ein deutlich besseres Ergebnis im Vergleich zu einer zufälligen Übereinstimmung. Die Laufzeit für eine Klassierung beträgt für jede Testinstanz jeweils unter einer Millisekunde. Für die Ablaufplanung sind diese Laufzeiten zu vernachlässigen, da sie nur zu Beginn einer Planung einmalig für jeden Auftrag der zu planenden Menge \mathcal{A} durchgeführt werden müssen. Die Metriken zeigen auf, dass der eingesetzte Klassifikator

²¹Vgl. [WFH11], S. 166 f.

²²Vergleiche hierzu Abschnitt 5.3.1.1 auf Seite 105.

Tabelle 6.4: Metriken zur Bestimmung der Effektivität und Effizienz des Klassifikationsverfahrens für die verschiedenen Testinstanzen

Instanz	Erfolgsrate in %	Quadratischer Verlust	Cohens Kappa	Laufzeit in ms
R1	97,24	472,02	0,96	0,05
R2	93,60	1018,75	0,93	0,10
R3	98,38	315,53	0,98	0,03
R4	98,25	340,18	0,98	0,04
S1	98,34	306,59	0,98	0,12
S2	98,54	12,74	0,98	0,14
S3	96,79	103,15	0,96	0,13

Aufträge zuverlässig den zuvor ermittelten Clustern zuordnen kann. Dies gilt hier auch für den Fall einer kleinen Testinstanz (vgl. Testinstanz S2) und somit wenigen historischen Aufträgen pro Cluster.

6.1.3.2 Ermittlung von Kontingenztafeln

Während im vorherigen Abschnitt die Effektivität auf Basis von Metriken, die für jede Testinstanz jeweils einen Ergebniswert ermitteln, bestimmt worden ist, wird in diesem Abschnitt die Testinstanz R2 näher untersucht, da für diese die im Vergleich zu anderen Testinstanzen schlechtesten Ergebnisse erzielt worden sind. Hierfür wird die für die Kappa-Metrik erzeugte Kontingenztafel, die aufzeigt, wie oft die Aufträge, die zu einem Cluster gehören, durch den Klassifikator diesem Cluster zugeordnet worden sind bzw. welchen anderen Clustern sie zugewiesen wurden, näher untersucht. Auf diese Weise lässt sich ermitteln, welche Cluster sich besonders ähnlich sind, weshalb Aufträge dieser Cluster besonders häufig verwechselt werden. Des Weiteren kann jede Zelle dieser Kontingenztafel durch eine Häufigkeitsverteilung angereichert werden, die aufzeigt, welche Zugehörigkeitswahrscheinlichkeit durch den Klassifikator bei der Klassierung zu dem entsprechenden Cluster ermittelt worden ist.

Die Ergebnisse für die neun Cluster C_1^* bis C_9^* der Testinstanz R2 nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung sind in Tabelle 6.5 aufgeführt. Es zeigt sich, dass der Großteil der Aufträge korrekt klassiert wurde (vgl. die Diagonale der Matrix). Einige der Aufträge wurden hingegen einem falschen Cluster zugeordnet, z. B. wurden 93 Aufträge, die zu Cluster C_7^* gehören, durch den Klassifikator dem Cluster C_1^* zugeordnet (vgl. Zelle $C_7^*-C_1^*$). Die hierbei ermittelten Zugehörigkeitswahrscheinlichkeiten sind exemplarisch für die Zellen $C_5^*-C_8^*$ und $C_7^*-C_1^*$ in Form einer Häufigkeitsverteilung in Tabelle 6.6 aufgeführt. Als Vergleich dienen die Häufigkeitsverteilungen der berechneten Zugehörigkeitswahrscheinlichkeiten der jeweils korrekt klassierten Aufträge und somit der Zellen $C_5^*-C_5^*$ sowie $C_7^*-C_7^*$. Es zeigt sich, dass bei einer korrekten Klassierung die Zugehörigkeitswahrscheinlichkeiten häufiger im Intervall $(0,9; 1,0]$ liegen (99,3% für $C_5^*-C_5^*$ und 87,9% für $C_7^*-C_7^*$), als dies bei den falschen Klassierungen der Fall ist (33,3% für

Tabelle 6.5: Kontingenztafel für Testinstanz R2 nach Durchführung einer zehnfachen stratifizierten Kreuzvalidierung

		Anz. Zuordnungen Rater 2								
		\mathcal{C}_1^*	\mathcal{C}_2^*	\mathcal{C}_3^*	\mathcal{C}_4^*	\mathcal{C}_5^*	\mathcal{C}_6^*	\mathcal{C}_7^*	\mathcal{C}_8^*	\mathcal{C}_9^*
Anz. Zuordnungen Rater 1	\mathcal{C}_1^*	1005	0	14	0	0	1	37	0	3
	\mathcal{C}_2^*	0	1362	0	24	0	1	1	49	0
	\mathcal{C}_3^*	0	0	1799	56	12	0	0	0	0
	\mathcal{C}_4^*	0	1	33	1165	0	0	33	0	0
	\mathcal{C}_5^*	0	4	13	0	403	0	0	60	0
	\mathcal{C}_6^*	5	2	0	0	0	910	72	4	18
	\mathcal{C}_7^*	93	0	0	0	0	11	1441	0	0
	\mathcal{C}_8^*	0	4	18	1	0	0	0	612	2
	\mathcal{C}_9^*	14	0	0	0	0	30	0	21	666

Tabelle 6.6: Durch den Klassifikator ermittelte Zugehörigkeitswahrscheinlichkeiten für vier ausgewählte Zellen der Kontingenztafel von Testinstanz R2

		Anteil Zuordnungen in %			
		$\mathcal{C}_5^*-\mathcal{C}_5^*$	$\mathcal{C}_5^*-\mathcal{C}_8^*$	$\mathcal{C}_7^*-\mathcal{C}_1^*$	$\mathcal{C}_7^*-\mathcal{C}_7^*$
Zugehörigkeits- wahrscheinlichkeit	$[0,0;0,1]$	0	0	0	0
	$(0,1;0,2]$	0	0	0	0
	$(0,2;0,3]$	0	0	0	0
	$(0,3;0,4]$	0	0	0	0
	$(0,4;0,5]$	0,5	11,7	0	0,1
	$(0,5;0,6]$	0,2	38,3	14,0	0,9
	$(0,6;0,7]$	0	8,3	40,9	1,2
	$(0,7;0,8]$	0	6,7	3,2	4,4
	$(0,8;0,9]$	0	1,7	2,1	5,5
	$(0,9;1,0]$	99,3	33,3	39,8	87,9

$\mathcal{C}_5^*-\mathcal{C}_8^*$ und 39,8% für $\mathcal{C}_7^*-\mathcal{C}_1^*$). Letztere weisen zudem eine größere Streuung über die Intervalle auf.

6.2 Validierung der multikriteriellen Ablaufplanung der Abfertigungsprozesse

Im Folgenden erfolgt die Validierung der in Abschnitt 5.3 beschriebenen multikriteriellen Ablaufplanung. Hierzu werden zunächst die verwendeten Testinstanzen in Abschnitt 6.2.1 dargestellt. Für die Validierung werden anschließend drei Experimente durchgeführt. Das erste dient der Ermittlung geeigneter Parameterwerte für die verschiedenen Parameter des genetischen Algorithmus. Auf Basis der ermittelten Parameterkonfiguration werden anschließend im zweiten Experiment die Testinstanzen in der proaktiven Ablaufplanung verwendet und die Effektivität und Effizienz des Verfahrens werden anhand verschiedener Metriken bestimmt. Die Berücksichtigung der Stabilitätszielfunktion erfolgt abschließend im dritten Experiment.

6.2.1 Testinstanzen für die Validierung der multikriteriellen Ablaufplanung

Für die Validierung der proaktiven Ablaufplanung werden drei Testinstanzen verwendet. Die einzuplanenden Aufträge der ersten Instanz (R5) werden aus der realen Testinstanz R1²³ nach Durchführung des Clusteringverfahrens, aber vor der Durchführung der Klassifizierung für R1, entnommen. Hierdurch wird sichergestellt, dass diese Aufträge für den Klassifikator unbekannt sind und somit die Situation bei einer realen Ablaufplanung nachgebildet wird, mit dem Unterschied, dass die tatsächlichen Bearbeitungszeiten für die Evaluation der Ablaufplanung zur Verfügung stehen. Der für die Testinstanz R1 trainierte Klassifikator wird im Folgenden zur Klassierung der Aufträge aus R5 verwendet. Des Weiteren werden zwei synthetische Testinstanzen (S4 und S5) mit Hilfe der gleichen Verteilungen, die für die Erstellung der Testinstanzen S1 bzw. S2²⁴ verwendet wurden, erzeugt. Zur Klassierung wird der für diese Instanzen trainierte Klassifikator verwendet. In Tabelle 6.7 sind die Aufteilungen der drei Testinstanzen in Auf- ($e_1 = 0$), Abbau- ($e_1 = 1$) und Transferaufträge ($e_1 = 2$) sowie eine weitere Unterteilung nach dem verwendeten Ladehilfsmittel ($e_2 = 0$ für Luftfrachtcontainer und $e_2 = 1$ für -paletten) aufgeführt.

In Abhängigkeit der Auftragsgröße werden in den verwendeten Testinstanzen zwischen zwei und fünf Mitarbeiter benötigt, um einen Auftrag zu bearbeiten. Die Durchschnittswerte liegen für R5 bei 3,1 Mitarbeitern, für S4 bei 2,0 und für S5 bei 3,6 Mitarbeitern.

Die für die Testinstanzen zur Verfügung stehenden Ressourcen sind in Tabelle 6.8 aufgeführt. Es gibt Arbeitsstationen, die ausschließlich für Aufbauaufträge (bezeichnet mit BU) bzw. für Abbauaufträge (bezeichnet mit BD) verwendet werden können. Die Transferarbeitsstationen können für beide vorgenannten Auftragsstypen sowie für Transferaufträge verwendet werden. Des Weiteren ist die zur Verfügung stehende Menge an Personal für Auf- und Abbauaufträge aufgeführt. Für die reale Testinstanz R5 wurde schließlich, auf Basis des realen Luftfrachtterminals, ein vereinfachtes Netzwerkmodell

²³Vgl. Abschnitt 6.1.1.

²⁴Vgl. Abschnitt 6.1.1.

Tabelle 6.7: Anzahl der einzuplanenden Aufträge in den realen und synthetischen Testinstanzen, unterteilt nach den nicht (quasi)stetigen Eigenschaften

Instanz	Anz. einzuplanende Aufträge					
	$e_1 = 0$		$e_1 = 1$		$e_1 = 2$	
	$e_2 = 0$	$e_2 = 1$	$e_2 = 0$	$e_2 = 1$	$e_2 = 0$	$e_2 = 1$
R5	42	39	40	54	0	0
S4	10	15	20	0	0	10
S5	15	15	20	20	10	10

Tabelle 6.8: Für die proaktive Ablaufplanung zur Verfügung stehende Ressourcen

Instanz	Anz. Arbeitsstationen			Anz. Mitarbeiter	
	BU ¹	BD ²	Transfer	BU ¹	BD ²
R5	11	19	15	40	50
S4	3	6	3	12	16
S5	5	9	6	45	36

Legende:¹ Build-up = Aufbauauftrag² Break-down = Abbauauftrag

des innerbetrieblichen Transportsystem erstellt. Für die synthetischen Instanzen S4 und S5 wurde ein abstraktes Netzwerkmodell mit verschiedenen Lagerbereichen und Transportstrecken konstruiert.

Als initialer Planungszeitpunkt für die proaktive Ablaufplanung wird jeweils 7:30 Uhr und als Planungszyklus wird eine Stunde festgesetzt. Weitere Planungszeitpunkte werden für die ersten zwei Experimente (siehe Abschnitt 6.2.2 und 6.2.3) nicht betrachtet. Des Weiteren existiert für diese kein in der Ausführung befindlicher Ablaufplan aus einem vorherigen Planungszyklus. Eine Berücksichtigung des Zielkriteriums der Stabilität entfällt daher bei den zwei nachfolgenden Experimenten. Dieses wird in dem Experiment in Abschnitt 6.2.4 berücksichtigt. Für die Instanz R5 werden Aufträge mit einem Fälligkeitstermin zwischen 10:00 Uhr und 14:00 Uhr und für die Instanzen S4 und S5 Aufträge mit einem Fälligkeitstermin zwischen 10:00 Uhr und 12:00 Uhr betrachtet.

6.2.2 Parametrisierungsexperiment für den genetischen Algorithmus

Mit dem folgenden Parametrisierungsexperiment soll eine geeignete Konfiguration für die verschiedenen Parameter des genetischen Algorithmus ermittelt werden. Eine Auflistung aller Parameter ist in Tabelle 6.9 aufgeführt. Einige Parameterwerte sind bereits im

Tabelle 6.9: Für die proaktive Ablaufplanung zu ermittelnde Parameterwerte und festgelegte Parameter

Parameter	Symbol	Parameterwerte
Populationsgröße	$popsiz$	15, 30 und 50
Rekombinationswahrscheinlichkeit	p^{cx}	50 %, 70 % und 90 %
Verhältnis erster zu zweiter Mutationsart	p^{mut}	0:1, 1:2, 2:1, 1:0
Mutationswahrscheinlichkeit erste Mutationsart	pa^{mut}	auto. ¹ , 3 % und 6 %
Mutationswahrscheinlichkeit zweite Mutationsart	pm^{mut}	auto. ¹ , 3 % und 6 %
Parameter ω	ω	10 %
Parameter φ	φ	5 %
Anz. Szenarien + Anz. Spezialszenarien	$ \mathcal{SZ} $	50 + 4
Prozentsatz des Quantils für alle Cluster	pc	80 %
Anzahl Klassen pro Cluster	nc^*	10
Gewichtungsfaktoren der Aufträge	w_a	$w_a = 1/ A $

Legende:

¹ auto. = Automatische Festlegung der Wahrscheinlichkeit zu $1/|A|$ %

Vorhinein festgelegt (Parameter ω , φ , \mathcal{SZ} , pc , nc^* und w_a), die anderen werden nacheinander auf Basis der im vorherigen Abschnitt beschriebenen drei Testinstanzen auf ihre Eignung hin überprüft. Die Eignung wird anhand der Entwicklung der Zielfunktionswerte der ersten²⁵ und zweiten²⁶ Zielfunktion sowie des Verlaufs der Strafkosten²⁷ für die Überschreitung der Restriktion (5.22) auf Seite 109 für die Individuen der Front \mathcal{F}_1 bzw. durch diese kodierten Ablaufpläne über einen Zeitverlauf von 15 Minuten bemessen. Jede Parameterkonfiguration wird in drei voneinander unabhängigen Läufen ausgewertet und jeweils das arithmetische Mittel gebildet. Die 15 Minuten beginnen nach der Erstellung der initialen Szenariomenge und Population \mathcal{P}_0 . Die benötigten Laufzeiten zu deren Erstellung sind in Tabelle 6.10 für jede Testinstanz aufgeführt. Die Zeiten beziehen sich auf die Erstellung von 50 Szenarien und 4 Spezialszenarien sowie 30 Individuen für die Initialpopulation \mathcal{P}_0 und ergeben sich aus dem arithmetischen Mittel der in dem Parametrisierungsexperiment ermittelten Laufzeiten.

Die Ergebnisse des Parametrisierungsexperiments sind in den Abbildungen 6.2–6.6 dargestellt. Die Abbildungen 6.2(a)–6.2(i) zeigen den Einfluss des Parameters $popsiz$ auf die Verläufe²⁸ der beiden Zielfunktionswerte sowie der Strafkosten aufgrund der

²⁵Siehe Gleichung (5.19) auf Seite 108.

²⁶Siehe Gleichung (5.24) auf Seite 109.

²⁷Siehe Gleichung (5.33) auf Seite 130.

²⁸Der zu Beginn nicht monoton fallende (erste Zielfunktion) bzw. monoton steigende (zweite Zielfunktion) Verlauf der Zielfunktionswerte lässt sich durch die Dominanz von Individuen bzw. Ablaufplänen mit geringeren Strafkosten in dem verwendeten Selektionsmechanismus des NSGA-II-Algorithmus erklären. Sobald eine zulässige Lösung gefunden worden ist, enthält die Front \mathcal{F}_1 ausschließlich zulässige Lösungen und die eigentliche Optimierung der Zielfunktionswerte beginnt.

