

DISSERTATION

Entwicklung eines hybriden Optimierungsverfahrens für dynamische multikriterielle Produktionsglättung am Beispiel der Lebensmittelindustrie

durchgeführt an der

TECHNISCHEN UNIVERSITÄT WIEN



an der

Fakultät für Maschinenwesen und Betriebswissenschaften

Institut für Managementwissenschaften

Arbeitsbereich Betriebstechnik und Systemplanung

Betreuer: Univ.-Prof. Dr.-Ing. Dipl. Wirtsch.-Ing. Prof. eh. Dr. h.c. Wilfried Sihm

1. Gutachter: Ao. Univ.-Prof. Dipl.-Ing. Dr. techn. Norbert Seifert (Montanuniversität Leoben)
2. Gutachter: Ao. Univ.-Prof. Dipl.-Ing. Dr. techn. Burkhard Kittl (Technische Universität Wien)

Vorgelegt von: Dipl.-Ing. Florian Felix Kamhuber, BSc
Matrikelnummer: 0435138

zum

Erlangen des akademischen Grades

Doktor der technischen Wissenschaften

(Dr. techn.)

„Just as a child creates magnificent fortresses through the arrangement of simple blocks of wood, so does a genetic algorithm seek near optimal performance through the juxtaposition of short, low-order, high-performance schemata, or building blocks.”

(David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning)

„According to Darwin’s Origin of Species, it is not the most intellectual of the species that survives; it is not the strongest that survives; but the species that survives is the one that is able best to adapt and adjust to the changing environment in which it finds itself.”

(Leon C. Megginson, Lessons from Europe for American Business)

„Species do not evolve to perfection, but quite the contrary. The weak, in fact, always prevail over the strong, not only because they are in the majority, but also because they are the more crafty.”

(Friedrich Nietzsche, The Twilight of the Idols)

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass die vorliegende Arbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen von mir selbstständig erstellt wurde. Alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, sind in dieser Arbeit genannt und aufgelistet. Die aus den Quellen wörtlich entnommenen Stellen, sind als solche kenntlich gemacht.

Das Thema dieser Arbeit wurde von mir bisher weder im In- noch Ausland einer Beurteilerin/einem Beurteiler zur Begutachtung in irgendeiner Form als Prüfungsarbeit vorgelegt. Diese Arbeit stimmt mit der von den Begutachterinnen/Begutachtern beurteilten Arbeit überein.

Ich nehme zur Kenntnis, dass die vorgelegte Arbeit mit geeigneten und dem derzeitigen Stand der Technik entsprechenden Mitteln (Plagiat-Erkennungssoftware) elektronisch-technisch überprüft wird. Dies stellt einerseits sicher, dass bei der Erstellung der vorgelegten Arbeit die hohen Qualitätsvorgaben im Rahmen der geltenden Regeln zur Sicherung guter wissenschaftlicher Praxis „Code of Conduct“ an der TU Wien eingehalten wurden. Zum anderen werden durch einen Abgleich mit anderen studentischen Abschlussarbeiten Verletzungen meines persönlichen Urheberrechts vermieden.

Wien, am 01.08.2022

Unterschrift:

Felix Kamhuber

Florian Felix Kamhuber

Danksagung

Die vorliegende Dissertation wurde in der Zeit von November 2017 bis August 2022 am Institut für Managementwissenschaften an der Technischen Universität Wien verfasst.

Besonderen Dank seitens der Technischen Universität Wien gilt daher meinem Betreuer Herrn Prof. Dr. Sihm für seine ausgezeichnete Betreuung während des kompletten Zeitraums.

Darüber hinaus möchte ich mich bei Herrn Dipl.-Ing. Peter Schieder (Geschäftsbereichsleiter, Fraunhofer Austria Research GmbH) sowie Herrn Dipl.-Wirtsch.-Ing. Dipl.-Ök. Jan Henjes (Gruppenleiter, Fraunhofer Austria Research GmbH) v.a. dafür bedanken, dass sie mich auf das Thema Produktionsglättung in Bezug auf ein umfangreiches Industrieprojekt aufmerksam gemacht haben. Außerdem wurde auf deren Empfehlung ein thematisch entsprechender Forschungsantrag (ASPeCT) bei der FFG eingereicht und genehmigt, welcher es mir ermöglicht hat, mich im Zuge der Projektbearbeitung noch intensiver mit diesem Thema in Bezug auf verschiedene Branchen und seinen Ausprägungen und Aspekten in Form der Anwendung von (Meta-) Heuristiken, Simulation und Data Science zu beschäftigen.

Außerdem möchte ich mich ganz besonders bei Dr. Thomas Sobottka sowie Dipl.-Ing. Markus Spielbichler für die ausgezeichnete Zusammenarbeit in den gemeinsamen Forschungsprojekten BaMa und ASPeCT bedanken. In diesem Rahmen sind viele gute Anregungen sowie Ideen entstanden, welche in diversen wissenschaftlichen Beiträgen bzw. in diese Arbeit eingeflossen sind. Herzlichen Dank an dieser Stelle für diverse Anregungen, Anmerkungen und Ergänzungen sowie Korrekturen. Ein weiterer Dank geht an Dr. Bernhard Heinzl, der in den genannten Forschungsprojekten und Publikationen tatkräftig unterstützt hat.

Die Begeisterung für das Thema Operations Research, insbesondere den Fachbereich (meta-) heuristische Optimierung, wurde im Laufe der vorangegangenen Bachelor- und Masterstudien Industrielogistik an der Montanuniversität Leoben in mir geweckt und entstand speziell im Rahmen der Vorlesungen „Mathematische Grundlagen des Operations Research“ sowie „Optimierung für Industrielogiker“, welche von Herrn Prof. Dr. Seiffter abgehalten worden sind. Prof. Seiffter fungiert als einer der Gutachter für die vorliegende Dissertation. Prof. Dr. Kittl von der TU Wien übernimmt die Rolle des zweiten Gutachters.

Zuletzt möchte ich mich beim Geschäftsführer bzw. dem IT-Leiter von Seiten des industriellen Entwicklungspartners aus der Lebensmittelindustrie bedanken. Diese haben es mir durch die Zurverfügungstellung des entsprechenden industriellen Anwendungsfalles und der dazugehörigen Daten (auf einer Datenbank) ermöglicht, mich entsprechend intensiv und grundlegend mit den OR-Themen der heuristischen und Kombinatorischen Optimierung in Bezug auf eine industrielle Anwendung mit dem Fokus auf das Thema Produktionsglättung beschäftigen zu können.

Diese Arbeit, die ich in LyX verfasst und in LaTeX formatiert habe, widme ich meinen Eltern Karl und Anna Kamhuber, meinen Brüdern Christoph und Alex Kamhuber sowie meiner Freundin und Verlobten Alice Steinbauer, die mich während meines Doktoratsstudiums fortwährend unterstützt haben.

Kurzfassung

Absatzfluktuation resultiert in der Unternehmenspraxis aus Nachfrageschwankungen. Außerdem erfordern nicht stufenlos zu- und abschaltbare Produktionsressourcen zur Deckung des Kapazitätsbedarfs in der Regel Mehrkosten in der Produktion. Eine gute Möglichkeit, diesen sich zusätzlich ergebenden Flexibilitätskosten effizient zu begegnen, besteht in der Minimierung dieser Kosten durch eine gezielte Glättung der Produktionsauslastung. Die Optimierung selbst kann durch (meta-) heuristische Optimierungsverfahren bzw. hybride Algorithmen effizient bewerkstelligt werden. Dabei wird die Reduktion der Absatzschwankungen in der Produktion durch eine entsprechende Zielfunktion berücksichtigt. Die Produktionsglättung verfolgt das Ziel, ein Produktionssystem über einen längeren Zeitraum möglichst gleichmäßig auszulasten und diese Schwankungen erheblich und kostenwirksam bzw. wirtschaftlich zu reduzieren. In der Praxis vorherrschende einfache bzw. regelbasierte Methoden für die Produktionsglättung weisen diverse Mängel auf, z.B. die mangelnde Aufteilung von Produktionslosen, eine fehlende separate Betrachtung von Aktions- und Kurantmengen und eine rein statische Problembetrachtung, wobei dynamische Nebenbedingungen und eine rollierende Planung vernachlässigt werden. Darüber hinaus verfügen diese Methoden nicht über einen holistischen Ansatz, welcher das dynamische Produktionssystem aus mehreren Planungsdimensionen integriert und simulationsgestützt bewertet. Aus methodischer Sicht fehlt den zur Verfügung stehenden Ansätzen eine Variabilisierung der Prozessdurchlaufzeit sowie die laufende Überwachung und Anpassung von Stammdaten. Dadurch kann sich das Verfahren nicht selbstständig an veränderte Rahmenbedingungen anpassen.

Diese Dissertation beschäftigt sich mit der Entwicklung einer Methodik für eine optimierte Mittelfristplanung, welche es ermöglicht, mehrere Planungsdimensionen gleichzeitig, in Bezug auf eine gleichmäßige Auslastung und Versorgung der Produktion (Mengen, Kapazitäten, Ressourcen) am Beispiel der fleischverarbeitenden Industrie dynamisch zu betrachten und zu optimieren. Die Methodik umfasst einen für die multikriterielle Glättung von Produktionsmengen speziell angepassten hybriden Algorithmus, der eine problemspezifisch zugeschnittene Metaheuristik konsekutiv auf eine wissensbasierte Produktionsglättungsheuristik aufsetzt. Darüber hinaus sammelt das Optimierungsverfahren rollierend Daten, auf deren Basis die Informationen und Aktionen der Vergangenheit bewertet bzw. schlussfolgernd in Form eines Verbesserungs-Regelkreises die Parameter und spezifischen Strategien für die Bestandsoptimierung angepasst werden. Dadurch unterscheidet sich der methodische Ansatz dieser Dissertation signifikant von den Limitationen bestehender Methoden für die Glättung von Produktionsmengen. Im Ergebnis soll dadurch eine deutlich leistungsfähigere Glättungsmethodik für die mittel- bis langfristige Produktionsplanung entstehen. Gemäß den Anforderungen, auf Basis der vorliegenden Problemstellung eines Produktionssystems mit schwankendem Produktionsausstoß, evaluiert diese Arbeit die Leistungsfähigkeit und Laufzeiteffizienz der entwickelten Referenz-Verfahren und schlussfolgert daraus eine Empfehlung für eine multikriterielle Produktionsglättungsmethodik.

Schlagwörter zur Arbeit: Heuristiken, Metaheuristiken, Multikriterielle Optimierung, Produktionsglättung, Rollierende Produktionsplanung, Losgrößenplanung.

Abstract

In practice, fluctuations in sales result from fluctuations in demand. In addition, production resources that cannot be switched on and off continuously to meet capacity requirements generally require additional costs in production. A good way of efficiently counteracting these additional flexibility costs is to minimize these costs by selectively smoothing production capacity utilization. The optimization itself can be efficiently accomplished by (meta-) heuristic optimization methods or hybrid algorithms. Here, the reduction of sales fluctuations in production is taken into account by a corresponding objective function. Production smoothing pursues the goal of utilizing a production system as evenly as possible over a longer period of time and reducing these fluctuations significantly and cost-effectively or economically. In general, prevailing simple or rule-based methods for production smoothing have various deficiencies, e.g. the lack of division of production lots, a lack of separate consideration of promotion and curant quantities and a purely static problem consideration, neglecting dynamic constraints and rolling planning. Furthermore, these methods do not have a holistic approach, which integrates the dynamic production system from several planning dimensions and evaluates it based on simulation. From a methodological point of view, the available approaches lack a variabilization of the process lead time as well as the continuous monitoring and adjustment of master data. As a result, the method is not able to adapt independently to changing conditions.

This dissertation deals with the development of a methodology for an optimized medium-term planning, which allows to dynamically consider and optimize several planning dimensions at the same time, with respect to a consistent utilization and supply of production (quantities, capacities, resources) using the example of the meat processing industry. The methodology includes a hybrid algorithm specially adapted for multi-criteria smoothing of production quantities, which consecutively superimposes a problem-specific tailored metaheuristic on a knowledge-based production smoothing heuristic. In addition, the optimization procedure collects data on a rolling basis, on the basis of which past information and actions are evaluated or, conclusively, the parameters and specific strategies for inventory optimization are adjusted in the form of an improvement control loop. Thus, the methodological approach of this dissertation differs significantly from the limitations of existing methods for smoothing production quantities. As a result, this should lead to a significantly more powerful smoothing methodology for medium to long term production planning. According to the requirements, based on the present problem of a production system with fluctuating production output, this thesis evaluates the performance and runtime efficiency of the developed reference methods and concludes a recommendation for a multi-criteria production smoothing methodology.