Tabelle 6.10: Arithmetisches Mittel der benötigten Zeiten zur Erstellung von 54 Szenarien und einer Initialpopulation \mathcal{P}_0 von 30 Individuen für drei Testinstanzen

Instanz	Szenarioerstellung	Initialpopulation
	in s	in s
R5	73	25
S4	21	7
S5	35	12

Restriktionsverletzung. Für die erste Zielfunktion wird bei Verwendung einer mittleren Anzahl von 30 Individuen für den Populationsgrößenparameter $popsiz$ bei allen drei Testinstanzen die geringste Anzahl säumiger Aufträge erzielt. Ein ähnliches Bild zeigt sich bei der Wahl dieses Parameters für die zweite Zielfunktion. Hier wird für die Instanz S5 eine hundertprozentige Robustheit, für R5 der beste Wert und für S4 der zweithöchste Zielfunktionswert, der zudem sehr nahe am höchsten Wert liegt, erzielt. Für zwei der drei Testinstanzen (vgl. Abbildung 6.2(g) und 6.2(i)) wird mit 30 Individuen nach ca. 270 s bzw. 160 s eine Lösung gefunden, die die durch den Parameter ω vorgegebene Restriktion einhält. Für die Testinstanz S4 konnte innerhalb des Zeitraums von 15 Minuten lediglich mit der Parametereinstellung von 15 Individuen eine solche Lösung gefunden werden. Diese Einstellung führte für die Testinstanz R5 allerdings nicht zu einem Absinken der Strafkosten auf 0 %. Für die nachfolgenden Parameterevaluierungen wird daher die Populationsgröße auf 30 Individuen festgelegt.

Die Ergebnisse für die Auswertung des Rekombinationswahrscheinlichkeitsparameters p^{cx} sind in den Abbildungen 6.3(a)–6.3(i) dargestellt. Für eine Rekombinationswahrscheinlichkeit von 90 % werden für die Testinstanzen R5 und S4 die besten Werte sowohl für die erste als auch die zweite Zielfunktion erzielt. Bei der Instanz S5 wird hingegen das schlechteste Ergebnis im Vergleich zu den beiden anderen Parametereinstellungen erreicht. Dieses liegt allerdings nahe an dem jeweils besten Wert. Eine zulässige Lösung wird bei Wahl dieses Parameterwertes für die Instanz R5 nach ca. 430 s und für S5 nach ca. 110 s gefunden. Für die Instanz S4 konnte bei keinem der getesteten Parameterwerte eine zulässige Lösung gefunden werden. Bei einer Rekombinationswahrscheinlichkeit von 90 % werden hier jedoch die geringsten Strafkosten erzielt. Für die nachfolgenden Parameterevaluierungen wird der Parameter p^{cx} auf 90 % festgelegt.

Die Verläufe der Zielfunktionswerte und Strafkosten für den Parametertest zur Variation des Verhältnisses der ersten zur zweiten Mutationsart sind in den Abbildungen 6.4(a)–6.4(i) dargestellt. In den Testinstanzen R5 und S4 zeigen sich bei einer 2:1-Verwendung der ersten (Mutation der Aufträge) zur zweiten (Mutation der Maschinenzuweisung) Mutationsart die besten Ergebnisse für die erste und zweite Zielfunktion. Für die Testinstanz S5 wird jeweils das zweitbeste Ergebnis für die erste und zweite Zielfunktion erzielt. Für die Strafkosten zeigt sich, dass für R5 nach 120 s und für S5 nach 420 s eine zulässige Lösung ermittelt werden kann, was dem besten bzw. drittbesten Ergebnis entspricht. In der Instanz S4 konnte erneut für keine der Parametereinstellungen eine zulässige Lösung innerhalb der 15 Minuten ermittelt werden.

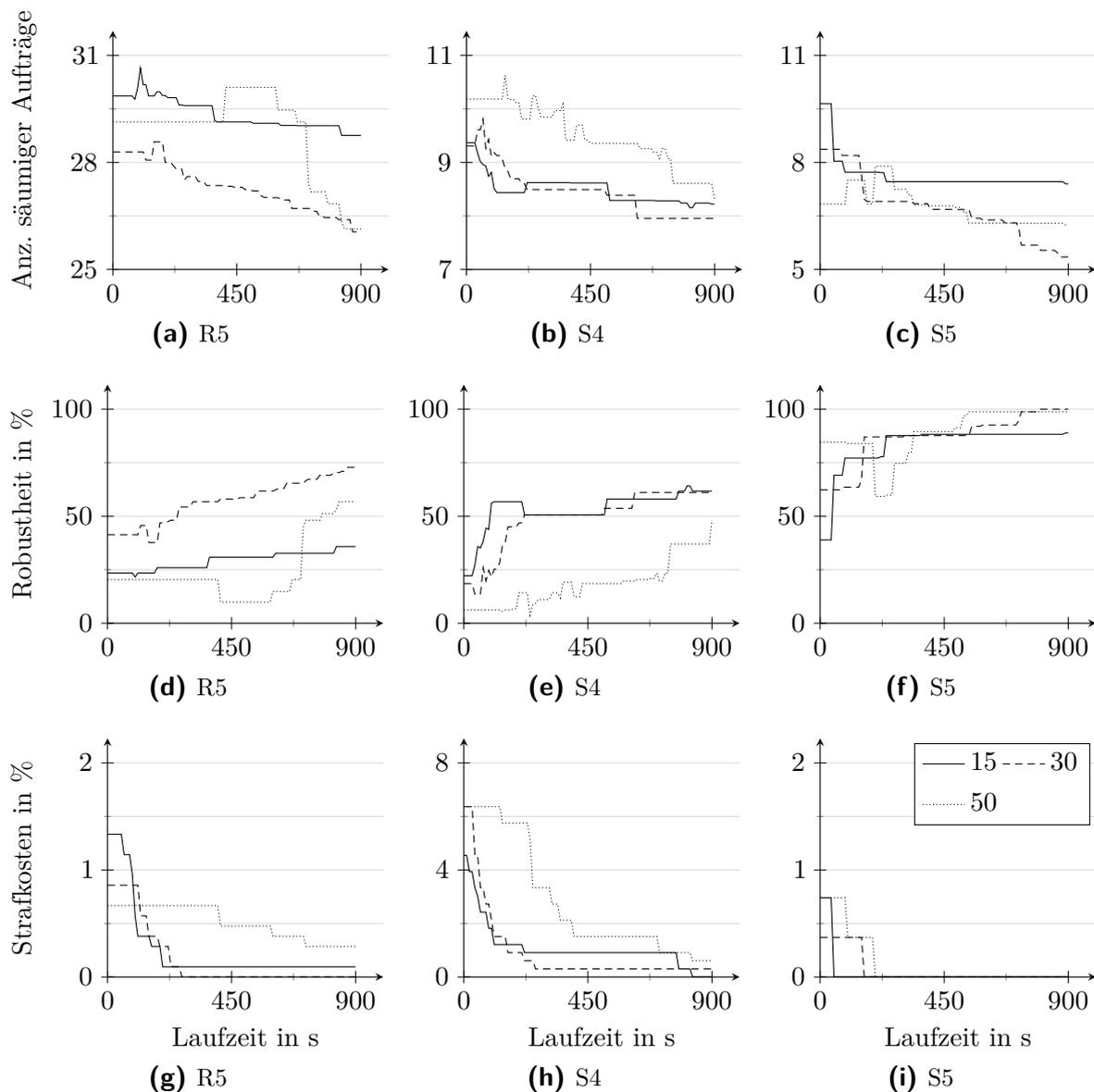


Abbildung 6.2: Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Populationsgröße

Für die weiteren Auswertungen wird der Parameter p^{mut} auf ein Verhältnis von 2:1 festgesetzt.

Die Abbildungen 6.5(a)–6.5(i) stellen den Einfluss der Mutationswahrscheinlichkeit der ersten Mutationsart auf den Verlauf der Zielfunktionswerte und Strafkosten dar. Die Wahl einer automatisch bestimmten Mutationswahrscheinlichkeit ($1/|A|$ %) führte für die Testinstanz R5 zu dem jeweils besten Ergebnis für die erste und zweite Zielfunktion. Für S4 wird bei den beiden Zielfunktionen jeweils das schlechteste Ergebnis und für die Strafkosten das zweitbeste Ergebnis erzielt. Bei der Testinstanz S5 wird für die erste Zielfunktion das schlechteste und für die zweite Zielfunktion und Strafkosten das zweitbeste

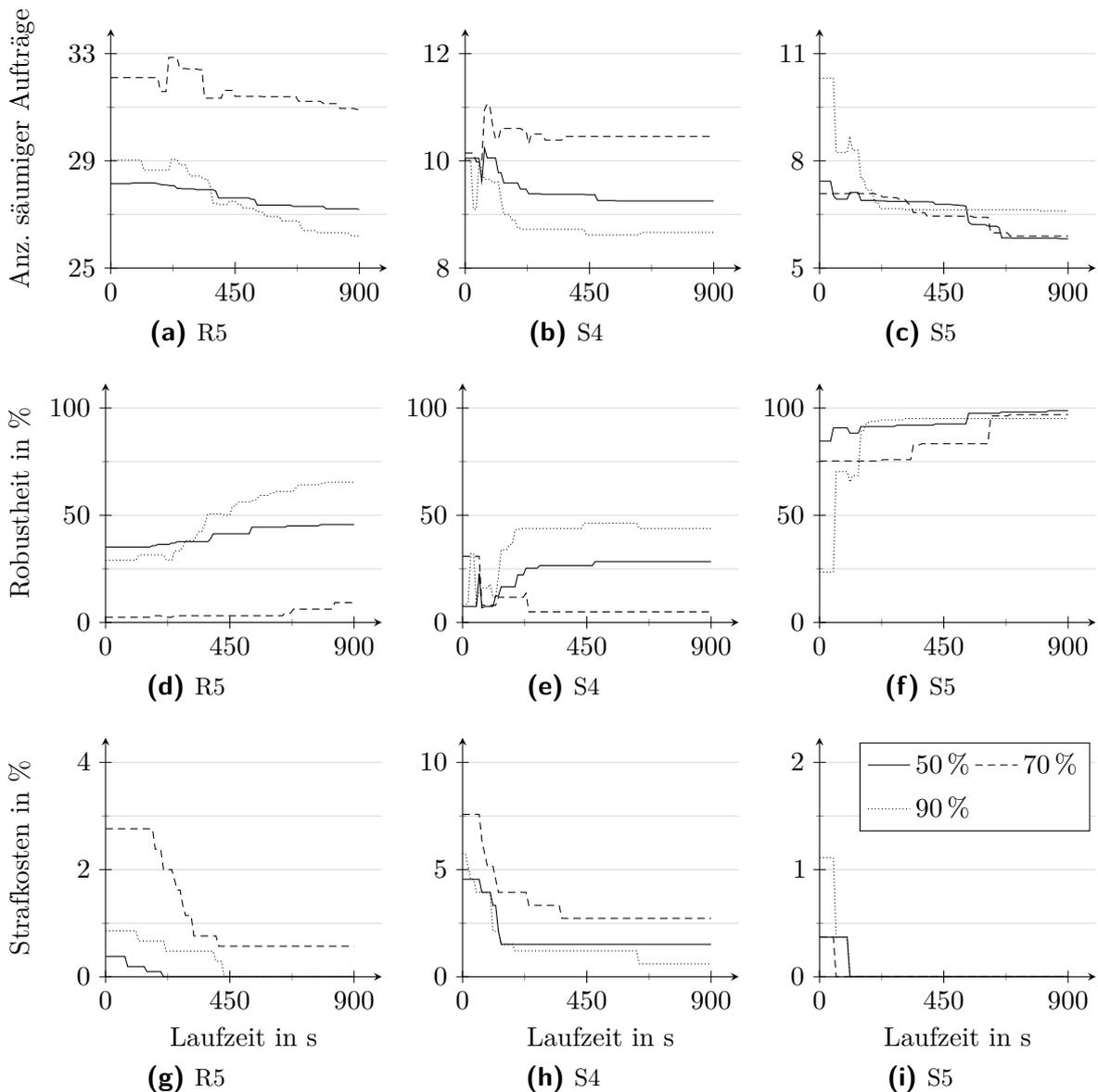


Abbildung 6.3: Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Rekombinationswahrscheinlichkeit für drei Testinstanzen mit einer Populationsgröße von 30

Ergebnis erreicht. Die im Vergleich hierzu guten Ergebnisse für die Parametereinstellung von 6 % müssen im Hinblick auf die Strafkosten in der Testinstanz R5 beurteilt werden. Hier konnten die Strafkosten nicht auf 0 % gesenkt werden. Der Parameter pa^{mut} wird daher im Folgenden auf *auto.* festgelegt, da mit diesem die besten Resultate in der realen Testinstanz R5 erzielt werden konnten.

Die Auswertung des Mutationswahrscheinlichkeitsparameters pm^{mut} für die zweite Mutationsart ist in den Abbildungen 6.6(a)–6.6(i) dargestellt. Die Parametereinstellung von 3 % wird nicht näher betrachtet, da mit dieser in keiner Testinstanz eine zulässige

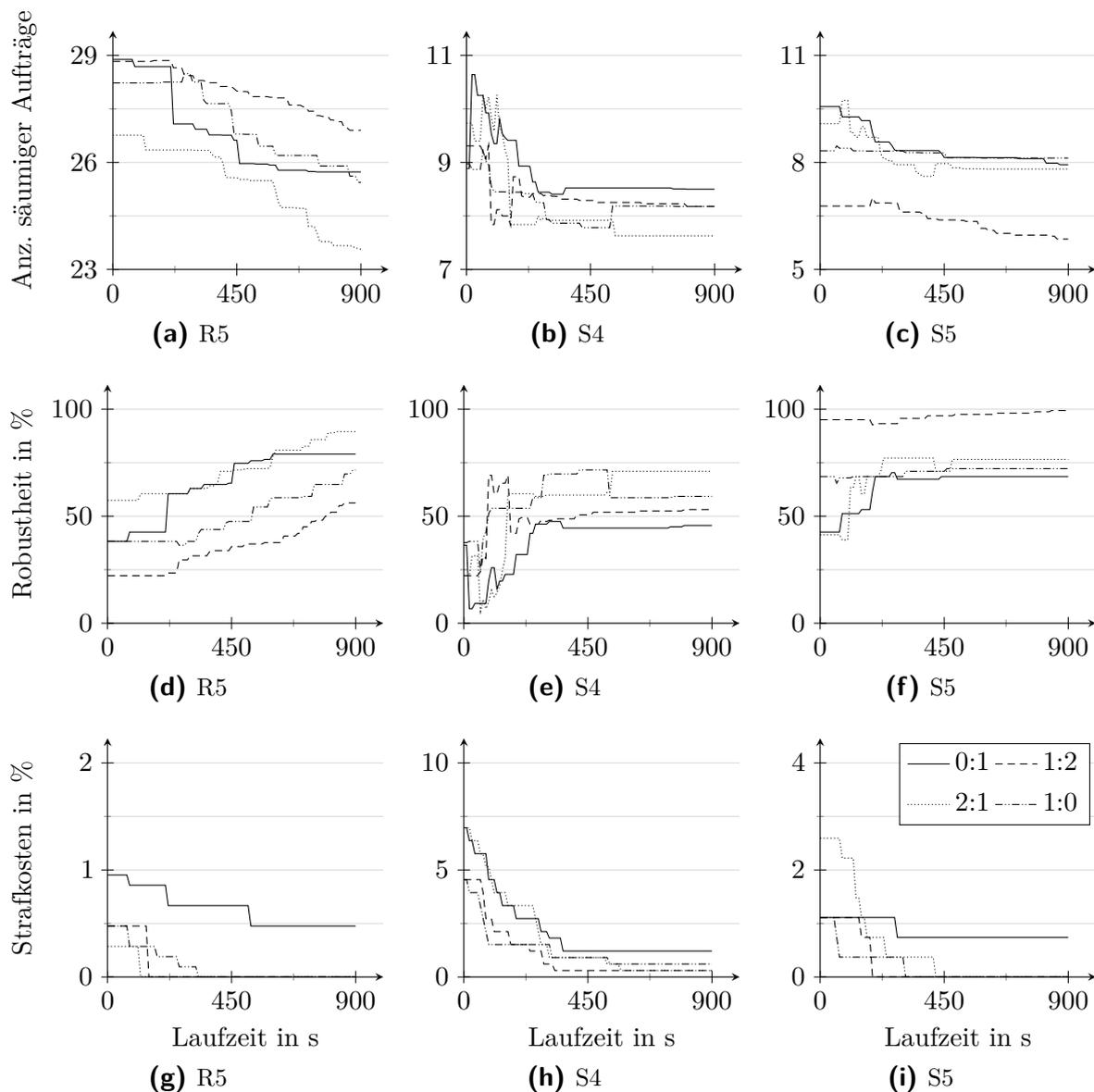


Abbildung 6.4: Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation des Verhältnisses der ersten zur zweiten Mutationsart für drei Testinstanzen mit einer Populationsgröße von 30 und einer Rekombinationswahrscheinlichkeit von 90 %

Lösung gefunden werden konnte. Mit der höchsten Parametereinstellung von 6 % wird für die Testinstanz R5 das zweitbeste Ergebnis für die erste und das beste Ergebnis für die zweite Zielfunktion erreicht. Für S5 zeigt sich das umgekehrte Bild. Für S4 wird in beiden Zielfunktionen das beste Ergebnis erzielt. Für S4 konnte erneut mit keiner Parametereinstellung eine zulässige Lösung innerhalb von 15 Minuten gefunden werden. Es werden mit 6 % jedoch die geringsten Strafkosten erzielt. Der Parameter pm^{mut} wird somit auf den Wert 6 % festgelegt.