Keywords: Heuristics, Metaheuristics, Multi-criteria optimization, Production smoothing, Rolling production planning, Lot size planning.

Inhaltsverzeichnis

1	Einleitung	8
1.1	Motivation	9
1.2	Allgemeine Problemstellung	12
1.3	Ausgangspunkt der Lösungssuche für die vorliegende Problemstellung	13
1.4	Im Rahmen dieser Arbeit behandelte Forschungsfragen	14
1.5	Forschungsmethode und inhaltlicher Aufbau	14
1.5.1	Forschungsmethodik	14
1.5.2	Inhaltlicher Aufbau	17
2	Grundlagen von Produktion und Optimierung	20
2.1	Begriffsdefinitionen	20
2.2	Grundlagen zur Planung von Produktion und Logistik	21
2.2.1	Produktionsplanung und -steuerung	21
2.2.2	Ebenen der Kapazitätsplanung	22
2.2.3	Kapazitätsanpassungsstrategien	22
2.2.4	Programmorientierte Bedarfsermittlung	23
2.2.5	Losgrößenplanung und Produktionssystemoptimierung	24
2.2.6	Produktionsglättung	24
2.3	Mathematische Grundlagen der Nichtlinearen Multikriteriellen Optimierung	26
2.3.1	Methodische Grundlagen	27
2.3.1.1	Pareto-Optimalität und Dominanz	28
2.3.1.2	Konvexität	30
2.3.1.3	Komplexitätstheorie	31
2.3.1.4	Landau-Notation	32
2.3.2	Herausforderungen bei der Modellierung und Lösungssuche	33
2.3.3	Fazit	37
2.4	Thematische Eingliederung innerhalb des Operations Research	37
2.5	Systematische Eingrenzung des Betrachtungsbereichs	38
2.5.1	Eingrenzung auf Grund der Charakteristika des Produzenten (Branche) und Produktionsprozesses	39
2.5.2	Eingrenzung auf Grund der Anforderungen an den Planungshorizont und die Planungsaufgabe	40
2.6	Methodische Problemabgrenzung	41
2.6.1	Abgrenzung gegenüber Absatzprognoseglättung	41
2.6.2	Abgrenzung gegenüber Kapazitätsflexibilitätsstrategien	42
2.6.3	Abgrenzung gegenüber kommerziellen Advanced Planning and Scheduling Systemen (APS) und Solvern	43
2.7	Fazit	44
3	Nichtlineare Optimierungsverfahren: Stand der Technik	46
3.1	Überblick	46
3.2	Deterministische Verfahren	48
3.2.1	Numerische Verfahren	48
3.2.2	Heuristiken	51
3.2.2.1	Vorteile von Heuristiken	51
3.2.2.2	Klassifikation von Heuristiken	52

3.2.3	Präferenz ausdrückende Ersatzoptimierungsverfahren	55
3.2.3.1	Die Gewichtete-Summen-Methode (GSM)	57
3.2.3.2	Die Epsilon-Constraint Methode (ECM)	59
3.2.3.3	Die Methode der gewichteten Metriken (GM)	60
3.2.3.4	Lexikographische Optimierung (LO)	61
3.2.3.5	Goal Programming (GP)	62
3.2.3.6	Referenzpunkt-Methode (RM)	62
3.2.3.7	Tchebycheff Methode (TM)	63
3.2.3.8	Bewertung und Diskussion	64
3.2.4	Fazit	65
3.3	Stochastische Verfahren: Populationsbasierte Metaheuristiken und trajektorienbasierte Algorithmen	66
3.3.1	Grundlegende Definitionen und Klassifikation von Metaheuristiken	66
3.3.2	Genetische und Evolutionäre Algorithmen (GA, EA)	68
3.3.3	Simulated Annealing (SA)	74
3.3.4	Fazit	77
3.4	Hybride Optimierungsverfahren	78
3.4.1	Varianten der Hybridisierung	79
3.4.2	Kombination mit Heuristiken bzw. Metaheuristiken	79
3.4.3	Simheuristics: Kombination mit Simulation	82
3.4.4	Kombinationen mit Machine Learning (ML) und Data-Mining (DM) Methoden	85
3.4.5	Spezifische Optimierungsstrategien	85
3.4.5.1	Gradual-Priority Weighting-Ansätze	86
3.4.5.2	Multi-objective Genetic Algorithm (MOGA)	87
3.4.5.3	Random Search Production Smoothing Heuristic	88
3.4.5.4	Advance Order Production Smoothing Heuristic	88
3.4.5.5	Line Balancing-Ansätze	89
3.4.5.6	Planung flexibler Produktionskapazitäten	90
3.4.5.7	Datengestützte evolutionäre Optimierung	90
3.4.6	Fazit	90
3.5	Rollierende Planungsansätze	91
3.6	Klassische und moderne kapazitive Losgrößenprobleme	91
3.7	Umgang mit Nebenbedingungen	93
3.8	Bewertung und Vergleich von Optimierungsverfahren	94
3.9	Diskussion	95
3.10	Entwicklungsbedarf	98
3.11	Fazit	101
4	Produktionsproblembeschreibung und Anforderungsübersicht	102
4.1	Produktionsablauf des vorliegenden Produktionssystems	102
4.2	Spezifika des vorliegenden Produktionssystems	103
4.3	Spezifische Anforderungsübersicht an das zu entwickelnde multikriterielle Optimierungsverfahren	105
4.4	Fazit	109
5	Entwicklung der Produktionsglättungsmethodik	110
5.1	Zielsetzung und erwartetes Ergebnis der in dieser Arbeit zu entwickelnden Methodik	110
5.2	Ableitung des konkreten Zielfunktionssystems	112
5.3	Notation des Zielfunktionssystems	113
5.3.1	Zielfunktionssystem: Teilzielfunktionen und Nebenbedingungen	114
5.3.2	Berücksichtigung von Nebenbedingungen im Zuge der Optimierung	116
5.4	Modellierung der Kapazitätsauslastung: Integrierte dynamische Simulation	118
5.5	Systemintegration und technische Anbindung	119

5.6	Entwicklung des hybriden Optimierungsverfahrens	120
5.6.1	Phase 1: Heuristische Optimierung	120
5.6.2	Phase 2: Metaheuristische Optimierung	124
5.6.2.1	Vorbereitungen	125
5.6.2.2	Gemeinsamer algorithmischer Ablauf (SA, GA)	126
5.6.2.3	Metaheuristische Optimierung: Widersprüchliche Teilzielsetzungen	128
5.6.2.4	Unterschiede Implementierung SA - GA	129
5.6.3	Objektorientierte Modellierung	130
5.7	Statistische Datenanalyse	131
5.8	Fazit	132
6	Validierung des entwickelten Optimierungsverfahrens	134
6.1	Voraussetzungen und Rahmenbedingungen für die Validierung	134
6.2	Auswertung und Interpretation der Ergebnisse	135
6.2.1	Ergebnisse der heuristischen Optimierung	135
6.2.2	Ergebnisse der vergleichenden metaheuristischen Optimierung	138
6.2.2.1	Parametrisierung des Simulated Annealing-Verfahrens	138
6.2.2.2	Optimierungsergebnisse des Simulated Annealing-Verfahrens	140
6.2.2.3	Implementierung und Optimierungsergebnisse des Genetischen Algorithmus	141
6.2.2.4	Vergleich der beiden Algorithmen: SA versus GA	142
6.2.2.5	Aktivierung der Rekombination (GA)	143
6.2.3	Ergebnisse der hybriden Optimierung	147
6.3	Theoretische und praktische Bewertung der Laufzeitkomplexität	149
6.4	Ergebnisse der statistischen Datenanalyse	151
6.5	Nutzen der Optimierungsmethode	152
6.5.1	Bewertung der Kostenrelevanz der Optimierungsergebnisse:	152
6.5.2	Simulation von Szenarien	154
6.5.3	Anpassungsfähigkeit	154
6.6	Abgleich mit den Anforderungen an das Optimierungsverfahren	154
6.7	Fazit	156
7	Conclusio und Ausblick	157
7.1	Diskussion	158
7.1.1	Kritische Betrachtung der Ergebnisse und Vergleich mit dem Stand der Technik	158
7.1.2	Beantwortung der Forschungsfragen	160
7.2	Ausblick	162
7.2.1	Erweiterungen der Methodik	162
7.2.2	Weiterer Forschungsbedarf	164
A	Anhang	165
A.1	Grundlegende Begriffe aus der Vektoranalysis	165
A.2	Numerische gradientenbasierte Verfahren	165
A.2.1	Das Gradientenverfahren	165
A.2.2	Das Newton-Verfahren	167
A.3	Das Nelder-Mead-Simplex Verfahren	168
A.4	Lokale Konstruktions- und Verbesserungsheuristiken	169
A.4.1	K-Opt Algorithmus	169
A.4.2	Insertion Search	170
A.4.3	Lin-Kernighan Heuristik	170
A.5	Goal-Chasing Heuristik	171
A.6	Weitere Ersatzoptimierungsverfahren	172
A.6.1	Methoden ohne Präferenz	172
A.6.1.1	Global Criterion Methode	173

A.6.1.2	Neutrale Kompromisslösung	173
A.6.2	Weitere interaktive Präferenz ausdrückende Methoden	173
A.6.2.1	Interaktive Surrogate Worth Trade-Off Methode	173
A.6.2.2	Geoffrion-Dyer-Feinberg Methode	174
A.6.2.3	Das STEM-Verfahren	174
A.7	Monte-Carlo Verfahren	175
A.8	Weitere metaheuristische Optimierungsverfahren	176
A.8.1	Ant Colony Optimization (ACO)	176
A.8.2	Partikelschwarmoptimierung (PSO)	180
A.8.3	Tabu Search (TS)	181
A.8.4	Variable Neighborhood Search (VNS)	181
A.8.5	Iterierte Lokale Suche (ILS)	182
A.9	Konvexe Optimierung	182
A.10	Entwicklung von ERP- und APS-Systemen	184
A.11	Zufallszahlen	186
A.12	Verteilungsplot	187
A.13	Gewichtete-Summen-Methode: Praktisches Anwendungsbeispiel	188
A.14	Schema-Theorem: Beispiel	189
A.15	Kritik am Schema-Theorem und der Building-Block-Hypothese	189
A.16	Ausschnitt aus dem Quellcode (Zielfunktionsberechnung in Java)	190
A.17	Wissenschaftliche Publikationen mit konkretem Bezug zum Thema dieser Dissertation	192
A.18	Lebenslauf des Autors	194

Die approbierte gedruckte Originalversion dieser Dissertation ist an der TU Wien Bibliothek verfügbar.
 The approved original version of this doctoral thesis is available in print at TU Wien Bibliothek.

Abbildungsverzeichnis

1.1	Der Bullwhip-Effekt entlang einer Supply-Chain (Quelle: eigene Darstellung, in Anlehnung an Dickmann 2015, S.186 und Beer 2014, S. 25)	10
1.2	Mengenschwankungen in Abhängigkeit von Unternehmensgröße bzw. Vorhersagezeitraum (Grundmann und Reinsch, 2006, S. 43)	10
1.3	Erweiterter Design Science Ansatz (Quelle: eigene Darstellung in Anlehnung an Hevner Alan R. 2007, S. 1-5)	15
1.4	Aufbau dieser Arbeit (Quelle: eigene Darstellung)	19
2.1	Gegenüberstellung zweier Kapazitätsanpassungsstrategien : (links) Emanzipation bzw. (rechts) partielle Emanzipation (Abele et al. 2008, S. 585-586)	23
2.2	Design- und Bildraum (Quelle: eigene Darstellung, in Anlehnung an Vaillant 2015, S. 67)	29
2.3	Konvexität von Mengen im \mathbb{R}^2 (Quelle: eigene Darstellung, angelehnt an Stein 2018, S. 39)	30
2.4	Konvexität von Funktionen (Quelle: eigene Darstellung, in Anlehnung an Stein 2018, S. 39)	30
2.5	Gegenläufigkeit von Liefertermintreue und Lagerhaltungskosten (März et al. 2011, S. 71)	35
2.6	Lokale und globale Optima in einer multimodalen Fitnesslandschaft (Schüller 2009, S. 34)	36
2.7	Systematische Eingrenzung des Betrachtungsbereichs (Quelle: eigene Darstellung) . . .	38
2.8	Anforderungen an den Prozess und Planungshorizont (Quelle: eigene Darstellung) . . .	45
3.1	Übersicht: Taxonomie von Optimierungsverfahren (Quelle: eigene Darstellung, in Anlehnung an Kux 2011, S. 15f. und Stöcker 2007, S.14)	47
3.2	Prinzip des (MIP) Branch-and-Bound-Verfahrens (Quelle: Meywerk 2007, S. 289) . . .	49
3.3	Erreichbare Punkte (A, B) und unerreichbarer Punkt (C) der Pareto-Front (Quelle: eigene Darstellung in Anlehnung an Branke et al. 2008, S. 5)	58
3.4	Optimale Lösungen der <i>Epsilon – Constraint</i> Methode für ein bikriterielles Problem ($j = 2, \varepsilon_1$) (Quelle: eigene Darstellung, in Anlehnung an Chiandussi et al. 2012, S. 920)	59
3.5	Gegenüberstellung eines (a) gut durchsuchten Lösungsraums mit großer Diversität und (b) weniger gut durchsuchten Lösungsraums mit mangelnder Diversität (vgl. Austin 2020)	70
3.6	Fitnessproportionale Selektion bzw. Roulette-Rad-Auswahl (Feltl 2003, S. 32)	71
3.7	<i>Flipmutation</i> (links) und <i>Swapmutation</i> (rechts) (Feltl 2003, S. 35)	72
3.8	Globale Optimierung mittels Simulated Annealing (Ghasemalizadeh et al. 2016, S. 1671)	76
3.9	High-Level Relay Hybridisierung (eigene Darstellung, angelehnt an Talbi 2016, S. 176)	81
3.10	Konzept der Simulationsgestützten Optimierung mittels SiRO (März et al. 2011, S. 68)	84
3.11	Verlauf der adaptiven Gewichtung (Zitzler et al. 2001, S. 100)	86
4.1	Betrachteter Ausschnitt des Produktionssystems: Vom Einsalzen bis zum Abschluss der Grundproduktion (vgl. Kamhuber et al. 2019, S. 2110)	103
5.1	Beispielhafter Ablauf der Produktionsglättungsmethodik (Quelle: eigene Darstellung) .	111
5.2	Dynamische Simulation zur Erstellung einer Kapazitätsauslastungsvorschau (vgl. Kamhuber et al. 2019, S. 2113)	118
5.3	Systemintegration und technische Anbindung (Quelle: eigene Darstellung)	119
5.4	Hybrides Optimierungskonzept (Quelle: eigene Darstellung)	120
5.5	Regelbasierte Produktionsglättung mittels Produktionslückenfüllung durch Versatz von Produktionsbelastungsspitzen (vgl. Kamhuber et al. 2019, S. 2112)	122
5.6	Lagerbestandsentwicklung: Schematische Aufteilung des Planungshorizonts (vgl. Kamhuber et al. 2020, S. 7; Absi und Kedad-Sidhoum 2007, S. 177)	124

5.7	Beispiel für einen schlechten Kompromiss (<i>trade-off</i>) (vgl. Kamhuber et al. 2020, S. 8)	128
5.8	Gegenüberstellung von Mutations- und Rekombinationsoperator (Quelle: eigene Darstellung)	130
5.9	Boxplot zur Bewertung der Bestandsvolatilität in Bezug auf die Produktionsartikel (vgl. Kamhuber et al. 2019, S. 2115)	131
5.10	Durchschnittliche statische und dynamische Produktionsabweichungen zwischen Produktionsvorschlag und Produktionsrückmeldung (vgl. Kamhuber et al. 2019, S. 2115)	132
6.1	Optimierungsergebnisse der heuristischen Optimierung, Datensatz 1 (vgl. Kamhuber et al. 2020, S. 10)	136
6.2	Optimierungsergebnisse der heuristischen Optimierung, Datensatz 2 (vgl. Kamhuber et al. 2020, S. 10)	137
6.3	SA Optimierungsergebnisse der rechnergestützten Studie (vgl. Kamhuber et al. 2020, S. 12)	139
6.4	Detailansicht: SA-Optimierung (Quelle: eigene Darstellung)	140
6.5	Ergebnisse der kombiniert heuristischen und SA Optimierung, Datensatz 2 (vgl. Kamhuber et al. 2019, S. 2114)	141
6.6	GA Optimierungsergebnisse für verschiedene Populationsgrößen, Datensatz 3 (vgl. Kamhuber et al. 2020, S. 12)	142
6.7	Vergleich zwischen GA und SA, Datensatz 3 (vgl. Kamhuber et al. 2020, S. 13)	143
6.8	Vergleich unterschiedlicher Rekombinationsraten, Datensatz 4 (Quelle: eigene Darstellung)	144
6.9	Vergleich unterschiedlicher Rekombinationsraten, Datensatz 4' (Quelle: eigene Darstellung)	145
6.10	GA mit Rekombination: Vergleich unterschiedlicher Populationsgrößen, Datensatz 4 (Quelle: eigene Darstellung)	145
6.11	Gegenüberstellung von GA-V2 und GA-V3, Datensatz 4' (Quelle: eigene Darstellung)	146
6.12	Hybride Optimierung: PGH und GA, Datensatz 3 (vgl. Kamhuber et al. 2020, S. 13)	147
6.13	Hybride Optimierung: PGH und GA, Datensatz 4' (Quelle: eigene Darstellung)	148
6.14	Validierung der Laufzeitkomplexität (vgl. Kamhuber et al. 2020, S. 14)	149
6.15	Effekt der Produktionsglättung auf Werksebene (f_2) (vgl. Kamhuber et al. 2020, S. 15)	153
6.16	Konzeptionelles Ergebnis: Optimierte Lösung versus Basislösung (vgl. Kamhuber et al. 2020, S. 2)	155
7.1	Integrations Ebenen vom Digital Model bis zum Digital Twin (Quelle: eigene Darstellung, in Anlehnung an: Kritzinger et al. 2018, S. 1017)	159
A.1	Lin-Kernighan Verfahren: Lokale Neuordnung (vgl. Karapetyan und Gutin 2011, S. 222)	171
A.2	Die Fähigkeit der Wegfindung bei der Futtersuche von Ameisenkolonien (Deepa und Senthilkumar 2016, S. 12)	177
A.3	Notwendige Optimalitätsbedingung (Quelle: eigene Darstellung, in Anlehnung an Stein 2018, S. 51)	183
A.4	Entwicklung von Systemen und Begriffen (eigene Darstellung, angelehnt an Hachtel und Holzbaur 2010, S. 111)	184
A.5	Supply Chain Planning Matrix und APS Module (Stadtler und Kilger 2005, S. 243)	185
A.6	Boxplot (Quelle: Keller 2013)	187
A.7	Ecken (Spezifikationen) und Kanten sowie Ebenen (jeweils Schemata) eines dreidimensionalen Würfels (Quelle: eigene Darstellung, in Anlehnung an Whitley 2007, S.8)	189
A.8	Lebenslauf Felix Kamhuber	194

Tabellenverzeichnis

2.1	Gegenüberstellung unterschiedlicher Laufzeitkomplexitäten (Korte und Vygen 2012, S. 7)	32
3.1	Gegenüberstellung wichtiger Eigenschaften der diskutierten Präferenz-Methoden (vgl. Branke et al. 2008, S. 22)	65
3.2	Analogien zwischen der Thermodynamik und einem Optimierungsproblem (Quelle: eigene Darstellung, in Anlehnung an Michalewicz und Fogel 2004, S. 121)	74
3.3	Vergleichende Bewertung methodisch relevanter Modellierungs- und Optimierungsansätze (Quelle: eigene Darstellung)	100
4.1	Übersicht über die min. und max. Teilprozessdurchlaufzeiten (Quelle: eigene Darstellung)	104
4.2	Anforderungsübersicht an das Optimierungsverfahren (Quelle: eigene Darstellung)	106
5.1	Verwendete Notation für das Zielfunktionssystem (vgl. Kamhuber et al. 2020, S. 5)	114
5.2	Verwendete Notation für die Variablen betreffend die Optimierung und Nebenbedingungen (vgl. Kamhuber et al. 2020, S. 5)	114
5.3	Matrizen zur Darstellung der Lösung (vgl. Kamhuber et al. 2020, S. 5)	114
5.4	Mapping (vm) aus Produktionsartikelversatzmengen (p_{ijt}) und dazugehörigen, noch möglichen Produktionsversatzmengen (p_{ijz}) (vgl. Kamhuber et al. 2020, S. 8)	126
6.1	Heuristische Versatzmengen im Verhältnis zur Gesamtproduktionsmenge (vgl. Kamhuber et al. 2020, S. 10)	136
6.2	Teilzielergebnisse der heuristischen Optimierung (vgl. Kamhuber et al. 2020, S. 11)	138
6.3	Simulated Annealing: Parametersätze und Szenarien (vgl. Kamhuber et al. 2020, S. 11)	139
6.4	SA Optimierungsergebnisse pro Szenario (vgl. Kamhuber et al. 2020, S. 11)	139
6.5	Empfehlung für die SA-Optimierungsparameter (vgl. Kamhuber et al. 2020, S. 12)	140
6.6	GA Empfehlungen (GA-V1) für verschiedene Laufzeiten (vgl. Kamhuber et al. 2020, S. 14)	142
6.7	Umsetzungsvarianten des GA (Quelle: eigene Darstellung)	146
6.8	GA Empfehlungen (V2 und V3) für verschiedene Laufzeiten (Quelle: eigene Darstellung)	147
6.9	Veränderung von f sowie der einzelnen Teilzielfunktionen (Quelle: eigene Darstellung)	149
6.10	Empfehlungen auf Basis der statistischen Datenanalyse (vgl. Kamhuber et al. 2019, S. 2116)	152
A.1	Maximierungsproblem: Bestimmung des optimalen Staubsaugerroboters (Quelle: eigene Darstellung, in Anlehnung an Hakanen et al. 2013, F. 21)	188
A.2	Publikationsverzeichnis des Autors (Quelle: eigene Darstellung)	193

Algorithmenverzeichnis

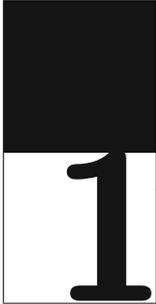
3.1	Der Kruskal Algorithmus (vgl. Korte und Vygen 2012, S. 146)	54
3.2	Lokale Suche (vgl. Glover und Kochenberger 2003, S. 221)	55
3.3	Aufbau und Prinzip eines GA (vgl. Michalewicz 1999, S. 2; Feltl 2003, S. 29)	69
3.4	Simulated Annealing (vgl. Boussaïd et al. 2013, S. 84; Schonert 2008, S. 59; Raidl und Chwatal 2008, F. 142)	75
3.5	Populationsbasierte Iterierte Lokale Suche (vgl. Blum et al. 2011, S. 4136f.)	80
5.1	Multikriterielle Produktionsglättungsheuristik, Phase 1 (vgl. Kamhuber et al. 2018, S. 6)	123
5.2	Phase 2: Simulated Annealing Optimierung (vgl. Kamhuber et al. 2020, S. 9)	127
A.1	Das Gradientenverfahren (vgl. Ulbrich und Ulbrich 2012, S.18):	166
A.2	Lokales Newton-Verfahren für Gleichungssysteme (vgl. Ulbrich und Ulbrich 2012, S. 42)	167
A.3	Das Newton-Verfahren für Optimierungsprobleme (vgl. Ulbrich und Ulbrich 2012, S. 47)	167
A.4	Das Nelder-Mead-Simplex Verfahren (vgl. Kux 2011, S. 20)	169
A.5	Der k-Opt Algorithmus (vgl. Boelter 2013, S.3)	169
A.6	Insertion Search (II, C , α) (vgl. Tseng und Lin 2009, S. 87)	170
A.7	Die Lin-Kernighan Heuristik (vgl. Karapetyan und Gutin 2011, S. 222)	171
A.8	Basisstruktur des ACO Algorithmus (Petri 2006, S. 153)	178
A.9	Partikelschwarmoptimierung (vgl. Boussaïd et al. 2013, S. 100)	180