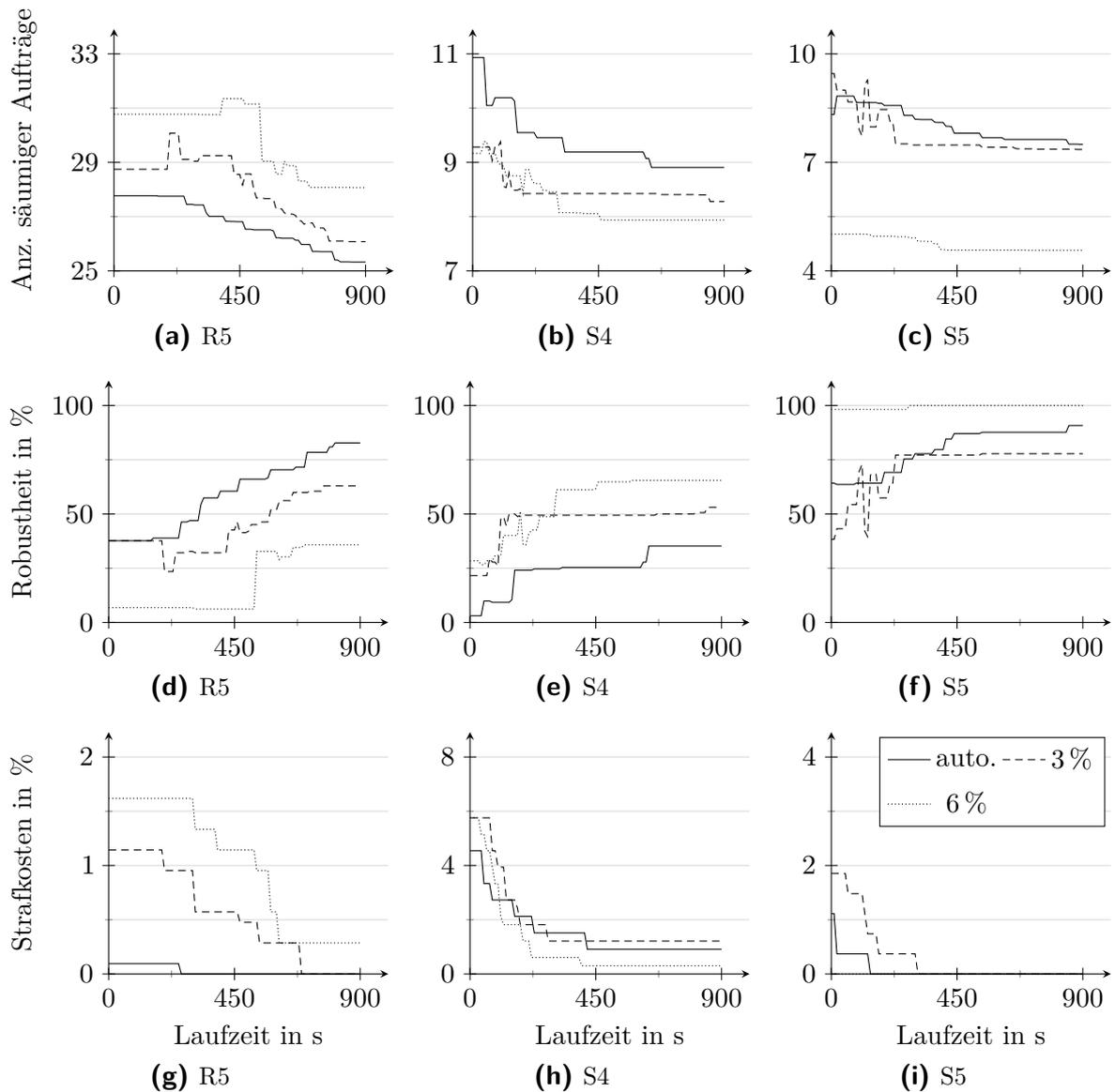


Abbildung 6.5: Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Mutationswahrscheinlichkeit der ersten Mutationsart für drei Testinstanzen mit einer Populationsgröße von 30, einer Rekombinationswahrscheinlichkeit von 90% und einem Verhältnis der ersten zur zweiten Mutationsart von 2:1

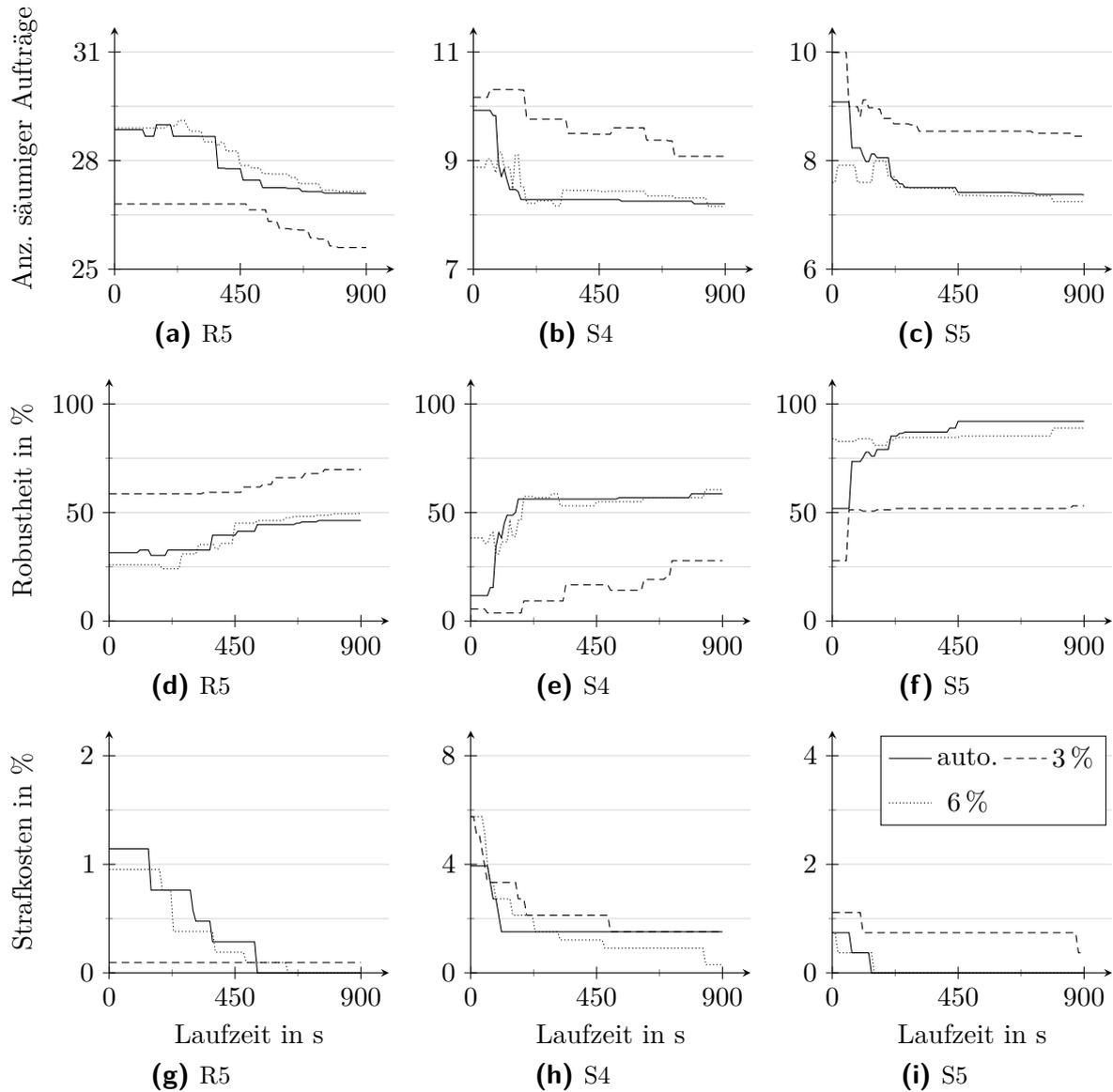


Abbildung 6.6: Durchschnittliche Verläufe der Zielfunktionswerte und Strafkosten unter Variation der Mutationswahrscheinlichkeit der zweiten Mutationsart für drei Testinstanzen mit einer Populationsgröße von 30, einer Rekombinationswahrscheinlichkeit von 90%, einem Verhältnis der ersten zur zweiten Mutationsart von 2:1 und einer Mutationswahrscheinlichkeit von $1/|A|$ % (Parameter *auto.*) für die erste Mutationsart

In dem vorherigen Parametrisierungsexperiment zeigte sich, dass für die Zielfunktionswerte der ersten Zielfunktion im Verlauf der 15 Minuten nur geringe Verbesserungen erzielt werden können. Dies ist damit zu erklären, dass bei der Erstellung der Initialpopulation \mathcal{P}_0 , die für die Spezialszenarien bereits erstellt und hinsichtlich der ersten Zielfunktion optimierten Individuen übernommen werden.²⁹ Somit sind bereits zu Beginn des Verfahrens gute Lösungen für die erste Zielfunktion in der Population vorhanden. Die verbleibenden säumigen Aufträge am Ende der 15 Minuten lassen sich zum einen mit fehlenden Ressourcen, insbesondere Maschinen, und zum anderen mit der durch den Parameter ω erzeugten Restriktion erklären. Für die Testinstanz S4 wird nur in einem der fünf Fälle mit den getesteten Parametereinstellungen eine zulässige Lösung innerhalb von 15 Minuten gefunden. Dies lässt sich mit einem für diese Testinstanz zu restriktiv gewählten Parameter ω erklären. Eine Reduktion des Parameters um die jeweilige Höhe der Strafkosten am Ende der Laufzeit wäre eine Alternative.

Im Folgenden werden für den genetischen Algorithmus der proaktiven Ablaufplanung 30 Individuen pro Generation verwendet. Die Rekombinationswahrscheinlichkeit wird auf 90 % festgelegt. Die Auswahlhäufigkeit zwischen erster und zweiter Mutationsart wird mit 2:1 definiert. Somit liegt die Präferenz in der Ermittlung alternativer Auftragsreihenfolgen auf den Arbeitsstationen im Vergleich zum Auffinden alternativer Zuordnungen der Aufträge zu den Arbeitsstationen. Die Mutationswahrscheinlichkeit der ersten Mutationsart wird in Abhängigkeit der Anzahl einzuplanender Aufträge bestimmt und berechnet sich über $1/|\mathcal{A}|$ %. Die Mutationswahrscheinlichkeit der zweiten Mutationsart wird auf 6 % festgelegt.

6.2.3 Effektivität und Effizienz der proaktiven Ablaufplanung

Im Folgenden wird die Effektivität und Effizienz der proaktiven Ablaufplanung untersucht. Hierzu werden die im vorausgegangenen Parametrisierungsexperiment ermittelten Parameter sowie die in Tabelle 6.9 bereits festgelegten Parametereinstellungen³⁰ verwendet und die Ablaufplanung wird für die in Abschnitt 6.2.1 beschriebenen Testinstanzen durchgeführt.

Zur Bemessung der Effektivität des Verfahrens wird der durch die proaktive Ablaufplanung ermittelte Ablaufplan mit der höchsten Robustheit (im Folgenden als RA bezeichnet) sowie der Ablaufplan mit der geringsten Anzahl säumiger Aufträge (im Folgenden als PA bezeichnet) unter Berücksichtigung der generierten Szenarien mit den folgenden alternativ³¹ erstellten Ablaufplänen hinsichtlich der Einhaltung der Parameter ω und φ verglichen:

- Erwartungswertoptimaler Ablaufplan (EA): Bei diesem Ablaufplan erfolgt die Optimierung hinsichtlich eines Erwartungswertszenarios für die Bearbeitungszeiten der Aufträge (vgl. Ablaufplan $ap_{s_{zex}}^*$ auf Seite 124).

²⁹Siehe Abschnitt 5.3.3.2 in dem das Vorgehen erläutert wird.

³⁰Die Anzahl der generierten Szenarien wird in diesem Experiment auf 100 erhöht. Hinzu kommen, wie zuvor, die vier Spezialszenarien.

³¹Siehe Abschnitt 5.3.3.2 für eine Erläuterung, wie diese Ablaufpläne erstellt werden.

Tabelle 6.11: Vergleich der Ablaufpläne RA, PA, EA und CA anhand von vier Metriken für die verschiedenen Testinstanzen

Inst.	A.-Plan	Anz. $sz > \omega$	Max. Regret in %	\emptyset Regret in %	Robustheit in %
R5	RA	0	6,67	2,44	96,15
	PA	0	6,67	2,44	96,15
	EA	6	20,57	4,41	68,27
	CA	6	22,29	4,43	60,58
S4	RA	12	10,48	4,41	62,50
	PA	12	10,48	4,41	62,50
	EA	21	27,27	4,60	51,92
	CA	18	21,82	3,64	68,27
S5	RA	0	5,19	0,75	100,00
	PA	0	5,19	0,75	100,00
	EA	2	14,44	1,82	96,15
	CA	6	14,44	4,46	64,42

- Optimaler Ablaufplan für die geschätzte Bearbeitungszeit (CA): Unter Verwendung einer Formel zur Abschätzung der Bearbeitungszeit jedes Auftrags wird ein optimaler Ablaufplan erstellt (vgl. Ablaufplan ap_{szco}^* auf Seite 124).

Grundlage des Vergleichs bildet der jeweils szenariooptimale Ablaufplan. Dieser Ablaufplan entspricht dem für das jeweilige Szenario besten Ablaufplan.³² Dieser würde ausgewählt werden, wenn ein deterministisches Planungsumfeld vorliegen würde, bei dem das eintretende Bearbeitungszeitszenario zu Beginn der Planung bekannt wäre. Das Experiment wird unter der Annahme, dass die rollierende Ablaufplanung einen Planungszyklus von einer Stunde aufweist, über einen Zeitraum von jeweils 25 Minuten durchgeführt und wird für jede Testinstanz dreimal wiederholt. Aus den jeweiligen Ergebnissen wird schließlich das arithmetische Mittel gebildet.

Die Ergebnisse der Auswertung sind in Tabelle 6.11 aufgeführt. Die Werte für die beiden Ablaufpläne RA und PA sind identisch, da in den jeweiligen Testinstanzen die Front \mathcal{F}_1 bei diesem Experiment ausschließlich eine Lösung enthielt, die somit zugleich dem Ablaufplan mit der geringsten Anzahl säumiger Aufträge sowie mit der höchsten Robustheit entsprach. Hierbei zeigt sich, dass aufgrund der Definition der beiden Zielfunktionen eine Reduzierung der durchschnittlichen Anzahl säumiger Aufträge tendenziell mit einer Erhöhung des hier beschriebenen Robustheitsmaßes einhergeht.

Eine mögliche Ausnahme, in der beide Zielfunktionen konfliktär zueinander sind und \mathcal{F}_1 damit mehrere Ablaufpläne enthielte, ergibt sich in dem folgenden Fall: Ein Ablaufplan nimmt für die Gleichung (5.17) bei einigen Szenarien im Vergleich zu den

³²Hierfür wird auf den für das jeweilige Szenario mit einem genetischen Algorithmus erstellten Ablaufplan zurückgegriffen (siehe Fußnote 74 auf Seite 123).

anderen besonders kleine Werte an. Ein anderer Ablaufplan nimmt für den überwiegenden Teil der Szenarien bei der Gleichung (5.17) kleinere Werte als der erste Ablaufplan an. Für einige Szenarien hingegen werden besonders große Werte erzielt, die sich aber noch innerhalb der durch ω festgelegten Restriktion befinden. In diesem Fall könnte der erste im Vergleich zum zweiten Ablaufplan in Abhängigkeit des Parameters φ bessere Werte für die erste, aber schlechtere Werte für die zweite Zielfunktion erreichen. Beide Pläne wären somit pareto-optimal.

Des Weiteren zeigt sich, dass der Ablaufplan RA bzw. PA in den beiden Testinstanzen R5 und S5 die durch den Parameter ω vorgegebene Restriktion für alle betrachteten Szenarien einhält (siehe dritte Spalte). Der maximale Regret betrug dabei 6,67 % bzw. 5,19 % und lag somit unter dem für den Parameter ω festgesetzten 10 % (siehe vierte Spalte). Für den Ablaufplan CA wurde der Parameter ω jeweils sechsmal in R5 bzw. S5 und für EA sechs- bzw. zweimal überschritten. Der maximale Regret lag für R5 bei beiden Ablaufplänen über 20 % und für S5 bei 14,44 %. Wie sich schon in dem vorherigen Parametrisierungsexperiment zeigte, ist es für die Testinstanz S4 sehr schwer eine zulässige Lösung zu ermitteln. Dementsprechend stellt keiner der hier ermittelten Ablaufpläne eine zulässige Lösung für diese Instanz dar. Die Anzahl an Szenarien, bei denen ω überschritten wird, und der maximale Regret sind für RA bzw. PA aber erneut deutlich niedriger als für die beiden anderen Ablaufpläne.