Abkürzungsverzeichnis

ACO	Ant Colony Optimization (dt.: Ameisenalgorithmus)
APO	Advanced Planning and Optimization
APS	Advanced Planning and Scheduling
ATP	Available to Promise
BaB	Branch-and-Bound
Charge	Bezeichnung für eine Produktionsmenge in der Lebensmittelindustrie, welche an einem bestimmten Stichtag erfolgt und mit einem definierten (garantierten) Mindesthaltbarkeitsdatum versehen worden ist.
CLSP	Capacitated Lot-Sizing Problem (kapazitiv beschränktes Losgrößenproblem)
CVRP	Capacitated Vehicle Routing Problem
DES	Discrete event simulation (diskrete, ereignisorientierte Simulation)
DES	Discrete event simulation (dt.: Diskrete, ereignisgestützte Simulation)
DP	Dynamic Programming
DS	Design Science
Ds., ds.	Durchschnittlich bzw. durchschnittlich
DSRM	Design Science Research Methodology
EA	Evolutionärer Algorithmus (engl.: Evolutionary Algorithm)
ECM	Epsilon-Constraint Methode
EI	Elite-Individuum
Enumeration	Enumeration wertet alle möglichen und gültigen Lösungen vollständig aus.
EP	Evolutionäre Programmierung (engl.: Evolutionary Programming)
EPEI	Every Part Every Intervall
ERP	Enterprise Resource Planning
ES	Evolutionsstrategie(n)
FMCG	Fast Moving Consumer Goods (dt.: schnelldrehende Konsumgüter)
GA	Genetischer Algorithmus (engl.: Genetic Algorithm)
GA-V1/V2/V3	In dieser Arbeit umgesetzte Varianten (1, 2, 3) des GA

GM	Die Methode der gewichteten Metriken
GM	Gewichtete Metriken (Methode)
GP	Goal Programming
GRASP	Greedy Randomized Adpatived Search Procedure
GSM	Gewichtete-Summen-Methode
GUI	Graphical user interface (dt.: grafische Benutzeroberfläche)
ILS	Iterierte Lokale Suche (engl.: Iterated Local Search)
Ind.	Individuum
IS	Insertion Search
IT	Informationstechnologie
JIT	Just-In-Time
Job Shop	Werkstattfertigung
JSP	Job Shop Scheduling Problem (dt.: Maschinenbelegungsplanung)
KMU	Kleine und mittelständische Unternehmen
KPI	Key Performance Indicator (dt.: Leistungsindikator)
KW	Kalenderwoche
LEH	Lebensmitteleinzelhandel
LO	Lexikographische Optimierung
MHD	Mindesthaltbarkeitsdatum
MIP	Mixed Integer Programming, Mixed Integer Problem
ML	Machine Learning
MOGA	Multi-Objective Genetic Algorithm
MOMPMP	Multi-Objective Multi-Product Multi-Period production planning problem (dt.: Problemklasse aus der Produktionsplanung bestehend aus mehreren Produkten und Perioden mit einer multikriteriellen Zielsetzung)
MRP	Material Requirements Planning (dt.: Materialbedarfsplanung)
ms	Millisekunde(n)
NB	Nebenbedingung
NP	Non-deterministic Polynomial-time: Komplexitätsklasse für Probleme, welche in nichtdeterministisch polynomieller Zeit lösbar sind
NP-Schwere	Probleme, die mindestens so schwer sind wie alle Probleme in NP
NP-Vollständigkeit	NP-vollständige Probleme stellen die am schwierigsten lösbaren Probleme innerhalb der Klasse NP dar und sind sowohl in NP als auch NP-schwer

NSGA	Non-dominated Sorting Genetic Algorithm
OR	Operations Research (dt.: Unternehmensforschung)
ORMA	Operations Research Modelling Approach
PFS(S)P	Permutation Flow Shop (Sequencing/Scheduling) Problem (dt.: Auftragsreihenfolgeproblem)
PGH	Produktionsglättungsheuristik
PO	Pareto-Optimalität
PORV	Production Output Rate Variation (Schwankung der Produktionsausstoßrate)
PPS	Produktionsplanungs- und Steuerungssystem
PRNG	Pseudo random number generator (Pseudozufallszahlengenerator)
PSO	Particle Swarm Optimization (dt.: Partikelschwarmalgorithmus)
RM	Referenzpunkt-Methode
Rollierende Planung	Periodisch kontinuierlich angepasste Planung
SA	Simulated Annealing (dt.: Simuliertes Glühen)
SGA	Standard Genetischer Algorithmus
SI	Smoothing Index (dt.: Glättungsindex)
SiRO	Simulation Rating Optimization
TM	Tchebycheff Methode
TOC	Theory of Constraints (dt.: Engpasstheorie)
TPS	Toyota Produktionssystem
Trade-Off	Abtausch, Kompromiss (-lösung)
TS	Tabu-Suche (engl.: Tabu Search)
TSP	Travelling Salesman Problem (dt.: Rundreiseproblem)
VNS	Variable Neighborhood Search
ZF	Zielfunktion (engl.: objective function, target function)
ZFA	Zielfunktionsauswertung


Kapitel 1
Einleitung

In der industriellen Praxis resultieren v.a. in der Lebensmittelindustrie in der Folge von zunehmenden Vertriebsaktivitäten bzw. Aktionen häufig Absatzschwankungen, welche das volatile kundenseitige Nachfrageverhalten abbilden. Durch die einerseits den Produktionssystemen immanente mangelnde Flexibilität seitens der Unternehmen bzw. den dazugehörigen nationalen bzw. rechtlichen Rahmenbedingungen sowie andererseits die in der Regel nicht stufenlos zu- und abschaltbare Verfügbarkeit von Produktionsressourcen ergeben sich dadurch erheblich höhere Flexibilitätskosten in der Produktion gegenüber einer Produktion mit einer relativ gleichmäßigen Produktionsausstoßrate. Derartigen Produktionsplanungsproblemen wird primär mit Methoden aus dem Fachbereich *Operations Research* (OR) begegnet.

Die Anfänge der OR genannten Disziplin gehen auf den Zweiten Weltkrieg zurück. Nach dem Krieg wurden die erfolgreichen OR Methoden auf andere Disziplinen übertragen, wo sie sukzessive weiterentwickelt worden sind. Die Standardmethoden, wie z.B. Lineare Optimierung, Dynamische Programmierung oder auch die Warteschlangentheorie, waren rasch etabliert und gut erforscht. Durch die Revolution des Computers konnten zunehmend größere Probleminstanzen erforscht und erfolgreich bzw. hinreichend genau gelöst werden. Die verschiedensten Problemstellungen aus dem OR können dabei entweder exakt oder auch approximativ, d.h. näherungsweise, gelöst werden. Während deterministische Verfahren auf einem reproduzierbaren Algorithmus aufgebaut sind, kombinieren stochastische Verfahren heuristische Ansätze mit der Auswahl von (zumindest teilweise) zufälligen Zahlen¹, um Optima zu finden (vgl. Hillier und Lieberman 2015, S. 1f.; Gendreau und Potvin 2019, S. 1ff.).

Die vorliegende Dissertation beschäftigt sich mit der Entwicklung einer Methodik aus dem Fachbereich des OR für eine optimierte Mittel- und Langfristplanung. Eine als *Produktionsglättung* bezeichnete Planung wird zur Vermeidung von Wechslen hochlastiger zu niedriglastiger Produktion verwendet und strebt eine gleichmäßige, kontinuierliche Produktionslast an (vgl. Tautrim 2014, S. 77). Die Anforderungen an das zu entwickelnde Optimierungsverfahren bestehen darin, mehrere Planungsdimensionen gleichzeitig zu berücksichtigen, wobei der Fokus auf eine im mittelfristigen Zeitverlauf möglichst gleichmäßige Werks- und Kapazitätsauslastung bzw. synchrone Versorgung der Produktion (Produktionsausstoßrate, Kapazitäten, Bestände, Rüstvorgänge) gelegt werden soll. Die Methodik, welche am Beispiel eines Unternehmens aus der fleischverarbeitenden Industrie entwickelt wird, soll darüber hinaus in der Lage sein, bestimmte Teile der Stammdaten rollierend zu bewerten und selbständig anzupassen. Dadurch entsteht ein adaptiver Verbesserungs-Regelkreis, wodurch eine insgesamt leistungsfähigere Methodik für Probleme im Bereich der Produktionsglättung mit multikriterieller Zielsetzung entsteht.

¹Reproduzierbarkeit kann durch die Verwendung von sogenannten *Pseudozufallszahlen* gewährleistet werden.

1.1 Motivation

Aktuelle Trends in Bezug auf globale bzw. volatile Märkte (vgl. Morawetz 2015, S. 1f.) sowie neue, global agierende Marktteilnehmer in Kombination mit der Individualisierung von Produkten (vgl. Schuh und Stich 2012, S. 65) bzw. ihrer Lebenszyklen erfordern flexiblere und reaktionsfähigere Produktionssysteme (vgl. Spath 2013, S. 4). Im Rahmen der Fraunhofer Studie „Produktionsarbeit der Zukunft - Industrie 4.0“ wird v.a. auf den zunehmend stärker schwankenden Bedarf an kurzfristiger personalseitiger Kapazitätsflexibilität hingewiesen. Dementsprechend benötigen Unternehmen innovative Lösungen, um zukünftig wettbewerbsfähig zu bleiben. Darüber hinaus lässt die Studie klar erkennen, dass die meisten Unternehmen Schwankungen im personalseitigen Kapazitätsbedarf in Bezug auf die Periodizität Woche (47%), gefolgt von Monat (27%), Tag (15%) und untertags (11%) angeben (Stichprobenumfang: 661 produzierende Unternehmen). Einige der zentralen Ergebnisse daraus leiten zielgerichtete und systematisch organisierte Flexibilität als Schlüsselfaktor für die zukünftige Produktionsarbeit ab (vgl. Spath 2013, S. 5-6).

Diese Ergebnisse liefern genügend Motivation für eine der Zeit entsprechende digitale und systematische Lösung dieser Herausforderung. Der Beitrag dieser Dissertation zu dieser Problematik besteht in der Entwicklung einer Methode, die den Fokus auf die Glättung der häufig absatzseitig entstandenen Auslastungsschwankungen in der Produktion unter Berücksichtigung bzw. Integration des werksseitig anfallenden Gesamtkapazitätsbedarfs setzt und diesen entsprechend reduziert. Die Produktionsglättung, die ein komplexes Optimierungsproblem mit mehreren Zielen und Zielkonflikten darstellt, ermöglicht prinzipiell geringere Flexibilitätskosten und eine optimierte Kapazitätsauslastung (vgl. Gorman und Brannon 2000, S. 173ff.). Da Flexibilisierungsmaßnahmen (Überstundenregelungen, Neukonfiguration des Fertigungslinienlayouts, etc.) entsprechend kostenintensiv sind, ist die Glättung der Produktion eine sinnvolle Maßnahme zur Minimierung der Flexibilitätskosten. Die globalen Märkte, die einerseits raschen Veränderungen mit zunehmender Produktkomplexität und andererseits einem hohem Individualisierungsgrad unterliegen, erfordern für reale industrielle Probleme schnell reagierende Produktionssysteme, die in der Lage sind, aufgrund häufiger Informationsaktualisierungen spezifisch angepasste und auch robuste Planungslösungen zu liefern (vgl. Kuhn et al. 2016, S. 418f.; Yavuz und Tufekci 2007, S. 3893ff.).

Schwankende und kurzfristige Bedarfe stellen die Hauptursache für Produktionsunterbrechungen dar. Der sogenannte *Peitscheneffekt* (engl.: *Bullwhip-Effekt*) beschreibt das Phänomen sich verstärkender Nachfrageschwankungen innerhalb einer Lieferkette, wobei der Endproduzent die niedrigsten Nachfrageschwankungen erfährt, während der*die unterste Lieferant*in in dieser Kette die höchsten Schwankungen trägt (vgl. Lödning 2008, S. 109), siehe auch Abb. 1.1. Diese Nachfrageschwankungen führen zu entsprechenden, teils verstärkten, Schwankungen in der Produktionsplanung, wobei ggf. mehrere Ursachen aufeinander treffen. Die häufigsten Ursachen betreffen Bestandsschwankungen auf Grund konstanter Durchlaufzeiten sowie Bestandsabbau und -aufbau in bestimmten Perioden, Änderungen oder Abweichungen der Prognose vom erreichten Wert sowie Kapazitätsrestriktionen, Preisschwankungen und Losgrößenbildung (vgl. Dickmann 2015, S. 186). Dieser Logistikkosten erhöhende Effekt vermindert die logistische Netzwerkleistung und hat insbesondere negative Auswirkungen auf die *Dimensionierung bzw. Auslastung der Kapazitäten*, weil eine mittlere Bedarfsanpassung zu Lieferengpässen an Perioden mit den Spitzenbedarfen führen wird. Eine Anpassung an den Maximalbedarf bedeutet hingegen hohe Auslastungsverluste. Dies ist insbesondere in der Fast-Moving-Consumer-Goods (FMCG) Branche bzw. speziell in der Lebensmittelindustrie essentiell, weil Kunden beim Einkauf im Lebensmitteleinzelhandel (LEH) besonders auf die Werbeform Flugblätter bzw. Prospekte achten (65%) und entsprechend danach ihre Einkäufe tätigen (vgl. Marketagent.com, 2017, S. 27). Entsprechend liefern die kostenwirksamen Auswirkungen des Bullwhip-Effekts weitere Motivationsgründe für die Findung einer Lösung, welche sich diesen Herausforderungen unter besonderer Berücksichtigung der kapazitiven Auslastungsverluste stellt.

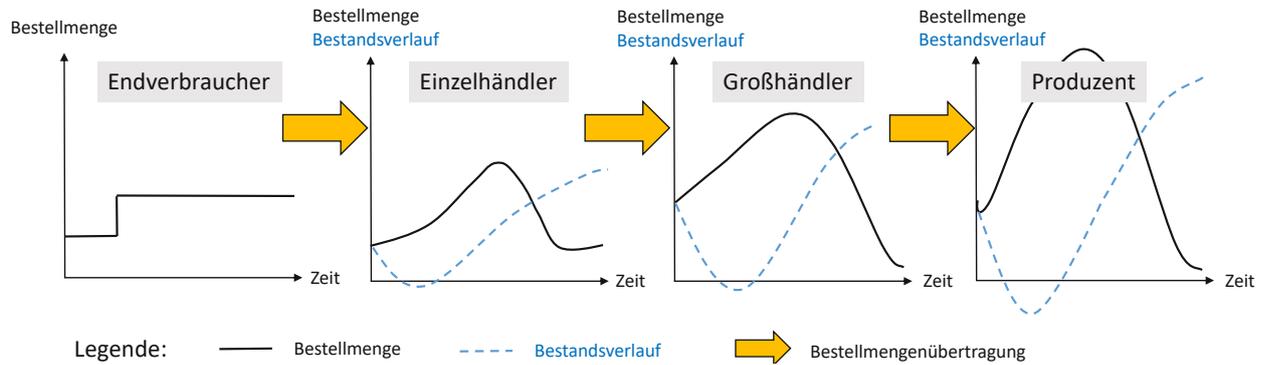


Abb. 1.1: Der Bullwhip-Effekt entlang einer Supply-Chain (Quelle: eigene Darstellung, in Anlehnung an Dickmann 2015, S.186 und Beer 2014, S. 25)

Auf Grund des*der taktbestimmenden Kund*in sind kleinere Unternehmen den Auswirkungen dieser Bedarfsschwankungen am stärksten ausgesetzt, da sie sich einerseits am Anfang der Lieferkette befinden, die Schwankungen hier entsprechend dem Bullwhip Effekt am stärksten ausgeprägt sind und das Handlungs- und Strategiepotezial auf Grund ihrer geringeren Marktmacht bzw. finanziellen Ressourcen sowie geringerer Mitarbeiter*innenkapazität stark eingeschränkt ist. Abb. 1.2 verdeutlicht, dass die auf Klein- und Mittelunternehmen (KMU) mit 150 - 1000 Mitarbeiter*innen wirkenden Nachfrageschwankungen einerseits deutlich größer sind als bei Großunternehmen (GU) mit > 1000 Mitarbeiter*innen und andererseits mit der Zunahme des Vorhersagezeitraums häufig ansteigen. Diese Schwankungen spiegeln sich in der Produktionsplanung im Zuge von verhältnismäßig stärker schwankend anfallendem Lager- und Kapazitätsbedarf deutlich wider und drücken die Kosteneffizienz insbesondere von KMU, die gleichzeitig einerseits weder über die Skaleneffekte noch die (finanziellen) Möglichkeiten von GU verfügen (vgl. Morawetz 2015, S. 2).

An diesem Punkt setzt diese Arbeit inhaltlich auf und verbessert die Wettbewerbsfähigkeit insbesondere von KMU durch eine geschickt geglättete Produktion. Die Kunst besteht nun darin, die in das Produktionssystem eintreffenden Aufträge möglichst derart auf die zur Verfügung stehenden Kapazitäten im System zu verteilen, sodass neben der Sicherstellung der Lieferfähigkeit als oberstes Ziel v.a. die Schwankungen in der Auslastung von Produktionsaggregaten und Personal möglichst gering gehalten werden.

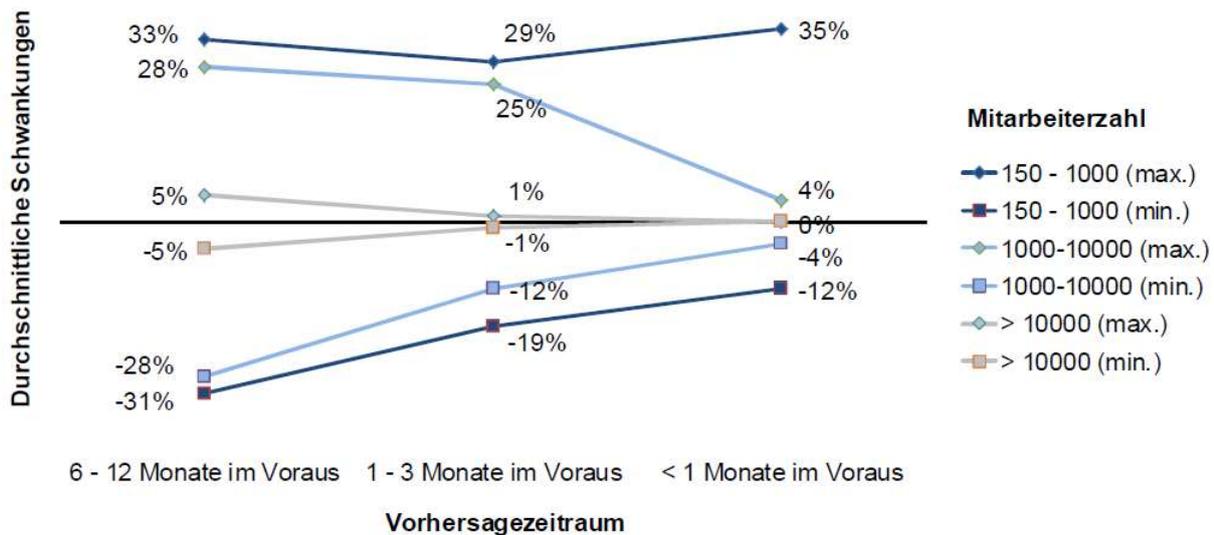


Abb. 1.2: Mengenschwankungen in Abhängigkeit von Unternehmensgröße bzw. Vorhersagezeitraum (Grundmann und Reinsch, 2006, S. 43)

Dementsprechend ist es ein Hauptziel dieser Arbeit, eine Methodik zu entwickeln, welche insbesondere die Schwankungen der Produktionsausstoßrate in der Mittel- und Langfristplanung am Beispiel der

Lebensmittelindustrie reduziert. Der LEH erlaubt bei denselben Produkten Preisunterschiede von bis zu 250% (vgl. Die Presse 2016). Die Periodizität des Endkund*innenbedarfs (siehe „Endverbraucher*in“ in Abb. 1.1) bzw. der durch vertriebsseitige Aktionen teils relativ hohe Preisunterschied führt bei vielen Produkten im „täglichen“ Warenkorb zu einer verstärkten Preissensibilität der Endkund*innen, alleine durch die häufige Konfrontation mit Endverbraucherpreisen im Geschäft. Darüber hinaus lässt sich der Grundbedarf eines jeden Menschen kaum substituieren, weil es sich häufig um sogenannte *Grundnahrungsmittel* handelt. Als weitere Probleme nennt Lödding Bestandsschwankungen bzw. Pönalen, mit welchen Lieferant*innen im Falle von Lieferengpässen konfrontiert werden. Ein hoher Bestand ist außerdem wegen des zu Grunde liegenden Mindesthaltbarkeitsdatums (MHD) gefährlich, welches bei Anlieferung im Lager auf die vertraglich geregelte *Restlaufzeit* (in Tagen) überprüft wird und im Falle einer Unterschreitung in der Regel abgelehnt wird. Das gilt insbesondere für den Lebensmittelhandel mit verderblichen Milch- und Molkereiprodukten bzw. Fleischprodukten, welche in dieser Dissertation den Anwendungsfall für die Methodik liefern. Der Überbestand hat also nicht nur eine höhere Kapitalbindung, sondern auch eine geringere Frische der biologischen Produkte bzw. im schlimmsten Fall sogenannte *Abschriften* zur Folge. Letzteres bedeutet, dass diese Charge nicht mehr an den Lebensmittelgroßhandel ausgeliefert werden darf. Entsprechend wird der*die Lieferant*in diese Charge entweder kostenpflichtig vernichten, vergünstigt *abverkaufen* oder kostenlos an Sozialmärkte ausliefern.

Lödding (vgl. Lödding 2008, S. 114f.) weist daraufhin, dass vergangenheitsbasierte Prognosemethoden zu schwankenden Mittelwerten und Standardabweichungen und damit verbunden veränderten Bestellmengen führen. Die zu Grunde liegenden sowohl hybriden (vgl. Smyl 2020, S. 75ff.) als auch klassischen Prognosemethoden, wie z.B. die Exponentielle Glättung 1. Ordnung oder das Holt-Winter Verfahren (vgl. Tratar et al. 2016, S. 162ff.), können die zu prognostizierende Zukunft teilweise nur wenig befriedigend prognostizieren, weil je nach eingestelltem Glättungsfaktor ein höherer Absatz in der jüngeren Vergangenheit höhere Prognosewerte für die Zukunft bedeutet. In der Realität kann diese Bestellung die verschiedensten Sachverhalte repräsentieren. Sie kann dazu dienen, den Bestand wieder aufzufüllen, ein Produktionslos zu bilden oder auf eine ganze Palette zu runden. Diesem Problem kann mit einer hohen Anzahl an Messwerten und einem entsprechend geringen Glättungsfaktor prinzipiell begegnet werden.

Ein Dilemma der Absatzplanung besteht grundsätzlich darin, dass diese in der Regel zeitlich lange im Voraus mit unvollständigen Informationen erstellt wird, weil dem*der Absatzplaner*in des Unternehmens notwendige Informationen fehlen, ob und wie lange sich eine vertriebsseitig geplante Aktion mit bestimmten Produkten seines Unternehmens mit Produkten eines anderen, z.B. konkurrierenden, Unternehmens überschneidet. Außerdem sind horizontale bzw. vertikale Abstimmungen (v.a. in Bezug auf Aktionspreise) zwischen Lieferant*innen und Händler*innen (wie REWE, SPAR, etc.) verboten, die Bundeswettbewerbsbehörde führt in Österreich (seit 2011) entsprechende Ermittlungen durch und veröffentlicht diese regelmäßig auf ihrer Homepage (vgl. Bundeswettbewerbsbehörde 2018) und aktualisiert die neu hinzugekommenen Entscheidungen in Bezug auf Verstöße inklusive Geldbußen (vgl. Bundeswettbewerbsbehörde 2014). Dementsprechend geht ein*e Absatzplaner*in bei Einplanung einer bestimmten zukünftig geplanten, vertriebsseitigen Aktion von realisierten Aktionspreisen in der Vergangenheit aus und erstellt entsprechend die Aktionsmengenplanung. Diverse weitere Effekte (Saison, Trends, Bestellverhalten, Preise von Artikeln der Konkurrenz, Wetter etc.) führen zu einer mit unvollständigem Wissen erstellten Absatzprognose. Zusammenfassend erschwert eine derart erstellte Absatzplanung einerseits, aber auch der Zeitversatz zwischen Planung und Zukunft andererseits, eine reibungslose Produktionsplanung.

Deshalb widmet sich diese Dissertation der Herausforderung, die Auswirkungen der diskutierten Effekte bzw. Planungsumstände systematisch im Zuge der Erstellung der wöchentlich rollierenden Produktionsplanung eines realen, industriellen Produktionssystems aus der Lebensmittelindustrie vorausschauend in der Produktion zu reduzieren bzw. zu *glätten*.