Für den durchschnittlichen Regret und die ermittelte Robustheit zeigt sich ein ähnliches Bild wie zuvor. Der Ablaufplan RA bzw. PA erzielt hier für die Testinstanzen R5 und S5 die besten Ergebnisse mit einem durchschnittlichen Regret von 2,44 % bzw. 0,75 % (siehe fünfte Spalte) sowie einer Robustheit von 96,15 % und 100 % (siehe sechste Spalte). Für die Testinstanz S4 werden die jeweils zweitbesten Ergebnisse erzielt. Hier werden mit dem Ablaufplan CA die besten Werte erreicht, wobei die ermittelte Robustheit vor dem Hintergrund einer nicht zulässigen Lösung zu sehen ist.

Zur Beurteilung der Effizienz des Verfahrens werden die Laufzeiten der einzelnen Schritte des genetischen Algorithmus zur proaktiven Ablaufplanung näher betrachtet. Zur Erstellung der 100 Szenarien und 4 Spezialszenarien wird annähernd doppelt so viel Zeit benötigt wie zur Erstellung der 54 Szenarien in Tabelle 6.10. Die Zeit zur Erstellung der Initialpopulation bleibt aufgrund des nicht geänderten Parameters hingegen gleich. In dem iterativen Prozess der Verbesserung der Individuen durch Anwendung der genetischen Operatoren macht die Evaluation der Individuen den größten Teil der benötigten Zeit aus. Hierbei ist die Zeit zur Ermittlung des Worst-Case-Szenario sz_{η}^{dis} für ein Individuum η zu vernachlässigen (siehe Tabelle 6.12). Die sich aus dem Modell (5.32) ergebenden Instanzen sind nach Anwendung des Pre-Solve-Schrittes des Optimierers Gurobi³³ klein genug, dass für die größte Testinstanz R5 im Durchschnitt nur 21,03 ms benötigt werden, um das Modell optimal zu lösen. Der Hauptanteil der Zeit bei der Evaluation eines Individuums η wird für die Ermittlung des optimalen Ablaufplans $ap_{sz_{\eta}^{\text{dis}}}^*$ für das Worst-Case-Szenario sz_{η}^{dis} benötigt. Für die Testinstanz R5 sind dies im Durchschnitt 7,5 s, für S4 2,1 s und für S5 3,7 s. Dieser Schritt muss für jedes Individuum in jeder Iteration durchgeführt werden und hat somit

³³Verwendet wurde die Version 6.0.5.

Tabelle 6.12: MIP-Statistiken des Modells zur Berechnung des Worst-Case-Szenarios nach dem Pre-Solve-Schritt für die verschiedenen Testinstanzen

Inst.	Anz. Rows	Anz. Columns	Anz. Nonzeros	Anz. Binaries	Solving Time in ms
R5	43,03	35,38	131,45	8,14	21,03
S4	16,82	16,64	52,50	3,21	10,00
S5	13,45	13,21	40,86	2,72	10,00

Tabelle 6.13: Anzahl der Aufträge mit unterschiedlichem Ausführungsstatus für die verschiedenen Testinstanzen zum Zeitpunkt $t_1 = 8:30$ Uhr

Instanz	$ \mathcal{A}_{t_1}^C $	$ \mathcal{A}_{t_1}^R $	$ \mathcal{A}_{t_1}^F $	$ \mathcal{A}_{t_1}^P $	$ \mathcal{A}_{t_1}^N $
R5	11	41	38	85	16
S4	1	12	12	30	28
S5	8	20	17	45	45

entscheidenden Einfluss auf die innerhalb eines vorgegebenen Zeitraums durchführbaren Evaluationen.

6.2.4 Betrachtung der rollierenden Ablaufplanung

In dem folgenden Experiment wird die Stabilitätszielfunktion zusätzlich zu den bereits zuvor betrachteten zwei Zielfunktionen berücksichtigt.³⁴ Hierfür wird die Situation zum Zeitpunkt 8:55 Uhr betrachtet, zu dem ein Ablaufplan ap_1^P aus der Menge \mathcal{AP}_1 auszuwählen und um 9:00 Uhr auszuführen ist. Diese Menge wurde von 8:30 Uhr bis 8:55 Uhr erstellt und berücksichtigte den in der Ausführung befindlichen Ablaufplan³⁵ ap_0^P für die Stabilitätsberechnungen sowie für die Instanz R5 alle Aufträge mit einem Fälligkeitstermin bis 15:00 Uhr und für die Instanzen S4 und S5 Aufträge mit einem Fälligkeitstermin bis 13:00 Uhr.³⁶

In Tabelle 6.13 ist die Anzahl der Aufträge mit unterschiedlichem Ausführungsstatus für die drei Testinstanzen zum Zeitpunkt $t_1 = 8:30$ Uhr aufgelistet. Die Zuordnungen der Aufträge aus $\mathcal{A}_{t_1}^R$ und $\mathcal{A}_{t_1}^F$ sind fix, beeinflussen jedoch die Ablaufplanung. Aufträge aus $\mathcal{A}_{t_1}^P$ können umgeplant werden, wobei Planänderungen durch die Stabilitätszielfunktion berücksichtigt werden.³⁷ Somit sind für die Erstellung der Ablaufplanmenge \mathcal{AP}_1

³⁴Es wird erneut von der Annahme ausgegangen, dass der Planungszyklus eine Stunde beträgt und die Ablaufplanung über 25 Minuten durchgeführt wird.

³⁵Der Ablaufplan $ap_0^P \in \mathcal{AP}_0$ wurde um 7:55 Uhr ausgewählt und um 8:00 Uhr implementiert und ausgeführt. Die Menge \mathcal{AP}_0 wurde von dem initialen Planungszeitpunkt um 7:30 Uhr bis um 7:55 Uhr erstellt (siehe Abschnitte 6.2.1 und 6.2.3).

³⁶Vgl. Abschnitt 6.2.1.

³⁷Vgl. das Modell (5.30) auf Seite 119 und die Erläuterungen in Abschnitt 5.3.2.

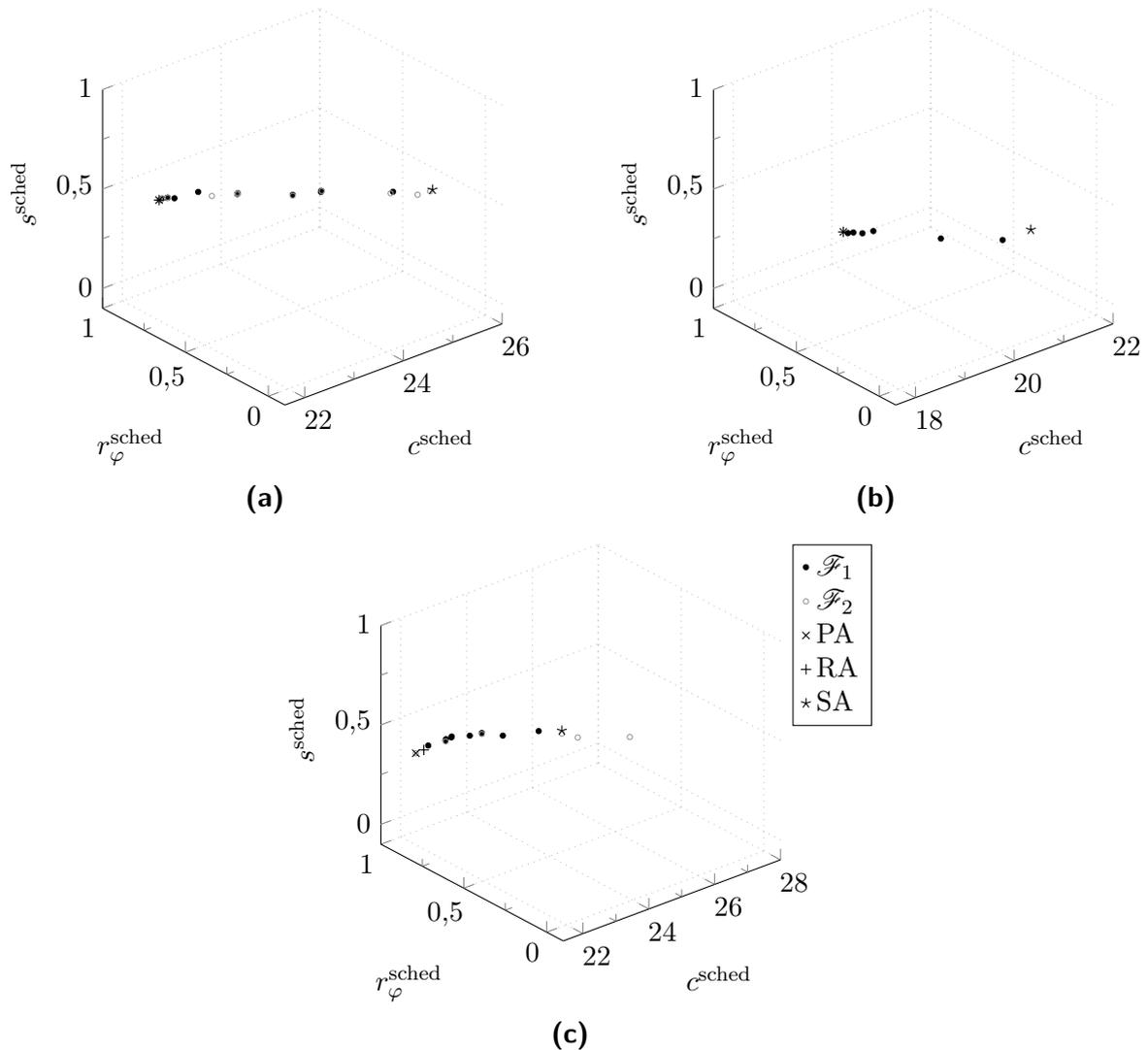


Abbildung 6.7: Pareto-Fronten der pareto-optimalen Ablaufpläne für drei Testinstanzen:
 (a) Testinstanz R5 (b) Testinstanz S4 (c) Testinstanz S5

die Aufträge aus $\mathcal{A}_{t_1}^P$ und $\mathcal{A}_{t_1}^N$ einzuplanen, wobei die Aufträge aus $\mathcal{A}_{t_1}^N$ erstmalig eingeplant werden.

In den Abbildungen 6.7(a)–6.7(c) sind die verschiedenen Pareto-Fronten aus \mathcal{AP}_1 für die drei Testinstanzen aufgeführt.³⁸ Für die Testinstanz R5 wurden 26 unterschiedliche Ablaufpläne ermittelt, von denen 14 auf \mathcal{F}_1 entfallen und damit pareto-optimal sind. Für S4 befinden sich alle 8 ermittelten Pläne in der ersten Front, von denen jedoch keiner eine zulässige Lösung darstellt, da der Parameter ω nicht eingehalten wird. Für S5 sind schließlich 18 unterschiedliche Ablaufpläne ermittelt worden, von denen sich 12 in \mathcal{F}_1 befinden.

³⁸Ablaufpläne, die hinsichtlich der Zielfunktionen und Strafkosten identisch waren, wurden entfernt.

Tabelle 6.14: Vergleich der Ablaufpläne RA, PA und SA anhand von sechs Metriken für die verschiedenen Testinstanzen

Inst.	A.-Plan	Anz. $sz > \omega$	Max.	\emptyset Regret	c^{sched}	$r_{\varphi}^{\text{sched}}$	s^{sched}
			Regret in %	in %		in %	in %
R5	RA	0	6,93	3,69	22,26	85,58	44,71
	PA	0	6,93	3,69	22,26	85,58	44,71
	SA	0	9,90	6,94	25,54	18,27	49,29
S4	RA	2	13,79	4,11	18,78	56,73	36,64
	PA	2	13,79	4,11	18,78	56,73	36,64
	SA	25	13,79	7,71	20,87	5,77	40,91
S5	RA	0	7,78	1,54	22,65	99,04	30,06
	PA	0	10,00	1,34	22,41	99,04	29,91
	SA	0	10,00	4,24	25,20	66,35	38,47

Des Weiteren sind in den Abbildungen jeweils PA und RA sowie der Ablaufplan mit der höchsten Stabilität (im Folgenden als SA bezeichnet) aus der Front \mathcal{F}_1 hervorgehoben. Für jede Testinstanz werden diese drei Ablaufpläne unter Berücksichtigung der generierten Szenarien und des zugehörigen szenariooptimalen Ablaufplans hinsichtlich der Einhaltung der Parameter ω und φ verglichen. Tabelle 6.14 zeigt die Ergebnisse der Auswertung. Im Gegensatz zu Abschnitt 6.2.3 sind bei diesem Experiment PA und RA für S5 nicht identisch, jedoch sehr ähnlich. Beide Ablaufpläne halten in R5 und S5 den Parameter ω für alle Szenarien ein und zeigen gute Werte für den durchschnittlichen Regret sowie für die Robustheit. Bei dem maximalen Regret erzielt RA in S5 die geringsten Werte auf Kosten eines leicht höheren durchschnittlichen Regrets im Vergleich zu PA.

Auch in diesem Experiment konnte für die Testinstanz S4 kein zulässiger Ablaufplan gefunden werden (siehe dritte Spalte). Der Parameter ω wird jedoch durch PA bzw. RA in wesentlich mehr Szenarien eingehalten als durch SA. Für SA zeigt sich zudem in jeder Testinstanz der negative Einfluss einer hohen Planstabilität sowohl auf den maximalen als auch den durchschnittlichen Regret und die Robustheit. Eine hohe Planstabilität verringert die Freiheitsgrade, Aufträge aus $\mathcal{A}_{t_1}^P$ umplanen zu können und somit bessere Werte für die beiden anderen Zielfunktionen zu erzielen. Demgegenüber ist die Stabilität des Ablaufplans RA nur geringfügig kleiner als für SA. Hier liegt die Trade-Off-Entscheidung, ob die bestmögliche Planstabilität die wesentlich geringere Robustheit aufwiegt oder aber ein ausgewogenerer Ablaufplan vorteilhafter ist, beim Entscheidungsträger.

Der zusätzliche zeitliche Aufwand zur Evaluation der Stabilitätszielfunktion beeinträchtigt nicht die Effizienz des Verfahrens zur proaktiven Ablaufplanung.

7 Zusammenfassung und Ausblick

Das steigende Luftfrachtaufkommen, ein kompetitiver Markt und ein klassischerweise stark kapazitativ beschränktes System stellt die Luftfrachtindustrie vor besondere Herausforderungen. Den Hauptanteil der Transportzeit für Luftfracht machen bodengebundene Prozesse aus, wobei den Luftfrachtterminals als Schnittstelle zwischen dem Land- und Luftverkehr eine zentrale Bedeutung bei der effektiven und effizienten Durchführung von Umschlagprozessen für den Export, Import sowie den Transfer zukommt.

7.1 Zusammenfassung

Für eine effektive und effiziente Durchführung der Umschlagprozesse innerhalb des Systems „Luftfrachtterminal“ ist eine Ablaufplanung und -steuerung, die die räumliche und zeitliche Koordination der Frachtabfertigungsaufträge unter Berücksichtigung der zur Verfügung stehenden Ressourcen und vielfältigen Restriktionen vornimmt, essenziell. Das dynamische und durch Nicht-Determinismus geprägte Planungsumfeld stellt einen weiteren Aspekt dar, den es zu beachten gilt. Ziel dieser Arbeit war daher die Konzeption eines Systems zur Ablaufplanung und -steuerung für die im Rahmen der Umschlagprozesse in Luftfrachtterminals durchzuführenden Frachtabfertigungsaufträge unter Berücksichtigung des dynamischen und stochastischen Planungsumfeldes.

Aufbauend auf einer Beschreibung der Luftfrachttransportkette und der involvierten Akteure sowie einer detaillierten Darstellung der internen Prozesse im System „Luftfrachtterminal“ erfolgte die Identifizierung von drei Problemfeldern und die Definition von Anforderungen, die es bei der Planung und Steuerung der Abfertigungsprozesse zu berücksichtigen galt: Erstens die Möglichkeit auf Basis einer vorhandenen Datenbasis automatisiert einen Informationsstand abzuleiten, der eine probabilistische Bewertung von Bearbeitungszeiten zukünftiger Aufträge erlaubt. Zweitens die eigentliche Ablaufplanung zur Erstellung robuster Ablaufpläne unter Beachtung eines multikriteriellen Zielsystems. Schließlich als drittes Problemfeld die Ablaufsteuerung.