1.2 Allgemeine Problemstellung

Im Rahmen der Produktionsplanung treten verschiedene Problemstellungen aus dem Bereich des Operations-Research (OR) auf, welche im Zuge des laufenden operativen Betriebs auf Grund der verschärften Wettbewerbssituation möglichst effizient gelöst werden müssen. Während sich viele Arbeiten der Optimierung der Auftragsreihenfolge (vgl. Rossit et al. 2021; Mokhtari und Hasani 2017) annehmen, beschäftigt sich diese Dissertation mit der Minimierung der Produktionsausstoßschwankungen innerhalb eines kapazitiv beschränkten Produktionssystems, auf welches derartige Schwankungen auf Grund der Charakteristika (Betriebsgröße, Branchenzugehörigkeit, Artikelvielfalt, Saison-Effekte, Bestellverhalten, etc.) über den gesamten Planungshorizont hinweg betrachtet einwirken. Dies erfordert i.d.R. einerseits entsprechend kostenwirksame Anpassungen oder führt andererseits ab dem Wareneingang entlang der gesamten innerbetrieblichen Produktion zu erhöhten Beständen, die speziell in der Lebensmittelindustrie zu MHD-Risiken führen oder in Abschriften resultieren. Derartige Anpassungen stellen einerseits dauerhafte, kapazitive Umstellungen der Infrastruktur des Produktionssystems dar wie z.B. die Inbetriebnahme einer weiteren Anlage oder den Ausbau von Lagerkapazitäten. Andererseits umfassen kostenwirksame Anpassungen auch temporäre, d.h. zeitlich begrenzte, Maßnahmen bzw. Flexibilitätskosten in Form von Überstunden, Mehrschichten oder kurzfristig zusätzlich benötigtem Personal, um die aktuelle Produktionssituation zu bewältigen. Beide Optionen sind grundsätzlich kostenintensiv sowie teilweise wiederkehrend und lösen das Problem nicht an der Wurzel. Dieses Problem ist insbesondere für KMU auf Grund mangelnder Skaleneffekte kostenrelevant und betrifft v.a. Produktionssysteme mit einem hohen Anteil an zusätzlich aus Vertriebsaktivitäten resultierend aufkommenden Produktionsmengen. Außerdem betrifft das Problem v.a. Hersteller, die unter starken saisonalen Effekten leiden, welche in einem entsprechend hohen Saisonanteil der Gesamtproduktion münden. Diese beiden Faktoren verstärken die Kostenrelevanz der Problematik insofern, weil sich die genannten Produktionsausstoßschwankungen direkt proportional, d.h. anteilig zu diesen Effekten, verstärken.

Im Englischen wird diese Problemklasse unter dem Fachbegriff **Multi-Objective Multi-Product Multi-Period (MOMPMP) capacitated production planning problem** zusammengefasst (vgl. Karimi-Nasab und Konstantaras 2012, S. 481). Durch die Möglichkeit der Losgrößenbestimmung² im Zuge der Problemstellung wird es gleichzeitig auch als kapazitiv beschränktes Losgrößenproblem bzw. **(Multi-Item) Capacitated Lot-Sizing (MILS, CLS) problem** bezeichnet (vgl. Masmoudi et al. 2016, S. 301ff.; Mercé und Fontan 2003, S. 97ff.).

Diese beiden Problemklassen verfügen über einige Gemeinsamkeiten als auch Unterschiede. Gemäß Karimi-Nasab und Konstantaras (vgl. Karimi-Nasab und Konstantaras 2012, S. 481) sind Single-Product Multi-Period (SPMP) Probleme bereits NP-vollständig. Dies trifft auch auf Auftragsreihenfolgeprobleme zu. Bereits für Probleme im Bereich der Maschinenbelegungsplanung, sogenannte Job-Shop-Scheduling Probleme (JSP), wurde die NP-Schwere bei Vorliegen von mindestens 3 Maschinen gemäß Deb (vgl. Deb et al. 2014, S. 26) bestätigt³. Beide Problemklassen verfügen im Zuge der Optimierung ggf. über eine multikriterielle Zielsetzung. Charakteristika sind neben mehreren Zielen häufig widersprüchliche Zielsetzungen und teilweise unvergleichbare Einheiten, die Lösung besteht i.d.R. aus der Berechnung bzw. Auswahl der besten Lösung (vgl. Zimmermann 2008, S. 28f.). Einzelne Zielfunktionsbestandteile werden häufig in Kombination mit diskreter oder hybrider Simulation (vgl. Sobottka et al. 2017, S. 441ff.) realistisch gelöst, z.B. die Kriterien Liefertreue, Durchlaufzeit und Energieverbrauch. Der Einsatz von Simulation für die Bewertung der Güte einer mittel- bis langfristig geglätteten Produktionsplanung ist nicht unbedingt erforderlich, wenn die zuvor genannten Zielfunktionsbestandteile nicht explizit im Zuge der Optimierung benötigt bzw. betrachtet werden. Während simulationsbasiert optimierte Auftragsreihenfolgeprobleme einen relativ kurzfristigen Planungshorizont umfassen (z.B. einige Tage bis wenige Wochen), wird im Zuge von Verfahren für die Produktionsglättung v.a. ein mittel- bis langfristiger Planungshorizont angestrebt, sodass die Produktionsausstoßrate möglichst gleichmäßig unter Einhaltung aller erforderlichen modellspezifischen Nebenbedingungen erfolgt.

²siehe auch die nachfolgenden Unterabschnitte 2.2.4 - 2.2.5 zur Begriffserklärung.

³Die Begriffe NP, NP-vollständigkeit und NP-Schwere werden im Abkürzungsverzeichnis bzw. in 2.3.1.3 erklärt.

1.3 Ausgangspunkt der Lösungssuche für die vorliegende Problemstellung

Üblicherweise werden derartige **kombinatorische Problemstellungen** aus dem Bereich des OR mittels speziell angepasster **Heuristiken oder Metaheuristiken** gelöst. Kombinatorische Probleme bestehen aus einer (diskreten und endlichen) Menge von (Problem-) Instanzen, wobei eine Instanz ein Paar (S, f) darstellt. Während S mit der Menge aller zulässigen Lösungen den Suchraum darstellt, stellt f die zu minimierende oder maximierende **Zielfunktion** dar. Auf die Definition und Erklärung dieser grundlegenden Begriffe sowie Unterschiede und Gemeinsamkeiten dieser Problemklassen wird in den Kapiteln 2 und 3 näher eingegangen (vgl. Raidl und Chwatal 2008, F. 4-6).

Ausgangspunkt für die Erforschung von Problemen im Bereich der Produktionsglättung ist die als *Heijunka* bezeichnete, nivellierte Produktionsplanungsmethode, welche dieses Problem systematisch und ganzheitlich behandelt. Das Ziel von Heijunka ist eine möglichst gleichmäßige Einlastung von Produktionsaufträgen in das Produktionssystem (vgl. Ohno et al. 2013, S. 79). Die Methodik besteht aus zwei Planungsphasen, der grundsätzlichen Auslegung des (produktiven) Systems in regelmäßigen, mittelfristigen Intervallen, und der täglichen am Engpass ausgerichteten (Ablauf- und Reihenfolge-) Planung im operativen Betrieb mittels Kanban. Nach der Identifikation des Schrittmacherprozesses folgt die Berechnung des Kundentaktes \hat{T}_i , der sich im Wesentlichen aus der verfügbaren Produktionszeit geteilt durch die Nachfrage nach einem Produkt i ergibt. Die theoretisch optimale, durchschnittliche Zykluszeit T_i einer JIT-Produktion entspricht exakt diesem Takt, sodass $T_i = \hat{T}_i$ folgt. In der Praxis ist die Fertigung in Form von Produktionslosen mit Reserven (für Störungen, etc.) organisiert, sodass sich $T_i < \hat{T}_i$ ergibt (vgl. Veit 2010, S. 20f.; Tegel 2012, S. 21ff.).

Der Produktionsprozess selbst wird im Zuge von Heijunka harmonisiert, indem die Reihenfolge zwischen anspruchsvollen und weniger anspruchsvollen Produkten abgewechselt wird. Heijunka teilt eine Monatsmenge eines bestimmten Produktionsartikels in Tagesmengen auf, wobei diese Tagesmengen auch als EPEI (Every Part Every Interval) Wert bezeichnet werden. Die EPEI-Planung bestimmt den Abstand zweier Lose pro Produkt sowie die maximale Produktionsmenge in einem definierten Produktionszyklus (vgl. Veit 2010, S. 22). Der EPEI stellt damit einen Kennwert dar, der die Losgröße aufzeigt, in welcher Kund*innenbedarfe in der Produktion berücksichtigt werden. Dadurch werden Kund*innenaufträge betreffend Menge, Variantenspektrum und Ressourcen möglichst gleichmäßig in die Produktion eingelastet und ggf. mittels sogenannter *Heijunka-Tafeln* visualisiert (vgl. Dickmann 2015, S. 164f., S.206, S. 318 u. S.397; Korytkowski et al. 2013, S. 1554). Das grundsätzliche Ziel der Planung stellt ein ruhiger, möglichst kontinuierlicher und getakteter Materialfluss ohne Kapazitätsengpässe dar. Flexibilität wird auch als Puffer betreffend Zeit, Bestand und Anlagenkapazität verstanden. Mit zunehmender Flexibilität und den damit einhergehenden Kosten werden (teils verkettete) Störungen und Ausweichstrategien („*Snowball Effekt*“) sowie die grundsätzliche Problematik mit Mindestlosgrößen entschärft. Das *Flexibilitätsparadoxon* versucht dabei ein Optimum aus möglichst hoher Flexibilität bei vergleichsweise geringen damit verbundenen Kosten zu erreichen (vgl. Dickmann 2015, S. 184f.).

Moderne multikriterielle Optimierungsverfahren arbeiten mit problemspezifisch angepassten Metaheuristiken (z.B. Partikelschwarm-Optimierung, Genetische Algorithmen) oder Heuristiken, Fuzzy-Simulation als auch Künstlichen Neuronalen Netzen (vgl. Karimi-Nasab und Konstantaras 2012, S. 479ff.; Wari und Zhu 2016, S. 328ff.; Yavuz und Tufekci 2007, S. 3893ff.). Trotz der Effizienz dieser Verfahren wird der Betrachtungsfokus durch (deterministische) regelbasierte Heuristiken während der Suche stark eingeschränkt. Daher werden diese Verfahren häufig z.B. mit Metaheuristiken hybridisiert, wobei die Lösung der Heuristik als Ausgangspunkt für eine globale metaheuristisch-stochastische Suche verwendet wird (vgl. Gendreau und Potvin 2019, S. 1ff.; Yang 2015, S. 1-23). Diese rein mathematischen Modelle genügen nicht immer der Realität, weil bestimmte Teilziele manchmal nur durch eine detaillierte Simulation entsprechend realitätsnahe bewertet werden können und dadurch eine Kopplung der kombinatorischen Optimierung mit einer Simulation notwendig wird (vgl. Juan et al. 2015, S. 62ff.). Diese Kopplung führt häufig zu langen Rechenzeiten, welche für den industriellen Einsatz nicht praktikabel ist (vgl. Sihn et al. 2018, S. 447ff.). Kapitel 3 beinhaltet den für diese Arbeit relevanten Teil des Stands der Technik, welcher v.a. die zuletzt genannten modernen multikriteriellen Optimierungsverfahren sowie diverse Hybridisierungen systematisch diskutiert.

1.4 Im Rahmen dieser Arbeit behandelte Forschungsfragen

Im Zuge des vorangegangenen Abschnitts lassen sich nun zwei zentrale Forschungsfragen stellen, welche diese Arbeit beantwortet:

1. Wie gestaltet sich eine effizient implementierte, dynamische, multikriterielle Produktionsglättungsmethode, welche die Schwankungen der Produktionsausstoßrate effektiv minimiert?
 - a) Wodurch unterscheidet sich das zu optimierende Zielfunktionssystem im Rahmen einer Produktionsglättung von den typischen Zielgrößen (Liefertreue, Lagerhaltung, Rüstkosten, etc.) eines typischen Ablaufreihenfolgeproblems?
 - b) Wie unterscheidet sich eine dynamische Methodik für die Produktionsglättung von einer entsprechend statischen Methodik?
 - c) Welchen Nutzen bietet die entwickelte Optimierungsmethode mit dem Fokus auf die Produktionsglättung?
 - d) Auf welche Art und Weise wird die Lösungsgüte als auch die (Laufzeit-) Effizienz der in dieser Arbeit entwickelten Optimierungsmethode in Theorie und Praxis bewertet und validiert?
 - e) Welche konzeptionelle Unterschiede ergeben sich durch die jeweilige Verwendung der entwickelten Methodik beim einmaligen Einsatz gegenüber der rollierenden Anwendung im Zuge des operativen Einsatzes?
 - f) Welche Möglichkeiten bestehen für den Aufbau einer integrierten Datenanalyse zur laufenden Optimierung der Bestandsstammdaten und welches Potential lässt sich daraus ableiten?
2. Welche Anforderungen und Voraussetzungen sind für den erfolgreichen Einsatz der in dieser Dissertation entwickelten Methodik für multikriterielle Produktionsglättung in der fleischverarbeitenden Industrie erforderlich? Welche Eigenschaften und Merkmale sind für eine industrielle Anwendung in generalisierter Form zu erfüllen, damit diese Methodik sinnvoll anwendbar ist?

1.5 Forschungsmethode und inhaltlicher Aufbau

Das Vorgehen dieser Dissertation beruht auf der **integrierten Kombination zweier Forschungsmethoden**, welche im folgenden Abschnitt vorgestellt und diskutiert werden.

1.5.1 Forschungsmethodik

Für Informationssysteme, zu denen die in dieser Arbeit entwickelte Methodik zählt, hat sich die **Design Science Research Methodology (DSRM)** als geeignete Forschungsmethode im Bereich des Ingenieurwesens und der Informationstechnologie im Rahmen der Problemlösung herausgestellt. Design Science stellt Konzept und Forschungsmethode gleichermaßen in der Ausprägung eines Prozesses zur Verfügung, wobei das übergeordnete Forschungsziel praxisrelevante Problemlösungen beinhalten sollte.

Der **Design Science (DS) Ansatz** umfasst gemäß Prof. Hevner im Zuge seiner Anwendung folgende Richtlinien, welche man einhalten sollte (vgl. Hevner et al. 2004, S. 83 ff.):

1. „*Design as an Artifact*“: Im Rahmen des Design-Science Ansatzes wird ein brauchbares Produkt bzw. Modell identifiziert oder eine anwendbare Methodik entwickelt.
2. „*Problem Relevance*“: Es handelt sich im Rahmen der Problemstellung um ein entsprechend wichtiges und relevantes wirtschaftliches behandeltes Problem.
3. „*Design Evaluation*“: Nutzen, Qualität und Wirksamkeit sowie Anwenbarkeit des *Artefakts* (Produkt bzw. Problemlöser, Modell bzw. Methodik) wird fundiert ausgewiesen und bestätigt.

4. „*Research Contributions*“: Die Ergebnisse sind in dem jeweiligen Bereich des *Artefakts* klar, überprüfbar und liefern einen verifizierbaren Beitrag.
5. „*Resarch Rigor*“: Aufbau und Evaluierung der Methodik erfolgen, basierend auf der jeweiligen Anwendung, entsprechend gründlich und umfassend.
6. „*Design as a Search Process*“: Es handelt sich um einen iterativen, heuristischen Verbesserungsprozess, wobei im Rahmen einer satisfizierenden Lösung gezeigt wird, dass sich die gefundene Lösung nahe am Optimum befindet.
7. „*Communication of Research*“: Die Ergebnisse sind sowohl technologisch als auch wirtschaftlich repräsentativ, d.h. sie ziehen einerseits eine technologische Weiterentwicklung nach sich bzw. versprechen andererseits auch einen wirtschaftlich gerechtfertigten Nutzen.

Die Problemrelevanz ist gemäß den Abschnitten 1.1 - 1.2 gegeben. Die sechste Richtlinie, der heuristische Verbesserungsprozess, stellt einen iterativen Designprozess dar, welcher essentiell ist, um ein Problem effektiv zu lösen. Der Problemlösungsprozess unter Nutzung aller verfügbaren Mittel gilt als beendet, wenn die geforderten Ziele erreicht sind und gleichzeitig die aus der Umgebung geforderten Bedingungen eingehalten werden. Die Iterationen bestehen einerseits aus dem Relevanzzyklus, welcher die Anforderungen im Zuge von Feldversuchen bzw. Demonstratoren kontinuierlich überprüft. Andererseits werden im Zuge des rollierenden Entwicklungszyklus die Design-Artefakte entwickelt und evaluiert. Darüber hinaus ergänzt der Härtezyklus stetig die dazugehörige Wissensbasis (vgl. Hevner Alan R. 2007, S. 87-92). Diese drei Zyklen stellen die Erweiterung der DSRM von Hevner (vgl. Hevner Alan R. et al. 2004, S.80) um eine zyklische Betrachtung dar, siehe auch Abbildung 1.3.

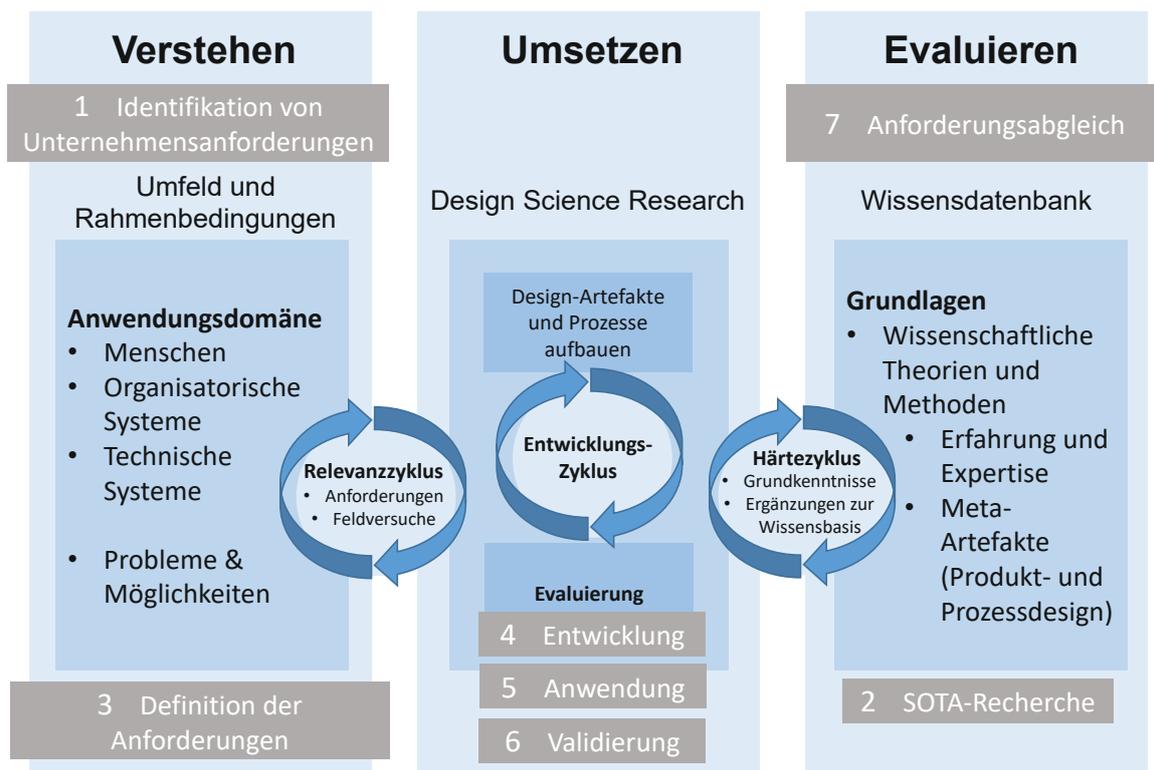


Abb. 1.3: Erweiterter Design Science Ansatz (Quelle: eigene Darstellung in Anlehnung an Hevner Alan R. 2007, S. 1-5)

Dementsprechend folgt der inhaltliche Aufbau der Arbeit diesen Richtlinien, wobei die fünfte Richtlinie, der **Aufbau und die Evaluierung der Methodik**, anhand des **Operations Research Modeling Approach (ORMA)** umgesetzt wird (vgl. Hillier und Lieberman 2015, S. 10). Diese Vorgehensweise

besteht üblicherweise aus den folgenden (auch überlappenden) Phasen innerhalb einer fundierten Arbeit im Bereich des Operations Research:

1. Problemdefinition und Datenaufbereitung.
2. Formulierung eines mathematischen Modells, welches das Problem geeignet repräsentiert.
3. Computerbasierte Entwicklung eines *Algorithmus*⁴, um Lösungen aus dem Modell zu generieren.
4. Umfangreiche Validierung und ggf. Verfeinerung des Modells.
5. Vorbereitung des Modells auf die (ggf. kontinuierliche bzw. rollierende) Anwendung.
6. Implementierung.

Im Zuge der Algorithmenentwicklung (Phase 3) stellt das algorithmische Denken einen essentiellen Lernprozess dar, welcher es ermöglicht, Prozesse in Programmiersprachen zu formalisieren. I.d.R. ist die allgemeine Datentransformation, beschrieben in Worten, leichter als die schrittweise Formulierung und der konstruktive Aufbau der entsprechenden Logik. Prokop liefert dafür mit dem Dreieckstausch zweier Variablen unter Verwendung einer Hilfsvariablen ein plakatives Beispiel (vgl. Prokop 2014, S. 61f.).

Nach der Implementierung ist ggf. eine nochmalige Validierung bzw. Abstimmung mit dem*der Stakeholder*in erforderlich, speziell in Bezug auf die kontinuierliche Ausführung der Methodik bzw. der Ergebnisse, welche im Zuge der periodischen Ausführung je nach Datensatz stärker schwanken können.

Goldratt's Theory of Constraints: Im Rahmen dieser Arbeit wird zur Problemdefinition (ORMA Phase 1) Goldratt's *Theory of Constraints* (TOC) bzw. *Engpasstheorie* als methodologische Basis ausgewählt (vgl. Lenort et al. 2012, S. 225). TOC beruht für Probleme im Bereich der Produktionsplanung und -steuerung auf der Annahme, dass i.d.R. kein Produktionssystem bereits so gut ausbalanciert ist, dass es keinen Engpass enthält. Stattdessen wird von schwankenden Kapazitätsengpässen ausgegangen, die als Nebenbedingungen in das konkret betrachtete Problem einfließen. Die Methode umfasst 5 *Fokussierungsschritte*, beginnend von der (1) Identifikation des (Kapazitäts-) Engpasses sowie der (2) Entscheidungen über die Möglichkeiten der maximalen Auslastung dieses Engpasses, wobei sämtliche weiteren Maßnahmen und Unterstützungsprozesse (3) der Auslastungsentscheidung aus dem 2. Schritt untergeordnet behandelt werden. In einem 4. Schritt wird der Engpass behoben, wodurch ein neuer Engpass (5) entsteht und man, gemäß eines von Verbesserungen geprägten Kreislaufs, zurück zu Schritt 1 springt.

Aufstellen eines geeigneten mathematischen Modells: Nach der Definition des konkret vorliegenden Problems (Phase 1) in Abstimmung mit dem*der jeweiligen Entscheidungsträger*in (z.B. bei einem Unternehmen) erfolgt der inhaltliche Kern: Die Formulierung dieses Problems in Form eines mathematischen Modells in der Phase 2. Dieses beinhaltet ein Set bestehend aus mathematischen (Un-) Gleichungen und beschreibt dadurch die inhaltlich behandelten Schwerpunkte. Wenn beispielsweise n unterschiedliche quantifizierbare Entscheidungen getroffen werden müssen, sind diese als **Entscheidungsvariablen** (z.B. x_1, x_2, x_3, \dots) in das Modell integriert und müssen entsprechend bestimmt werden. Die **Zielfunktion** beinhaltet all diese Entscheidungsvariablen in Form einer resultierenden Funktion (z.B. $ZF = 3x_1 + 2x_2 - x_3 + \dots + x_n$). Alle im System vorliegenden relevanten **Nebenbedingungen** werden in Form von Gleichungen oder Ungleichungen formuliert. Typischerweise stellen z.B. kapazitive Restriktionen Ungleichungen der Form $\{<, \leq\}$ dar, während z.B. Losgrößen oftmals Untergrenzen darstellen, die lediglich überschritten werden dürfen, also die Form $\{>, \geq\}$ annehmen. Wenn bestimmte Bedingungen exakt erfüllt werden müssen, so wird dies in Form von Gleichungen ausgedrückt. Die Konstanten in diesen Nebenbedingungen werden auch als **Parameter** bezeichnet.