Ausgehend von den identifizierten Problemfeldern wurde der Stand der Forschung erarbeitet und bestehende Arbeiten wurden auf ihre Eignung hinsichtlich der definierten Anforderungen analysiert. Für das erste Problemfeld wurden Klassenbildungs- sowie Klassifikationsverfahren betrachtet. Hierbei erwiesen sich das partitionierende k-medoids-Verfahren zur Klassenbildung sowie ein naiver Bayes-Klassifikator zur Klassifikation als geeignete Grundlage zur problemspezifischen Anpassung. Im Anschluss wurden exakte sowie heuristische Verfahren, die eine proaktive Ablaufplanung ermöglichen, näher untersucht und hinsichtlich der Anforderungen für das zweite Problemfeld bewertet. Aufgrund der Komplexität des Problems erwiesen sich evolutionäre Metaheuristiken

als zielführend. Ein Literaturüberblick zeigte, dass keine Arbeiten bekannt sind, die alle zuvor definierten und praxisrelevanten Anforderungen umsetzen. Schließlich wurden für das dritte Problemfeld verschiedene Verfahren zur Umplanung im Rahmen der Ablaufsteuerung untersucht. Partielle Umplanungsverfahren erwiesen sich hierbei als vielversprechender Ansatz. Auf Basis der Literaturanalyse konnte sodann die zu leistende Arbeit durch sieben Arbeitspakete, die den einzelnen Problemfeldern zugeordnet wurden, präzisiert werden.

Der Adressierung der einzelnen Arbeitspakete ging die Erstellung einer Architektur voraus, in der die für die Problemfelder zu entwickelnden Verfahren als abstrakte Module eingeordnet und ihre Interdependenzen in Form eines Ablaufprozesses beschrieben wurden. Für die Ableitung eines Informationsstandes aus der -basis wurde ein Prozess beschrieben, der die Daten auf Basis einer Clusteranalyse zu Intervallen und Häufigkeitsverteilungen der Bearbeitungszeitabweichungen aggregiert. Die erstellten Cluster und die in diesen hinterlegten Informationen bildeten anschließend die Ausgangsbasis zur Erstellung eines Klassifikators, der in der proaktiven Ablaufplanung eingesetzt werden kann.

Für die proaktive Ablaufplanung wurde zunächst eine Formalisierung des Problems unter der Annahme eines statischen Planungsumfeldes vorgenommen. Berücksichtigt wurde die Zielfunktion zur Minimierung der erwarteten gewichteten Anzahl säumiger Aufträge sowie eine weitere zur Quantifizierung der Robustheit. Das dynamische Planungsumfeld wurde anschließend durch die Einbettung des Modells in eine rollierende Ablaufplanung und die Einführung einer weiteren Zielfunktion – zur Quantifizierung der Planstabilität – in die Formalisierung adressiert. Auf Basis dieses mathematischen Modells wurde eine evolutionäre Metaheuristik auf Grundlage des NSGA-II-Algorithmus entworfen, welche Ablaufpläne unter Berücksichtigung des multikriteriellen Zielsystems erstellt. Anschließend erfolgte die Formalisierung der Personaleinsatzplanung und Materialflussteuerung und deren Integration in die proaktive Ablaufplanung.

Zur Berücksichtigung von Ereignissen, deren Auswirkungen größer sind als durch den robusten Ablaufplan kompensiert werden können, wurde eine Ablaufsteuerung konzipiert, die den aktiven Ablaufplan in diesen Fällen anpassen kann. Zur Erkennung dieser Ereignisse wurde die Notwendigkeit einer Ausführungsüberwachung aufgezeigt und eine Unterscheidung in Primär- und Sekundärstörungen vorgenommen. Die Zuordnung der erkannten Ereignisse zu verschiedenen Auftragsmengen determinierte hierbei das zur Verfügung stehende Verfahren bzw. Vorgehen.

Für die experimentelle Validierung der konzipierten Verfahren zur Clusterbildung und Klassifikatorerstellung wurden sieben Testinstanzen verwendet. Vier dieser Instanzen basierten auf Realdaten eines großen Luftfrachtterminals. Drei weitere Instanzen wurden synthetisch erstellt. Für die Validierung des Clusteringverfahrens wurden zwei Experimente durchgeführt. Das erste diente dabei der Ermittlung, welche von drei Initialisierungsmethoden zur Bestimmung der initialen Clustermedoiden am geeignetsten ist. Hierbei wurde auf Basis von drei Metriken die Initialisierungsmethode *Farthest-First* als geeignet identifiziert. Das zweite Experiment diente der Ermittlung der Anzahl zu betrachtender Cluster auf Basis der Veränderung des Silhouettenkoeffizienten. Anschließend wurde die Effektivität und Effizienz des Klassifikationsverfahrens in zwei weiteren

Experimenten untersucht. Die Auswertung von vier Metriken zeigte die Eignung des Verfahrens für eine zuverlässige Klassierung einzuplanender Aufträge. Hinsichtlich der Effizienz wurden Laufzeiten erzielt, die für die Ablaufplanung vernachlässigbar gering sind.

Auf Basis einer realen und zweier synthetischer Testinstanzen wurden zur Validierung der proaktiven Ablaufplanung drei weitere Testinstanzen geschaffen. Anschließend wurden drei Experimente durchgeführt: Das erste diente der Parametrisierung des genetischen Algorithmus und wurde für die Parameter „Populationsgröße“, „Rekombinationswahrscheinlichkeit“, „Verhältnis erster zu zweiter Mutationsart“, „Mutationswahrscheinlichkeit erste Mutationsart“ sowie „Mutationswahrscheinlichkeit zweite Mutationsart“ durchgeführt. Das zweite Experiment diente der Ermittlung der Effektivität und Effizienz der proaktiven Ablaufplanung, indem die durch dieses Verfahren erstellten pareto-optimalen Ablaufpläne RA bzw. PA mit einem erwartungswertoptimalen EA und einem für die geschätzte Bearbeitungszeit optimierten Ablaufplan CA hinsichtlich der Einhaltung der Parameter ω und φ verglichen wurden. Im Vergleich mit den beiden letztgenannten Ablaufplänen wurde durch den robusten Ablaufplan RA der Parameter ω in zwei der drei Testinstanzen für jedes Szenario eingehalten. Des Weiteren war der maximale und durchschnittliche Regret zur szenariooptimalen Lösung bei diesem signifikant geringer. Die erzielten Laufzeiten zeigten zudem, dass der Einsatz der proaktiven Ablaufplanung in der Praxis durchführbar ist. Schließlich wurde in dem dritten Experiment die Stabilitätszielfunktion berücksichtigt und die Pareto-Fronten für die drei Testinstanzen wurden ermittelt und beschrieben. Für die jeweiligen Ablaufpläne RA, PA und SA wurde erneut die Einhaltung der Parameter ω und φ untersucht. Hierbei zeigte sich der negative Einfluss einer hohen Planstabilität – wie sie durch den Ablaufplan SA erzielt wird – auf die beiden anderen Zielfunktionen. Ein ähnlich negativer Einfluss einer hohen – und durch den Ablaufplan RA forcierten – Robustheit auf die Planstabilität konnte nicht festgestellt werden.

Mit der Validierung wurde die Eignung der entwickelten Verfahren hinsichtlich der definierten Anforderungen nachgewiesen. Die Integration in eine Architektur stellt dabei das Zusammenwirken der einzelnen Komponenten sicher und ermöglicht eine automatische Adaption an die bestehenden Prozessunsicherheiten auf Basis der Interdependenzen zwischen Datenbasis und Ablaufplanung. Der Entscheidungsträger wird in die Lage versetzt, aus einer Menge pareto-optimaler Ablaufpläne den – im Hinblick auf die Servicequalität für den Kunden – geeignetsten auszuwählen und darüber hinaus, den Grad der Absicherung gegen Unsicherheiten über Parameter zu bestimmen.

7.2 Weitere Forschungsmöglichkeiten

Das in dieser Arbeit vorgestellte System zur Ablaufplanung und -steuerung ist für den Kontext von großen und abgesehen von der manuellen Auftragsbearbeitung automatisierten Luftfrachtterminals entwickelt worden. Die Ausgestaltung der einzelnen Module der Architektur durch die in diesen umgesetzten Verfahren und deren Bewertung sind von den Anforderungen eines konkreten Luftfrachtterminals abgeleitet. Eine Ausweitung der Betrachtung auf weitere vergleichbare Luftfrachtterminals und damit der weiteren

Validierung der hier vorgestellten Konzepte sowie die mögliche Erweiterung der Architektur wären interessante Aspekte für zukünftige praxisnahe Arbeiten. Wünschenswert in diesem Zusammenhang wäre insbesondere der Praxiseinsatz im Hinblick auf eine Untersuchung der langfristigen Interdependenzen zwischen ermitteltem Informationsstand, der Qualität der generierten Ablaufpläne und der sich ändernden Datenbasis.

Die Ausweitung der beschriebenen Konzepte auf weitere Branchen mit ähnlichen Ablaufplanungsproblemen ist aufgrund der einfachen Austauschbarkeit der Module und damit Verfahren innerhalb der Architektur möglich. Hierfür könnten branchenunabhängige Module aus den hier vorgestellten abgeleitet und durch branchenspezifische ergänzt werden.

Weitere Forschungsmöglichkeiten ergeben sich bei der Ermittlung des Worst-Case-Szenarios zur Bewertung eines Ablaufplans und insbesondere bei der Erstellung des optimalen Ablaufplans für dieses Szenario. Diese beiden Schritte nahmen in der robusten Ablaufplanung die meiste Zeit in Anspruch. Eine Alternative zur Evaluation der Ablaufpläne stellt eine weitergehende Approximation dieser beiden Schritte durch die Verwendung von Heuristiken oder Ersatzzielfunktionen (sogenannten Surrogaten) dar.

Literaturverzeichnis

- [AA03] ALAGÖZ, O. ; AZIZOĞLU, M.: Rescheduling of identical parallel machines under machine eligibility constraints. In: *European Journal of Operational Research* 149.3 (2003), S. 523–532.
- [AA05] AZIZOĞLU, M. ; ALAGÖZ, O.: Parallel-machine rescheduling with machine disruptions. In: *IIE Transactions* 37.12 (2005), S. 1113–1118.
- [Abd12] ABDELAZIZ, F. B.: Solution approaches for the multiobjective stochastic programming. In: *European Journal of Operational Research* 216.1 (2012), S. 1–16.
- [AD08] ALOULOU, M. A. ; DELLA CROCE, F.: Complexity of single machine scheduling problems under scenario-based uncertainty. In: *Operations Research Letters* 36.3 (2008), S. 338–342.
- [AF07] ABONYI, J. ; FEIL, B.: *Cluster analysis for data mining and system identification*. Basel: Birkhäuser Verlag AG, 2007.
- [AGGM05] ANGLANI, A. ; GRIECO, A. ; GUERRIERO, E. ; MUSMANNO, R.: Robust scheduling of parallel machines with sequence-dependent set-up costs. In: *European Journal of Operational Research* 161.3 (2005), S. 704–720.
- [AIK⁺08] ARNOLD, D. ; ISERMANN, H. ; KUHN, A. ; TEMPELMEIER, H. ; FURMANS, K. (Hrsgg.): *Handbuch Logistik*. 3. Aufl. VDI-Buch. Berlin und Heidelberg: Springer, 2008.
- [Air13] AIRBUS: *Airbus global market forecast. Future payloads, freight forecast, 2013–2032*. 2013. URL: <http://www.airbus.com/presscentre/corporate-information/key-documents/> (besucht am 22.01.2014).
- [AKW⁺09] ASSENT, I. ; KRIEGER, R. ; WELTER, P. ; HERBERS, J. ; SEIDL, T.: Data mining for robust flight scheduling. In: CAO, L. ; YU, P. S. ; ZHANG, C. ; ZHANG, H. (Hrsgg.): *Data mining for business applications*. Boston, MA: Springer US, 2009, S. 267–282.
- [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* 161.1 (2005), S. 86–110.

- [AMW11] ASHFORD, N. J. ; MUMAYIZ, S. ; WRIGHT, P. H.: Air cargo facilities. In: ASHFORD, N. J. ; MUMAYIZ, S. ; WRIGHT, P. H. (Hrsgg.): *Airport engineering. Planning, design, and development of 21st century airports*. 4. Aufl. Hoboken, NJ: John Wiley & Sons, 2011, S. 457–488.
- [And73] ANDERBERG, M. R.: *Cluster analysis for applications*. Probability and mathematical statistics 19. New York: Academic Press (Ein Imprint von Elsevier), 1973.
- [AR10] AISSI, H. ; ROY, B.: Robustness in multi-criteria decision aiding. In: EHRGOTT, M. ; FIGUEIRA, J. R. ; GRECO, S. (Hrsgg.): *Trends in multiple criteria decision analysis*. International Series in Operations Research & Management Science 142. New York: Springer, 2010, S. 87–121.
- [Arn14] ARNAOUT, J.-P.: Rescheduling of parallel machines with stochastic processing and setup times. In: *Journal of Manufacturing Systems* 33.3 (2014), S. 376–384.
- [AS08] ARCHETTI, C. ; SPERANZA, M. G.: The split delivery vehicle routing problem: a survey. In: GOLDEN, B. L. ; RAGHAVAN, S. ; WASIL, E. A. (Hrsgg.): *The vehicle routing problem. Latest advances and new challenges*. Operations Research/Computer Science Interfaces 43. Boston, MA: Springer US, 2008, S. 103–122.
- [AS1825] SAE INTERNATIONAL: *Methodology of calculating aircraft cargo volumes*. Standard. 2008.
- [AS97] ABUMAIZAR, R. J. ; SVESTKA, J. A.: Rescheduling job shops under random disruptions. In: *International Journal of Production Research* 35.7 (1997), S. 2065–2082.
- [Bal11] BALIN, S.: Non-identical parallel machine scheduling using genetic algorithm. In: *Expert Systems with Applications* 38.6 (2011), S. 6814–6821.
- [BAM13] BRANKE, J. ; AVIGAD, G. ; MOSHAIOV, A.: Multi-objective worst case optimization by means of evolutionary algorithms. Unpublished. Coventry, UK, 2013. URL: <http://wrap.warwick.ac.uk/55724/> (besucht am 20.11.2014).
- [BBB10] BEN-TAL, A. ; BERTSIMAS, D. ; BROWN, D. B.: A soft robust model for optimization under ambiguity. In: *Operations Research* 58.4-part-2 (2010), S. 1220–1234.
- [BBC11] BERTSIMAS, D. ; BROWN, D. B. ; CARAMANIS, C.: Theory and applications of robust optimization. In: *SIAM Review* 53.3 (2011), S. 464–501.
- [BBMN91] BEAN, J. C. ; BIRGE, J. R. ; MITTENTHAL, J. ; NOON, C. E.: Matchup scheduling with multiple resources, release dates and disruptions. In: *Operations Research* 39.3 (1991), S. 470–483.

- [BEN09] BEN-TAL, A. ; EL GHAOU, L. ; NEMIROVSKI, A.: *Robust optimization*. Princeton series in applied mathematics. Princeton, NJ: Princeton University Press, 2009.
- [BEP⁺07] BŁAŻEWICZ, J. ; ECKER, K. H. ; PESCH, E. ; SCHMIDT, G. ; WĘGLARZ, J.: *Handbook on scheduling. From theory to applications*. International Handbooks on Information Systems. Berlin und Heidelberg: Springer, 2007.
- [Ber06] BERKHIN, P.: A survey of clustering data mining techniques. In: KOGAN, J. ; NICHOLAS, C. ; TEBoulLE, M. (Hrsgg.): *Grouping multidimensional data. Recent advances in clustering*. Berlin und Heidelberg: Springer, 2006, S. 25–71.
- [BFOS84] BREIMAN, L. ; FRIEDMAN, J. H. ; OLSHEN, R. A. ; STONE, C. J.: *Classification and regression trees*. Belmont, CA: Wadsworth International Group, 1984.
- [BG12] BERNECKER, T. ; GRANDJOT, H.-H.: *Leitfaden Luftfracht*. 3. Aufl. München: Huss-Verlag, 2012.
- [Bid05] BIDOT, J.: *A general framework integrating techniques for scheduling under uncertainty*. Dissertation. Frankreich: Institut National Polytechnique de Toulouse, Ecole Nationale d’Ingénieurs de Tarbes, 2005.
- [Bis06] BISHOP, C. M.: *Pattern recognition and machine learning*. Information science and statistics. New York: Springer, 2006.
- [BL07] BISHOP, C. M. ; LASSERRE, J.: Generative or discriminative? Getting the best of both worlds. In: *Bayesian statistics 8* (2007), S. 3–24.
- [BL11] BIRGE, J. R. ; LOUVEAUX, F.: *Introduction to stochastic programming*. 2. Aufl. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2011.
- [BLW14] BOXNICK, S. ; LAUCK, S. ; WEBER, J.: A data mining approach to support a data-driven scheduling system for air cargo terminals. In: s. ed.: *Proceedings of the IEEE Asia-Pacific world congress on computer science and engineering*. 2014, S. 180–187.
- [BM05] BRANKE, J. ; MATTFELD, D. C.: Anticipation and flexibility in dynamic scheduling. In: *International Journal of Production Research* 43.15 (2005), S. 3103–3129.
- [BM99] BIERWIRTH, C. ; MATTFELD, D. C.: Production scheduling and rescheduling with genetic algorithms. In: *Evolutionary computation* 7.1 (1999), S. 1–17.
- [BN00] BEN-TAL, A. ; NEMIROVSKI, A.: Robust solutions of linear programming problems contaminated with uncertain data. In: *Mathematical Programming* 88.3 (2000), S. 411–424.

- [BN98] BEN-TAL, A. ; NEMIROVSKI, A.: Robust convex optimization. In: *Mathematics of Operations Research* 23.4 (1998), S. 769–805.
- [BN99] BEN-TAL, A. ; NEMIROVSKI, A.: Robust solutions of uncertain linear programs. In: *Operations Research Letters* 25.1 (1999), S. 1–13.
- [Boc08] BOCK, H.-H.: Origins and extensions of the k-means algorithm in cluster analysis. In: *Journ@l Electronique d'Histoire des Probabilités et de la Statistique* 4.2 (2008), S. 1–18.
- [Boe14] BOEING: *Boeing world air cargo forecast 2014–2015*. 2014. URL: <http://www.boeing.com/boeing/commercial/cargo/> (besucht am 21. 01. 2015).
- [Box11] BOXNICK, S.: *Eine integrierte Lern- und Entscheidungsarchitektur für selbst-adaptive Systeme in nicht-deterministischen Umgebungen*. Unveröffentlicht. Masterarbeit. Paderborn: Wirtschaftsinformatik, insbesondere CIM, Heinz Nixdorf Institut, Universität Paderborn, 2011.
- [BPW10] BACHER, J. ; PÖGE, A. ; WENZIG, K.: *Clusteranalyse. Anwendungsorientierte Einführung in Klassifikationsverfahren*. 3. Aufl. München: Oldenbourg, 2010.
- [Bri90] BRIDLE, J. S.: Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: SOULIÉ, F. F. ; HÉRAULT, J. (Hrsgg.): *Neurocomputing. Algorithms, architectures and applications*. NATO ASI Series (Series F: Computer and Systems Sciences) 68. Berlin und Heidelberg: Springer, 1990, S. 227–236.
- [Bru07] BRUCKER, P.: *Scheduling algorithms*. 5. Aufl. Berlin und Heidelberg: Springer, 2007.
- [BS03] BERTSIMAS, D. ; SIM, M.: Robust discrete optimization and network flows. In: *Mathematical Programming* 98.1-3 (2003), S. 49–71.
- [BS04] BERTSIMAS, D. ; SIM, M.: The price of robustness. In: *Operations Research* 52.1 (2004), S. 35–53.
- [BS07] BEYER, H.-G. ; SENDHOFF, B.: Robust optimization – a comprehensive survey. In: *Computer Methods in Applied Mechanics and Engineering* 196.33–34 (2007), S. 3190–3218.
- [BSG04] BOKHORST, J. A. C. ; SLOMP, J. ; GAALMAN, G. J. C.: On the who-rule in dual resource constrained (DRC) manufacturing systems. In: *International Journal of Production Research* 42.23 (2004), S. 5049–5074.
- [Bul86] BULLINGER, H.-J. (Hrsg.): *Systematische Montageplanung. Handbuch für die Praxis*. München: Carl Hanser Verlag, 1986.
- [BVLB09] BIDOT, J. ; VIDAL, T. ; LABORIE, P. ; BECK, J. C.: A theoretic and practical framework for scheduling in a stochastic environment. In: *Journal of Scheduling* 12.3 (2009), S. 315–344.

- [CC59] CHARNES, A. ; COOPER, W. W.: Chance-constrained programming. In: *Management Science* 6.1 (1959), S. 73–79.
- [CCF13] COSTA, A. ; CAPPADONNA, F. A. ; FICHERA, S.: A hybrid genetic algorithm for job sequencing and worker allocation in parallel unrelated machines with sequence-dependent setup times. In: *The International Journal of Advanced Manufacturing Technology* 69.9-12 (2013), S. 2799–2817.
- [CCS58] CHARNES, A. ; COOPER, W. W. ; SYMONDS, G. H.: Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. In: *Management Science* 4.3 (1958), S. 235–263.
- [CD09] CHAUDHRY, I. A. ; DRAKE, P. R.: Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. In: *The International Journal of Advanced Manufacturing Technology* 42.5-6 (2009), S. 581–594.
- [CFR99] CORRÊA, R. C. ; FERREIRA, A. ; REBREYEND, P.: Scheduling multiprocessor tasks with genetic algorithms. In: *IEEE Transactions on Parallel and Distributed Systems* 10.8 (1999), S. 825–837.
- [CGT95] CHENG, R. ; GEN, M. ; TSUJIMURA, Y.: Minmax earliness/tardiness scheduling in identical parallel machine system using genetic algorithms. In: *Computers & Industrial Engineering* 29.1–4 (1995), S. 513–517.
- [CGT96] CHENG, R. ; GEN, M. ; TSUJIMURA, Y.: A tutorial survey of job-shop scheduling problems using genetic algorithms—I: representation. In: *Computers and Industrial Engineering* 30.4 (1996), S. 983–997.
- [CKY08] CARUANA, R. ; KARAMPATZIAKIS, N. ; YESSENALINA, A.: An empirical evaluation of supervised learning in high dimensions. In: MCCALLUM, A. ; ROWEIS, S. (Hrsgg.): *Proceedings of the twenty-fifth international conference on machine learning*. New York: ACM, 2008, S. 96–103.
- [CLM06] CHEUNG, R. K. ; LEE, A. ; MO, D.: Flow diversion approaches for shipment routing in automatic shipment handling systems. In: s. ed.: *Proceedings 2006 IEEE international conference on robotics and automation*. S. l.: IEEE, 2006, S. 695–700.
- [CLV07] COELLO COELLO, C. A. ; LAMONT, G. B. ; VELDHUIZEN, D. A. van: *Evolutionary algorithms for solving multi-objective problems*. 2. Aufl. Genetic and Evolutionary Computation. New York: Springer, 2007.
- [CM07] ÇAKICI, E. ; MASON, S. J.: Parallel machine scheduling subject to auxiliary resource constraints. In: *Production Planning & Control* 18.3 (2007), S. 217–225.

- [CN06] CARUANA, R. ; NICULESCU-MIZIL, A.: An empirical comparison of supervised learning algorithms. In: COHEN, W. W. ; MOORE, A. (Hrsgg.): *Proceedings of the twenty-third international conference on machine learning*. New York: ACM, 2006, S. 161–168.
- [Coe06] COELLO COELLO, C. A.: Evolutionary multi-objective optimization: a historical view of the field. In: *IEEE Computational Intelligence Magazine* 1.1 (2006), S. 28–36.
- [Coe14] COELLO COELLO, C. A.: *List of references on evolutionary multiobjective optimization*. 2014. URL: <http://delta.cs.cinvestav.mx/~ccoello/EMOO/> (besucht am 20.11.2014).
- [Coh60] COHEN, J.: A coefficient of agreement for nominal scales. In: *Educational and Psychological Measurement* 20.1 (1960), S. 37–46.
- [CU92] CHURCH, L. K. ; UZSOY, R.: Analysis of periodic and event-driven rescheduling policies in dynamic shops. In: *International Journal of Computer Integrated Manufacturing* 5.3 (1992), S. 153–163.
- [Dan55] DANTZIG, G. B.: Linear programming under uncertainty. In: *Management Science* 1.3–4 (1955), S. 197–206.
- [DC97] DANIELS, R. L. ; CARRILLO, J. E.: β -Robust scheduling for single-machine systems with uncertain processing times. In: *IIE Transactions* 29.11 (1997), S. 977–985.
- [DD11] DOMSCHKE, W. ; DREXL, A.: *Einführung in Operations Research*. 8. Aufl. Berlin: Springer, 2011.
- [Deb00] DEB, K.: An efficient constraint handling method for genetic algorithms. In: *Computer Methods in Applied Mechanics and Engineering* 186.2–4 (2000), S. 311–338.
- [Deb01] DEB, K.: *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience series in systems and optimization. Chichester et al.: Wiley, 2001.
- [Def77] DEFAYS, D.: An efficient algorithm for a complete link method. In: *The Computer Journal* 20.4 (1977), S. 364–366.
- [DG06] DEB, K. ; GUPTA, H.: Introducing robustness in multi-objective optimization. In: *Evolutionary computation* 14.4 (2006), S. 463–494.
- [DIN 30781] DEUTSCHES INSTITUT FÜR NORMUNG (DIN): *DIN 30781-1 (1989-05-00): Transportkette; Grundbegriffe*. Deutsche Norm. 1989.

- [DL02] DASGUPTA, S. ; LONG, P. M.: Performance guarantees for hierarchical clustering. In: KIVINEN, J. ; SLOAN, R. H. (Hrsgg.): *Computational learning theory. 15th annual conference on computational learning theory, COLT 2002, Sydney, Australia, July 8-10, 2002. Proceedings*. Lecture Notes in Artificial Intelligence 2375. Berlin und Heidelberg: Springer, 2002, S. 351–363.
- [DL05] DASGUPTA, S. ; LONG, P. M.: Performance guarantees for hierarchical clustering. In: *Journal of Computer and System Sciences* 70.4 (2005), S. 555–569.
- [DP96] DOMINGOS, P. ; PAZZANI, M.: Beyond independence: conditions for the optimality of the simple Bayesian classifier. In: SAITTA, L. (Hrsg.): *Proceedings of the thirteenth international conference on machine learning*. San Francisco, CA: Morgan Kaufmann (Ein Imprint von Elsevier), 1996, S. 105–112.
- [DP97] DOMINGOS, P. ; PAZZANI, M.: On the optimality of the simple Bayesian classifier under zero-one loss. In: *Machine Learning* 29.2–3 (1997), S. 103–130.
- [DPAM02] DEB, K. ; PRATAP, A. ; AGARWAL, S. ; MEYARIVAN, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), S. 182–197.
- [Ehr05] EHRGOTT, M.: *Multicriteria optimization*. 2. Aufl. Berlin, Heidelberg und New York: Springer, 2005.
- [EIS14] EHRGOTT, M. ; IDE, J. ; SCHÖBEL, A.: Minmax robustness for multiobjective optimization problems. In: *European Journal of Operational Research* 239.1 (2014), S. 17–31.
- [EKX95] ESTER, M. ; KRIEGEL, H.-P. ; XU, X.: Knowledge discovery in large spatial databases: focusing techniques for efficient class identification. In: EGENHOFER, M. J. ; HERRING, J. R. (Hrsgg.): *Advances in spatial databases. 4th international symposium, SSD '95, Portland, ME, USA, August 6 - 9, 1995. Proceedings*. Lecture Notes in Computer Science 951. Berlin und Heidelberg: Springer, 1995, S. 67–82.
- [EL97] EL GHAOU, L. ; LEBRET, H.: Robust solutions to least-squares problems with uncertain data. In: *SIAM Journal on Matrix Analysis and Applications* 18.4 (1997), S. 1035–1064.
- [EOL98] EL GHAOU, L. ; OUSTRY, F. ; LEBRET, H.: Robust solutions to uncertain semidefinite programs. In: *SIAM Journal on Optimization* 9.1 (1998), S. 33–52.
- [EPA00] ELMARAGHY, H. ; PATEL, V. ; ABDALLAH, I. B.: Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. In: *Journal of Manufacturing Systems* 19.3 (2000), S. 186–201.

- [ES00] ESTER, M. ; SANDER, J.: *Knowledge discovery in databases. Techniken und Anwendungen*. Berlin und Heidelberg: Springer, 2000.
- [FM09] FISCHETTI, M. ; MONACI, M.: Light robustness. In: AHUJA, R. K. ; MÖHRING, R. H. ; ZAROLIAGIS, C. D. (Hrsgg.): *Robust and online large-scale optimization. Models and techniques for transportation systems*. Lecture Notes in Computer Science 5868. Berlin und Heidelberg: Springer, 2009, S. 61–84.
- [FPS96a] FAYYAD, U. ; PIATETSKY-SHAPIRO, G. ; SMYTH, P.: From data mining to knowledge discovery in databases. In: *AI Magazine* 17.3 (1996), S. 37–54.
- [FPS96b] FAYYAD, U. M. ; PIATETSKY-SHAPIRO, G. ; SMYTH, P.: From data mining to knowledge discovery: an overview. In: FAYYAD, U. M. ; PIATETSKY-SHAPIRO, G. ; SMYTH, P. ; UTHURUSAMY, R. (Hrsgg.): *Advances in knowledge discovery and data mining*. Menlo Park, CA: American Association for Artificial Intelligence, 1996, S. 1–34.
- [FPS96c] FAYYAD, U. ; PIATETSKY-SHAPIRO, G. ; SMYTH, P.: The KDD process for extracting useful knowledge from volumes of data. In: *Communications of the ACM* 39.11 (1996), S. 27–34.
- [FR02] FRALEY, C. ; RAFTERY, A. E.: Model-based clustering, discriminant analysis, and density estimation. In: *Journal of the American Statistical Association* 97.458 (2002), S. 611–631.
- [Fri10] FRIEDERICHS, S.: *Air cargo schedule planning. Integrated models and solution procedures*. Inauguraldissertation. Köln: Universität zu Köln, 2010.
- [Fry03] FRYE, H.: Luftfracht im Wandel. Wie verändern neue Netz- und Hubstrategien das Verhältnis von Speditionen, Airlines, Airports und Abfertigern?. In: VEREIN DEUTSCHER INGENIEURE (Hrsg.): *Luftfracht im Wandel. 8. Jahrestagung Airport Logistics*. VDI-Berichte 1801. Düsseldorf: VDI Verlag, 2003, S. 7–16.
- [Fry11] FRYE, H.: *Flächenbezogene Optimierung von Luftfrachtterminals*. Dissertation. Hrsg. von JÜNEMANN, R. 2. Aufl. Logistik für die Praxis. Dortmund: Verlag Praxiswissen, 2011.
- [Fry13] FRYE, H.: Luftfrachtverkehr. In: CLAUSEN, U. ; GEIGER, C. (Hrsgg.): *Verkehrs- und Transportlogistik*. 2. Aufl. VDI-Buch. Berlin und Heidelberg: Springer Vieweg, 2013, S. 217–251.
- [GD96] GARGEYA, V. B. ; DEANE, R. H.: Scheduling research in multiple resource constrained job shops: a review and critique. In: *International Journal of Production Research* 34.8 (1996), S. 2077–2097.
- [Geb09] GEBEL, M.: *Multivariate calibration of classifier scores into the probability space*. Dissertation. Dortmund: Technische Universität Dortmund, 2009.

- [GF13] GASS, S. I. ; FU, M. C.: *Encyclopedia of operations research and management science*. 3. Aufl. Boston, MA: Springer US, 2013.
- [GJ79] GAREY, M. R. ; JOHNSON, D. S.: *Computers and intractability. A guide to the theory of NP-completeness*. A series of books in the mathematical sciences. New York: Freeman, 1979.
- [GLLR79] GRAHAM, R. L. ; LAWLER, E. L. ; LENSTRA, J. K. ; RINNOOY KAN, A. H. G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: HAMMER, P. L. ; JOHNSON, E. L. ; KORTE, B. H. (Hrsgg.): *Discrete optimization II*. Annals of Discrete Mathematics 5. Amsterdam: North-Holland (Ein Imprint von Elsevier), 1979, S. 287–326.
- [GMT14] GABREL, V. ; MURAT, C. ; THIELE, A.: Recent advances in robust optimization: an overview. In: *European Journal of Operational Research* 235.3 (2014), S. 471–483.
- [GMW07] GAN, G. ; MA, C. ; WU, J.: *Data clustering. Theory, algorithms, and applications*. ASA-SIAM series on statistics and applied probability 20. Philadelphia: Society for Industrial und Applied Mathematics, 2007.
- [GMW13] GABREL, V. ; MURAT, C. ; WU, L.: New models for the robust shortest path problem: complexity, resolution and generalization. In: *Annals of Operations Research* 207.1 (2013), S. 97–120.
- [Goh07] GOH, C. K.: *Evolutionary multi-objective optimization in uncertain environments*. PhD thesis. Singapur: Department of Electrical & Computer Engineering, National University of Singapore, 2007. URL: <http://www.scholarbank.nus.edu.sg/handle/10635/13352>.
- [Gol89] GOLDBERG, D. E.: *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley, 1989.
- [Gow71] GOWER, J. C.: A general coefficient of similarity and some of its properties. In: *Biometrics* 27.4 (1971), S. 857–871.
- [GP10] GENDREAU, M. ; POTVIN, J.-Y.: *Handbook of metaheuristics*. 2. Aufl. International Series in Operations Research & Management Science 146. New York: Springer, 2010.
- [GRR07] GRANDJOT, H.-H. ; ROESSLER, I. ; ROLAND, A.: *Air cargo guide. An introduction to the air cargo industry*. Best practice. München: Huss-Verlag, 2007.
- [GS08] GOREN, S. ; SABUNCUOGLU, I.: Robustness and stability measures for scheduling: single-machine environment. In: *IIE Transactions* 40.1 (2008), S. 66–83.
- [GS15] GOERIGK, M. ; SCHÖBEL, A.: *Algorithm engineering in robust optimization*. 2015. arXiv: 1505.04901v2 [math.OC].

- [GT09] GOH, C.-K. ; TAN, K. C.: *Evolutionary multi-objective optimization in uncertain environments. Issues and algorithms*. Studies in Computational Intelligence 186. Berlin und Heidelberg: Springer, 2009.
- [HAR94] HOU, E. S. H. ; ANSARI, N. ; REN, H.: A genetic algorithm for multi-processor scheduling. In: *IEEE Transactions on Parallel and Distributed Systems* 5.2 (1994), S. 113–120.
- [HBV01] HALKIDI, M. ; BATISTAKIS, Y. ; VAZIRGIANNIS, M.: On clustering validation techniques. In: *Journal of Intelligent Information Systems* 17.2/3 (2001), S. 107–145.
- [Her99] HERRMANN, J. W.: A genetic algorithm for minimax optimization problems. In: s. ed.: *Proceedings of the 1999 congress on evolutionary computation*. 1999, S. 1099–1103.
- [HKKR99] HÖPPNER, F. ; KLAWONN, F. ; KRUSE, R. ; RUNKLER, T.: *Fuzzy cluster analysis. Methods for classification, data analysis and image recognition*. Chichester et al.: John Wiley & Sons, 1999.
- [HKP12] HAN, J. ; KAMBER, M. ; PEI, J.: *Data mining. Concepts and techniques*. 3. Aufl. The Morgan Kaufmann Series in Data Management Systems. Amsterdam et al.: Morgan Kaufmann (Ein Imprint von Elsevier), 2012.
- [HMS01] HAND, D. J. ; MANNILA, H. ; SMYTH, P.: *Principles of data mining*. Cambridge, MA: MIT Press, 2001.
- [Hol75] HOLLAND, J. H.: *Adaptation in natural and artificial systems. An introductory analysis with applications to biology, control, and artificial intelligence*. Neuauflage von MIT Press 1992 und 1994. Ann Arbor, MI: University of Michigan Press, 1975.
- [HS85] HOCHBAUM, D. S. ; SHMOYS, D. B.: A best possible heuristic for the k -center problem. In: *Mathematics of Operations Research* 10.2 (1985), S. 180–184.
- [HTF09] HASTIE, T. ; TIBSHIRANI, R. ; FRIEDMAN, J.: *The elements of statistical learning. Data mining, inference, and prediction*. 2. Aufl. Springer Series in Statistics. New York: Springer, 2009.
- [Hua98] HUANG, Z.: Extensions to the k -means algorithm for clustering large data sets with categorical values. In: *Data Mining and Knowledge Discovery* 2.3 (1998), S. 283–304.
- [HY01] HAND, D. J. ; YU, K.: Idiot’s Bayes—not so stupid after all?. In: *International Statistical Review* 69.3 (2001), S. 385–398.
- [IATA2014] INTERNATIONAL AIR TRANSPORT ASSOCIATION (IATA): *IATA ULD regulations*. Handbuch. 2014.

- [IK14] IDE, J. ; KÖBIS, E.: Concepts of efficiency for uncertain multi-objective optimization problems based on set order relations. In: *Mathematical Methods of Operations Research* 80.1 (2014), S. 99–127.
- [IS15] IDE, J. ; SCHÖBEL, A.: Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. In: *OR Spectrum* (2015), S. 1–37.
- [IT14] IANCU, D. A. ; TRICHAKIS, N.: Pareto efficiency in robust optimization. In: *Management Science* 60.1 (2014), S. 130–147.
- [Jac12] JACCARD, P.: The distribution of the flora in the alpine zone. In: *New Phytologist* 11.2 (1912), S. 37–50. Übersetzung des französischen Originals: JACCARD, Paul: La distribution de la flore dans la zone alpine. In: *Revue générale des Sciences*, 15.12.1907, S. 961–967.
- [Jai10] JAIN, A. K.: Data clustering: 50 years beyond k-means. In: *Pattern Recognition Letters* 31.8 (2010), S. 651–666.
- [JB05] JIN, Y. ; BRANKE, J.: Evolutionary optimization in uncertain environments —a survey. In: *IEEE Transactions on Evolutionary Computation* 9.3 (2005), S. 303–317.
- [JD88] JAIN, A. K. ; DUBES, R. C.: *Algorithms for clustering data*. Prentice Hall advanced reference series : Computer science. Englewood Cliffs, NJ: Prentice Hall, 1988.
- [Jen01] JENSEN, M. T.: *Robust and flexible scheduling with evolutionary computation*. PhD Dissertation. Dänemark: Department of Computer Science, University of Aarhus, 2001.
- [JGL12] JOE, Y. Y. ; GAN, O. P. ; LEWIS, F. L.: Multi-commodity flow dynamic resource assignment and matrix-based job dispatching for multi-relay transfer in complex material handling systems (MHS). In: *Journal of Intelligent Manufacturing* 25.4 (2012), S. 681–697.
- [JI07] JIA, Z. ; IERAPETRITOU, M. G.: Generate pareto optimal solutions of scheduling problems using normal boundary intersection technique. In: *Computers & Chemical Engineering* 31.4 (2007), S. 268–280.
- [JL95] JOHN, G. H. ; LANGLEY, P.: Estimating continuous distributions in Bayesian classifiers. In: BESNARD, P. ; HANKS, S. (Hrsgg.): *Proceedings of the eleventh conference on uncertainty in artificial intelligence*. San Francisco, CA: Morgan Kaufmann (Ein Imprint von Elsevier), 1995, S. 338–345.
- [JMF99] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data clustering: a review. In: *ACM Computing Surveys* 31.3 (1999), S. 264–323.
- [JN07] JENSEN, F. V. ; NIELSEN, T. D.: *Bayesian networks and decision graphs*. 2. Aufl. Information science and statistics. New York: Springer, 2007.

- [JP14] JAEHN, F. ; PESCH, E.: *Ablaufplanung. Einführung in Scheduling*. Wiesbaden: Springer Gabler, 2014.
- [JS00] JÜNEMANN, R. ; SCHMIDT, T.: *Materialflußsysteme. Systemtechnische Grundlagen*. 2. Aufl. Bd. 2. Logistik in Industrie, Handel und Dienstleistungen. Berlin und Heidelberg: Springer, 2000.
- [Kas04] KASPERSKI, A.: Minimizing maximal regret in sequencing problems with interval parameters. In: ATANASSOV, K. T. ; HRYNIEWICZ, O. ; KACPRZYK, J. (Hrsgg.): *Soft computing. Foundations and theoretical aspects*. Problemy współczesnej nauki, teoria i zastosowania - Informatyka. Warszawa: Exit, 2004, S. 277–290.
- [Kas08] KASPERSKI, A.: *Discrete optimization with interval data. Minmax regret and fuzzy approach*. Studies in Fuzziness and Soft Computing 228. Berlin und Heidelberg: Springer, 2008.
- [KC93] KAHLBACHER, H. G. ; CHENG, T. C. E.: Parallel machine scheduling to minimize costs for earliness and number of tardy jobs. In: *Discrete Applied Mathematics* 47.2 (1993), S. 139–164.
- [KCS06] KONAK, A. ; COIT, D. W. ; SMITH, A. E.: Multi-objective optimization using genetic algorithms: a tutorial. In: *Reliability Engineering & System Safety. Special Issue - Genetic Algorithms and Reliability* 91.9 (2006), S. 992–1007.
- [KJF10] KUSTER, J. ; JANNACH, D. ; FRIEDRICH, G.: Applying local rescheduling in response to schedule disruptions. In: *Annals of Operations Research* 180.1 (2010), S. 265–282.
- [KKK12] KLAUS, P. ; KRIEGER, W. ; KRUPP, M. (Hrsgg.): *Gabler Lexikon Logistik. Management logistischer Netzwerke und Flüsse*. 5. Aufl. Wiesbaden: Springer Gabler, 2012.
- [KL07] KALAÏ, R. ; LAMBORAY, C.: L' α -robustesse lexicographique : une relaxation de la β -robustesse. In: ROY, B. ; ALOULOU, M. A. ; KALAÏ, R. (Hrsgg.): *Robustness in OR-DA*. Annales du Lamsade 7. Paris: Lamsade, 2007, S. 129–143.
- [KLR06] KHAYAT, G. E. ; LANGEVIN, A. ; RIOPEL, D.: Integrated production and material handling scheduling using mathematical programming and constraint programming. In: *European Journal of Operational Research* 175.3 (2006), S. 1818–1832.
- [KLV12] KALAÏ, R. ; LAMBORAY, C. ; VANDERPOOTEN, D.: Lexicographic α -robustness: an alternative to min–max criteria. In: *European Journal of Operational Research* 220.3 (2012), S. 722–728.
- [KM06] KURGAN, L. A. ; MUSILEK, P.: A survey of knowledge discovery and data mining process models. In: *The Knowledge Engineering Review* 21.1 (2006), S. 1–24.

- [Koe04] KOETHER, R. (Hrsg.): *Taschenbuch der Logistik*. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2004.
- [Kot07] KOTSIANTIS, S. B.: Supervised machine learning: a review of classification techniques. In: *Informatica* 31.3 (2007), S. 249–268.
- [KP96] KAUFMANN, H. ; PAPE, H.: Clusteranalyse. In: FAHRMEIR, L. ; HAMERLE, A. ; TUTZ, G. (Hrsgg.): *Multivariate statistische Verfahren*. 2. Aufl. Berlin und New York: de Gruyter, 1996, S. 437–536.
- [KR86] KAUFMAN, L. ; ROUSSEEUW, P. J.: Clustering large data sets. In: KANAL, L. N. ; GELSEMA, E. S. (Hrsgg.): *Pattern recognition in practice II*. Amsterdam: North-Holland (Ein Imprint von Elsevier), 1986, S. 425–437.
- [KR87] KAUFMAN, L. ; ROUSSEEUW, P. J.: Clustering by means of medoids. In: DODGE, Y. (Hrsg.): *Statistical data analysis: based on the L1-Norm and related methods*. Amsterdam: North-Holland (Ein Imprint von Elsevier), 1987, S. 405–416.
- [KR90] KAUFMAN, L. ; ROUSSEEUW, P. J.: *Finding groups in data. An introduction to cluster analysis*. Hoboken, NJ: John Wiley & Sons, 1990.
- [KY97] KOUVELIS, P. ; YU, G.: *Robust discrete optimization and its applications*. Nonconvex Optimization and Its Applications 14. Dordrecht, Boston und London: Kluwer Academic Publishers, 1997.
- [KZP06] KOTSIANTIS, S. B. ; ZAHARAKIS, I. D. ; PINTELAS, P. E.: Machine learning: a review of classification and combining techniques. In: *Artificial Intelligence Review* 26.3 (2006), S. 159–190.
- [Lar05] LAROSE, D. T.: *Discovering knowledge in data. An introduction to data mining*. Hoboken, NJ: John Wiley & Sons, 2005.
- [Les04] LESSING, H.: *Prozess zur multivariaten Prognose von Produktionsprogrammen für eine effiziente mehrperiodische Kapazitätsplanung bei typisierten Dienstleistungen*. HNI-Verlagsschriftenreihe 146. Zugl.: Paderborn, Univ., Diss., 2004. Paderborn: Heinz-Nixdorf-Institut, 2004.
- [Leu04] LEUNG, J. Y.-T.: Introduction and notation. In: LEUNG, J. Y.-T. (Hrsg.): *Handbook of scheduling. Algorithms, models, and performance analysis*. Chapman & Hall/CRC Computer and Information Science Series. Boca Raton, FL et al.: Chapman & Hall/CRC, 2004, S. 1-1–1-9.
- [LHLX06] LEE, C. ; HUANG, H. C. ; LIU, B. ; XU, Z.: Development of timed colour petri net simulation models for air cargo terminal operations. In: *Computers & Industrial Engineering* 51.1 (2006), S. 102–110.
- [LI08] LI, Z. ; IERAPETRITOU, M.: Process scheduling under uncertainty: review and challenges. In: *Computers & Chemical Engineering* 32.4–5 (2008), S. 715–727.

- [Llo57] LLOYD, S. P.: *Least squares quantization in PCM*. Techn. Ber. Bell Telephone Laboratories Paper. Murray Hill, NJ: Bell Telephone Laboratories, 1957.
- [Llo82] LLOYD, S. P.: Least squares quantization in PCM. In: *IEEE Transactions on Information Theory* 28.2 (1982), S. 129–137.
- [LLX⁺10] LIU, Y. ; LI, Z. ; XIONG, H. ; GAO, X. ; WU, J.: Understanding of internal clustering validation measures. In: WEBB, G. I. ; LIU, B. ; ZHANG, C. ; GUNOPULOS, D. ; WU, X. (Hrsgg.): *10th IEEE international conference on data mining*. 2010, S. 911–916.
- [LLY06] LEE, C.-Y. ; LEUNG, J. Y.-T. ; YU, G.: Two machine scheduling under disruptions with transportation considerations. In: *Journal of Scheduling* 9.1 (2006), S. 35–48.
- [LZ06] LAU, H. Y. ; ZHAO, Y.: Joint scheduling of material handling equipment in automated air cargo terminals. In: *Computers in Industry* 57.5 (2006), S. 398–411.
- [Mac67] MACQUEEN, J.: Some methods for classification and analysis of multivariate observations. In: LE CAM, L. M. ; NEYMAN, J. (Hrsgg.): *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Berkeley, CA: University of California Press, 1967, S. 281–297.
- [MCLL09] MO, D. Y. ; CHEUNG, R. K. ; LEE, A. W. ; LAW, G. K.: Flow diversion strategies for routing in integrated automatic shipment handling systems. In: *IEEE Transactions on Automation Science and Engineering* 6.2 (2009), S. 377–384.
- [Men13] MENSEN, H.: *Handbuch der Luftfahrt*. 2. Aufl. VDI-Buch. Berlin und Heidelberg: Springer Vieweg, 2013.
- [Mit05] MITCHELL, T. M.: *Generative and discriminative classifiers. Naive Bayes and logistic regression*. Kapitel Entwurf für 2. Aufl. von *Machine Learning*. 2005. URL: <http://www.cs.cmu.edu/~tom/NewChapters.html> (besucht am 15.07.2014).
- [Mit97] MITCHELL, T. M.: *Machine learning*. McGraw-Hill series in computer science. New York: McGraw-Hill, 1997.
- [ML13] MEHRAVARAN, Y. ; LOGENDRAN, R.: Non-permutation flowshop scheduling with dual resources. In: *Expert Systems with Applications* 40.13 (2013), S. 5061–5076.
- [MS96] MICHALEWICZ, Z. ; SCHOENAUER, M.: Evolutionary algorithms for constrained parameter optimization problems. In: *Evolutionary computation* 4.1 (1996), S. 1–32.

- [MWDM64] MACNAUGHTON-SMITH, P. ; WILLIAMS, W. T. ; DALE, M. B. ; MOCKETT, L. G.: Dissimilarity analysis: a new technique of hierarchical sub-division. In: *Nature* 202.4936 (1964), S. 1034–1035.
- [NC05a] NICULESCU-MIZIL, A. ; CARUANA, R.: Obtaining calibrated probabilities from boosting. In: BACCHUS, F. ; JAAKKOLA, T. (Hrsgg.): *Proceedings of the twenty-first conference on uncertainty in artificial intelligence*. Arlington, VA: AUAI Press, 2005, S. 28–33.
- [NC05b] NICULESCU-MIZIL, A. ; CARUANA, R.: Predicting good probabilities with supervised learning. In: DE RAEDT, L. ; WROBEL, S. (Hrsgg.): *Proceedings of the twenty-second international conference on machine learning*. New York: ACM, 2005, S. 625–632.
- [NH02] NG, R. T. ; HAN, J.: CLARANS: a method for clustering objects for spatial data mining. In: *IEEE Transactions on Knowledge and Data Engineering* 14.5 (2002), S. 1003–1016.
- [NH94] NG, R. T. ; HAN, J.: Efficient and effective clustering methods for spatial data mining. In: BOCCA, J. B. ; JARKE, M. ; ZANIOLO, C. (Hrsgg.): *Proceedings of the 20th international conference on very large data bases*. San Francisco, CA: Morgan Kaufmann (Ein Imprint von Elsevier), 1994, S. 144–155.
- [NJ02] NG, A. Y. ; JORDAN, M. I.: On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: DIETTERICH, T. G. ; BECKER, S. ; GHAHRAMANI, Z. (Hrsgg.): *Advances in neural information processing systems 14*. Cambridge, MA: MIT Press, 2002, S. 841–848.
- [NR98] NOBERT, Y. ; ROY, J.: Freight handling personnel scheduling at air cargo terminals. In: *Transportation Science* 32.3 (1998), S. 295–301.
- [NTD04] NSAKANDA, A. L. ; TURCOTTE, M. ; DIABY, M.: Air cargo operations evaluation and analysis through simulation. In: INGALLS, R. G. ; ROSSETTI, M. D. ; SMITH, J. S. ; PETERS, B. A. (Hrsgg.): *Proceedings of the 2004 winter simulation conference*. 2004, S. 711–719.
- [OA09] OZLEN, M. ; AZIZOĞLU, M.: Generating all efficient solutions of a rescheduling problem on unrelated parallel machines. In: *International Journal of Production Research* 47.19 (2009), S. 5245–5270.
- [OA11] OZLEN, M. ; AZIZOĞLU, M.: Rescheduling unrelated parallel machines with total flow time and total disruption cost criteria. In: *Journal of the Operational Research Society* 62.1 (2011), S. 152–164.
- [OP09] OUELHADJ, D. ; PETROVIC, S.: A survey of dynamic scheduling in manufacturing systems. In: *Journal of Scheduling* 12.4 (2009), S. 417–431.

- [Pet05] PETERSOHN, H.: *Data mining. Verfahren, Prozesse, Anwendungsarchitektur*. München: Oldenbourg, 2005.
- [Pet07] PETERSON, J.: *Air freight industry – white paper*. Techn. Ber. The Supply Chain and Logistics Institute, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, 2007.
- [Pia07] PIATETSKY-SHAPIRO, G.: Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from “university” to “business” and “analytics”. In: *Data Mining and Knowledge Discovery* 15.1 (2007), S. 99–105.
- [Pin12] PINEDO, M. L.: *Scheduling. Theory, algorithms, and systems*. 4. Aufl. New York et al.: Springer, 2012.
- [PJ09] PARK, H.-S. ; JUN, C.-H.: A simple and fast algorithm for k-medoids clustering. In: *Expert Systems with Applications* 36.2 (2009), S. 3336–3341.
- [PKM11] POPESCU, A. ; KESKINOCAK, P. ; MUTAWALY, I. al: The air cargo industry. In: HOEL, L. A. ; GIULIANO, G. ; MEYER, M. D. (Hrsgg.): *Intermodal transportation. Moving freight in a global economy*. Washington, D.C.: Eno Transportation Foundation, 2011, S. 209–237.
- [Pla00] PLATT, J. C.: Probabilities for SV Machines. In: SMOLA, A. J. ; BARTLETT, P. ; SCHÖLKOPF, B. ; SCHUURMANS, D. (Hrsgg.): *Advances in large margin classifiers*. Neural Information Processing series. Cambridge und London: MIT Press, 2000, S. 61–74. Ebenfalls bekannt unter dem Titel „Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods“.
- [QBY06] QI, X. ; BARD, J. F. ; YU, G.: Disruption management for machine scheduling: the case of SPT schedules. In: *International Journal of Production Economics* 103.1 (2006), S. 166–184.
- [Qui86] QUINLAN, J. R.: Induction of decision trees. In: *Machine Learning* 1.1 (1986), S. 81–106.
- [Qui93] QUINLAN, J. R.: *C4.5: programs for machine learning*. San Mateo, CA: Morgan Kaufmann (Ein Imprint von Elsevier), 1993.
- [RAAQ11] RENDÓN, E. ; ABUNDEZ, I. ; ARIZMENDI, A. ; QUIROZ, E. M.: Internal versus external cluster validation indexes. In: *International journal of computers and communications* 5.1 (2011), S. 27–34.
- [RAK07] ROY, B. ; ALOULOU, M. A. ; KALAI, R. (Hrsgg.): *Robustness in OR-DA*. Annales du Lamsade 7. Paris: Lamsade, 2007.
- [Ran71] RAND, W. M.: Objective criteria for the evaluation of clustering methods. In: *Journal of the American Statistical Association* 66.336 (1971), S. 846–850.

- [RCB10] RUSHTON, A. ; CROUCHER, P. ; BAKER, P.: *The handbook of logistics & distribution management*. 4. Aufl. London, Philadelphia und Daryaganj: Kogan Page, 2010.
- [RDL12] RANJBAR, M. ; DAVARI, M. ; LEUS, R.: Two branch-and-bound algorithms for the robust parallel machine scheduling problem. In: *Computers & Operations Research* 39.7 (2012), S. 1652–1660.
- [RG09] RONG, A. ; GRUNOW, M.: Shift designs for freight handling personnel at air cargo terminals. In: *Transportation Research Part E: Logistics and Transportation Review* 45.5 (2009), S. 725–739.
- [Ris01] RISH, I.: An empirical study of the naive Bayes classifier. In: s. ed.: *Proceedings of the 17th international joint conference on artificial intelligence workshop on empirical methods in artificial intelligence*. 2001, S. 41–46. Auch: IBM Technical Report RC22230.
- [RM08] ROKACH, L. ; MAIMON, O.: *Data mining with decision trees. Theory and applications*. Series in machine perception and artificial intelligence 69. Singapur: World Scientific, 2008.
- [RN10] RUSSELL, S. ; NORVIG, P.: *Artificial intelligence. A modern approach*. 3. Aufl. Prentice-Hall series in artificial intelligence. Upper Saddle River, NJ: Prentice Hall (Ein Imprint von Pearson), 2010.
- [Rok10] ROKACH, L.: A survey of clustering algorithms. In: MAIMON, O. ; ROKACH, L. (Hrsgg.): *Data mining and knowledge discovery handbook*. 2. Aufl. New York: Springer, 2010, S. 269–298.
- [Rou87] ROUSSEUW, P. J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. In: *Journal of Computational and Applied Mathematics* 20 (1987), S. 53–65.
- [Roy07] ROY, B.: La robustesse en recherche opérationnelle et aide à la décision : Une préoccupation multi facettes. In: ROY, B. ; ALOULOU, M. A. ; KALAI, R. (Hrsgg.): *Robustness in OR-DA*. Annales du Lamsade 7. Paris: Lamsade, 2007, S. 209–235.
- [Roy10] ROY, B.: Robustness in operational research and decision aiding: a multifaceted issue. In: *European Journal of Operational Research* 200.3 (2010), S. 629–638.
- [Sal13] SALES, M.: *The air logistics handbook. Air freight and the global supply chain*. Mit einem Vorw. von BRANSON, Sir R. Abingdon, UK und New York: Routledge (Ein Imprint von Taylor & Francis Group), 2013.
- [Sau02] SAUER, J.: *Multi-Site Scheduling. Hierarchisch koordinierte Ablaufplanung auf mehreren Ebenen*. Habilitationsschrift. Oldenburg: Fachbereich Informatik, Universität Oldenburg, 2002.

- [SB00] SABUNCUOGLU, I. ; BAYIZ, M.: Analysis of reactive scheduling problems in a job shop environment. In: *European Journal of Operational Research* 126.3 (2000), S. 567–586.
- [Sch01] SCHOLL, A.: *Robuste Planung und Optimierung. Grundlagen – Konzepte und Methoden – Experimentelle Untersuchungen*. Heidelberg: Physica-Verlag, 2001.
- [Sch96] SCHLÜCHTERMANN, J.: *Planung in zeitlich offenen Entscheidungsfeldern*. Neue betriebswirtschaftliche Forschung 161. Wiesbaden: Gabler Verlag, 1996.
- [SCXL10] SHI, N. ; CHEUNG, R. K. ; XU, H. ; LAI, K. K.: An adaptive routing strategy for freight transportation networks. In: *Journal of the Operational Research Society* 62.4 (2010), S. 799–805.
- [SD06] SNYDER, L. V. ; DASKIN, M. S.: Stochastic p -robust location problems. In: *IIE Transactions* 38.11 (2006), S. 971–985.
- [SDR09] SHAPIRO, A. ; DENTCHEVA, D. ; RUSZCZYŃSKI, A.: *Lectures on stochastic programming. Modeling and theory*. MOS-SIAM Series on Optimization. Philadelphia: Society for Industrial und Applied Mathematics, 2009.
- [SF12] SCHAPIRE, R. E. ; FREUND, Y.: *Boosting. Foundations and algorithms*. Adaptive computation and machine learning. Cambridge und London: MIT Press, 2012.
- [SG09] SABUNCUOGLU, I. ; GOREN, S.: Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. In: *International Journal of Computer Integrated Manufacturing* 22.2 (2009), S. 138–157.
- [Sha13] SHARAFI, A.: *Knowledge Discovery in Databases. Eine Analyse des Änderungsmanagements in der Produktentwicklung*. Informationsmanagement und Computer Aided Team. Wiesbaden: Springer Gabler, 2013.
- [Sib73] SIBSON, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. In: *The Computer Journal* 16.1 (1973), S. 30–34.
- [Sil04] SILVER, E. A.: An overview of heuristic solution methods. In: *The Journal of the Operational Research Society* 55.9 (2004), S. 936–956.
- [Soy73] SOYSTER, A. L.: Convex programming with set-inclusive constraints and applications to inexact linear programming. In: *Operations Research* 21.5 (1973), S. 1154–1157.
- [Ste12] STEIN, A.: Umschlagsprozesse in der Logistik. In: KLAUS, P. ; KRIEGER, W. ; KRUPP, M. (Hrsgg.): *Gabler Lexikon Logistik. Management logistischer Netzwerke und Flüsse*. 5. Aufl. Wiesbaden: Springer Gabler, 2012, S. 600–606.
- [Ste56] STEINHAUS, H.: Sur la division des corps matériels en parties. In: *Bulletin de l'Académie Polonaise des Sciences* 4.12 (1956), S. 801–804.

- [Tal09] TALBI, E.-G.: *Metaheuristics. From design to implementation*. Hoboken, NJ: John Wiley & Sons, 2009.
- [TBNA12] TALBI, E.-G. ; BASSEUR, M. ; NEBRO, A. J. ; ALBA, E.: Multi-objective optimization using metaheuristics: non-standard algorithms. In: *International Transactions in Operational Research* 19.1–2 (2012), S. 283–305.
- [TG08] TAN, K. C. ; GOH, C. K.: Handling uncertainties in evolutionary multi-objective optimization. In: ZURADA, J. M. ; YEN, G. G. ; WANG, J. (Hrsgg.): *Computational intelligence: research frontiers*. Lecture Notes in Computer Science 5050. Berlin und Heidelberg: Springer, 2008, S. 262–292.
- [Thi08] THIELE, A.: Robust stochastic programming with uncertain probabilities. In: *IMA Journal of Management Mathematics* 19.3 (2008), S. 289–321.
- [TK09] THEODORIDIS, S. ; KOUTROUMBAS, K.: *Pattern recognition*. 4. Aufl. Amsterdam: Elsevier, 2009.
- [TNL10] TANG, L. C. ; NG, T. S. ; LAM, S. W.: Improving air cargo service through efficient order release. In: CHEN, J. (Hrsg.): *The 7th international conference on service systems and service management*. 2010, S. 136–141.
- [TR14] TULABANDHULA, T. ; RUDIN, C.: *Robust optimization using machine learning for uncertainty sets*. 2014. arXiv: 1407.1097 [math.OA].
- [TSK06] TAN, P.-N. ; STEINBACH, M. ; KUMAR, V.: *Introduction to data mining*. Nachdruck 2009. Boston et al.: Addison Wesley (Ein Imprint von Pearson), 2006.
- [UM10] UNLU, Y. ; MASON, S. J.: Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. In: *Computers & Industrial Engineering* 58.4 (2010), S. 785–800.
- [VHL00] VIEIRA, G. E. ; HERRMANN, J. W. ; LIN, E.: Predicting the performance of rescheduling strategies for parallel machine systems. In: *Journal of Manufacturing Systems* 19.4 (2000), S. 256–266.
- [VHL03] VIEIRA, G. E. ; HERRMANN, J. W. ; LIN, E.: Rescheduling manufacturing systems: a framework of strategies, policies, and methods. In: *Journal of Scheduling* 6.1 (2003), S. 39–62.
- [VMOR99] VOSS, S. ; MARTELLO, S. ; OSMAN, I. H. ; ROUCAIROL, C. (Hrsgg.): *Metaheuristics. Advances and trends in local search paradigms for optimization*. Boston, MA: Springer US, 1999.
- [WFH11] WITTEN, I. H. ; FRANK, E. ; HALL, M. A.: *Data mining. Practical machine learning tools and techniques*. 3. Aufl. The Morgan Kaufmann Series in Data Management Systems. Amsterdam et al.: Morgan Kaufmann (Ein Imprint von Elsevier), 2011.

- [WM97] WILSON, D. R. ; MARTINEZ, T. R.: Improved heterogeneous distance functions. In: *Journal of Artificial Intelligence Research* 6 (1997), S. 1–34.
- [XCLQ13] XU, X. ; CUI, W. ; LIN, J. ; QIAN, Y.: Robust makespan minimisation in identical parallel machine scheduling problem with interval data. In: *International Journal of Production Research* 51.12 (2013), S. 1–17.
- [XLC14] XU, X. ; LIN, J. ; CUI, W.: Hedge against total flow time uncertainty of the uniform parallel machine scheduling problem with interval data. In: *International Journal of Production Research* 52.19 (2014), S. 1–15.
- [XT08] XUE, J.-H. ; TITTERINGTON, D. M.: Comment on “On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes”. In: *Neural Processing Letters* 28.3 (2008), S. 169–187.
- [XXX11] XU, J. ; XU, X. ; XIE, S. Q.: Recent developments in dual resource constrained (DRC) system research. In: *European Journal of Operational Research* 215.2 (2011), S. 309–318.
- [XZMC14] XU, D. ; ZHANG, C. W. ; MIAO, Z. ; CHEUNG, R. K.: A flow allocation strategy for routing over multiple flow classes with an application to air cargo terminals. In: *Computers & Operations Research* 51 (2014), S. 1–10.
- [YCC06] YAN, S. ; CHEN, C.-K. ; CHEN, C.-H.: Cargo terminal shift setting and manpower supplying in short-term operations. In: *Journal of Marine Science and Technology* 14.2 (2006), S. 109–118.
- [YCC08] YAN, S. ; CHEN, C.-H. ; CHEN, C.-K.: Short-term shift setting and manpower supplying under stochastic demands for air cargo terminals. In: *Transportation* 35.3 (2008), S. 425–444.
- [Ye03] YE, N.: *The handbook of data mining*. Mahwah, NJ und London: Lawrence Erlbaum Associates, 2003.
- [ZB05] ZÄPFEL, G. ; BRAUNE, R.: *Moderne Heuristiken der Produktionsplanung. Am Beispiel der Maschinenbelegung*. WiSo-Kurzlehrbücher : Reihe Betriebswirtschaft. München: Vahlen, 2005.
- [ZB12] ZIO, E. ; BAZZO, R.: A comparison of methods for selecting preferred solutions in multiobjective decision making. In: KAHRAMAN, C. (Hrsg.): *Computational intelligence systems in industrial engineering. With recent theory and applications*. Atlantis Computational Intelligence Systems 6. Paris: Atlantis Press, 2012, S. 23–43.
- [ZE02] ZADROZNY, B. ; ELKAN, C.: Transforming classifier scores into accurate multiclass probability estimates. In: HAND, D. ; KEIM, D. ; NG, R. (Hrsgg.): *KDD - 2002. Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining*. 2002, S. 694–699.

-
- [Zha04] ZHANG, H.: The optimality of naive Bayes. In: BARR, V. ; MARKOV, Z. (Hrsgg.): *Proceedings of the seventeenth international Florida artificial intelligence research society conference*. Menlo Park, CA: AAAI Press, 2004, S. 562–567.
- [Zha05] ZHANG, H.: Exploring conditions for the optimality of naïve Bayes. In: *International Journal of Pattern Recognition and Artificial Intelligence* 19.02 (2005), S. 183–198.
- [ZM14] ZAKI, M. J. ; MEIRA JR., W.: *Data mining and analysis. Fundamental concepts and algorithms*. New York: Cambridge University Press, 2014.
- [ZWB12] ZÄPFEL, G. ; WASNER, M. ; BRAUNE, R.: Integration between flow scheduling and personnel allocation in a parallel line system in automotive industry. In: *Zeitschrift für Betriebswirtschaft* 82.4 (2012), S. 417–432.