⁴ *Algorithmen* stellen schrittweise Anleitungen zur Problemlösung dar, welche eine Sequenz von Instruktionen bilden und dabei systematisch vorzunehmende Datenmodifikationen erlauben. Sie sind beliebig aufteilbar, sodass Programme aus einer Reihe dieser Handlungsanweisungen zur Lösung diverser Teilprobleme bestehen (vgl. Prokop 2014, S. 61). Siehe auch Definition 1.

Die Zielfunktion wird i.d.R. entweder in Bezug auf Kosten minimiert oder den Umsatz bzw. Gewinn maximiert (vgl. Hillier und Lieberman 2015, S. 13).

Der entscheidende Prozess im Rahmen der Erstellung des Modells liegt in der **Erhebung von relevanten Daten bzw. sich daraus ergebenden Informationen als Grundlage für die richtige Wahl der Parameter bzw. Entscheidungsvariablen und Nebenbedingungen**. Manchmal können Parameter nur geschätzt bzw. im Zuge einer Sensitivitätsanalyse bestimmt werden. Der große Vorteil von mathematischen gegenüber verbalen Modellen liegt darin begründet, dass die erstgenannten ein Problem detaillierter bzw. konkreter beschreiben. Dadurch wird einerseits die Problemstruktur nachvollziehbarer als auch die Problemklasse, also die Schwierigkeit bzw. Komplexität dieses Problems, nachvollziehbarer. Außerdem formt ein mathematisches Modell eine Brücke zur EDV-gestützten automatisierten Anwendung. Ein weiterer fundamentaler Schritt besteht im Aufstellen einer geeigneten Zielfunktion, welche die Güte der ermittelten Lösung im Einklang mit der betrieblichen Realität bewertet. Diese Güte, häufig aus mehreren Teilzielen zusammengesetzt, kann sowohl eine abstrakte dimensionslose als auch direkt messbare Größe (Kosten, Umsatz, Gewinn) darstellen (vgl. Hillier und Lieberman 2015, S. 14).

Die mathematische Modellbildung erfolgt in dieser Arbeit in Abschnitt 5.3.

Im Zuge der Modellentwicklung bietet es sich an, mit einem relativ einfachen Modell und dazugehörigem Optimierungsverfahren (Phasen 2 - 3) zu starten und dieses dann sukzessive um weitere Inhalte zu erweitern, bis das finale Modell vorliegt. Dieser *Modell-Anreicherung* genannte Prozess (Phase 4) ist nur anwendbar, solange das Modell auch nachvollziehbar ist bzw. bleibt. Dabei muss stets eine Abwägung getroffen werden zwischen der Modell-Genauigkeit und seiner Nachvollziehbarkeit (vgl. Hillier und Lieberman 2015, S. 14). Im Rahmen der Entwicklung einer dazugehörigen Software kann dieser Kompromiss auch durch eine objektorientierte, transparente Lösungsfindung erfolgen. Im Rahmen dieser Lösungsfindung wird jeder errechnete Zwischenschritt als Teil der gesamten Lösung ausgegeben. Es empfiehlt sich, diese Strategie auf das vorliegende Problem anzuwenden, um maximale Transparenz im Vorgehen gewährleisten zu können. Dies schließt (kommerzielle) Solver-Lösungen grundsätzlich aus, die diese (schrittweise bzw. grafische) Transparenz nicht gewährleisten. Abschließend wird das finale Modell, nach einer entsprechend intensiven Validierung, auf die (kontinuierliche) Anwendung vorbereitet, an die EDV-Struktur angebunden und implementiert (Phase 6).

1.5.2 Inhaltlicher Aufbau

Kapitel 1 führt an die Motivation für das Thema heran und erläutert die allgemeine Problemstellung und Zielsetzung der Arbeit. Außerdem werden einerseits die durch diese Arbeit zu beantwortenden Forschungsfragen aufgestellt sowie der Einsatz der entsprechenden Forschungsmethode begründet.

In Kapitel 2 werden grundlegende Begriffe aus den Bereichen Produktions- und Kapazitätsplanung sowie Operations Research definiert und erklärt. Darauf aufbauend wird die Arbeit innerhalb dieser beiden Bereiche entsprechend eingeordnet und der Betrachtungsbereich eingegrenzt. Das vorliegende MOMPMP kann in zwei Teile untergliedert werden, einerseits in den Bereich Multikriterielle Optimierung, als auch in den zweiten planungsspezifischen Teil, welcher sich mit dem Charakter des Problems beschäftigt. Aus dem ersten der beiden Bereiche folgt die unmittelbare Auswahl einer geeigneten Lösungsmethodik. Kapitel 2 stellt außerdem das notwendige theoretische und konzeptionelle Verständnis für Probleme aus dem Bereich der nichtlinearen multikriteriellen Optimierung zur Verfügung, welches für die weiteren Kapitel dieser Arbeit erforderlich ist.

Kapitel 3 bietet einen fundierten Überblick über den aktuellen und relevanten Stand der Technik in Bezug auf Optimierungsverfahren mit einer fokussierten Betrachtung auf multikriterielle nichtlineare Probleme. Neben deterministischen bzw. mathematisch exakten Methoden werden v.a. heuristische bzw. stochastisch basierende, metaheuristische Optimierungsverfahren als auch problemspezifische (Konzepte für) Heuristiken und Hybridisierungen besprochen, und deren Vor- und Nachteile diskutiert. Abgerundet wird Kapitel 3 mit einer Diskussion der aktuell in der Literatur verfügbaren Ansätze mit der größten thematischen Relevanz sowie einer vergleichenden Bewertung der Leistungsfähigkeit der vorgestellten

Verfahren hinsichtlich ihrer Eignung bzw. Mängel bezüglich der Erfüllung der Anforderungen an eine Methodik für die Produktionsglättung. Daraus resultierend werden die Schlussfolgerungen in Bezug auf die Forschungslücke im Stand der Technik abgeleitet.

Im Zuge von Kapitel 4 erfolgt, basierend auf den Produktionsabläufen des konkret betrachteten Produktionssystems, die konkrete Problembeschreibung inklusive der sich daraus ergebenden Anforderungsübersicht und methodischen Problemabgrenzung. Die Anforderungen an die zu entwickelnde Methodik werden dabei in Form einer modularen Übersicht zur Verfügung gestellt.

Kapitel 5 beschäftigt sich mit der Entwicklung der Produktionsglättungsmethodik. Zuerst erfolgt die Modellbildung, welche das Problem in Form eines mathematischen Modells vollumfänglich inklusive Zielfunktionssystem und aller vorliegenden Planungsrestriktionen beschreibt und ausformuliert. Im Anschluss an die Modellbildungsphase werden die hybriden Optimierungsalgorithmen für das zuvor aufgestellte Problem entwickelt. In diesem Kapitel wird auch der Bezug auf die Forschungslücke(n) hinsichtlich der Lösung derartiger Probleme hergestellt.

In Kapitel 6 erfolgt die Auswertung und Interpretation der Ergebnisse bezüglich des entwickelten Optimierungsverfahrens inklusive einer theoretischen und praktischen Bewertung der Laufzeitkomplexität. Darüber hinaus erfolgt eine Nutzenbewertung.

In Kapitel 7 werden zuerst die Schlussfolgerungen aus dem Vergleich des entwickelten Verfahrens mit dem aktuellen Stand der Technik gezogen bzw. ein Abgleich mit den in Kapitel 4 gestellten Anforderungen an das Verfahren durchgeführt. Anschließend werden die in Kapitel 1 gestellten Forschungsfragen beantwortet. Schlussendlich wird ein Ausblick bezüglich möglicher Erweiterungen der entwickelten Produktionsglättungsmethodik gegeben. Abschließend wird der weitere Forschungsbedarf skizziert.

Abbildung 1.4 folgt dem im vorherigen Abschnitt beschriebenen Design-Science (DS) Ansatz mit spezieller Berücksichtigung des ORMA und gibt eine schematische Übersicht über den Aufbau dieser Arbeit. Der Relevanzzyklus wird in Kapitel 4 adressiert. Der Entwicklungszyklus ist teilweise in Kapitel 5 im Zuge der methodischen Entwicklung integriert und weiters in Kapitel 6 im Zuge der Validierung. Die Ergänzungen zur Wissensbasis werden in Kapitel 7 festgehalten.

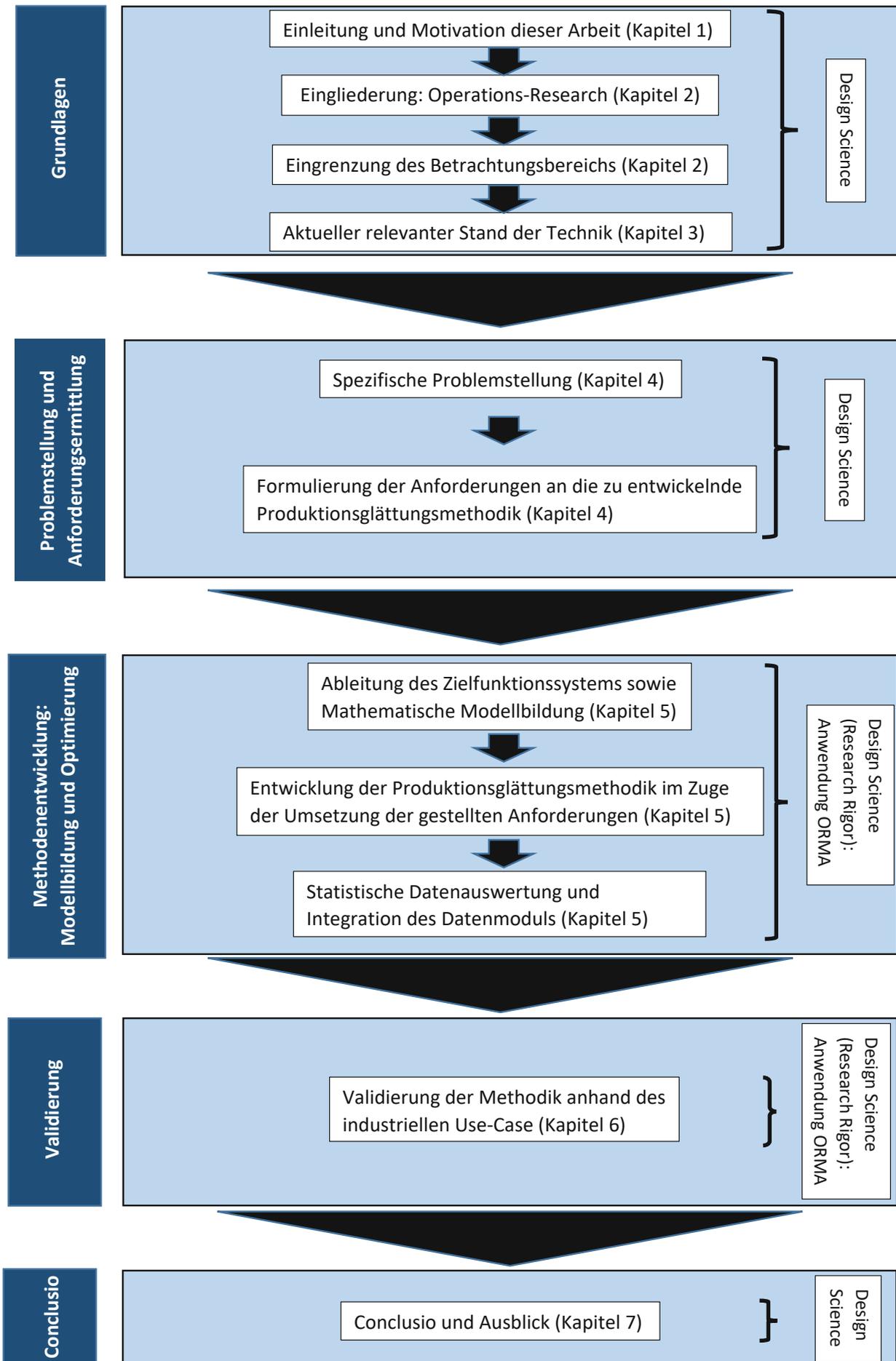


Abb. 1.4: Aufbau dieser Arbeit (Quelle: eigene Darstellung)

2

Kapitel 2

Grundlagen von Produktion und Optimierung

Das Thema der vorliegenden Dissertation fällt in das Fachgebiet des Operations Research, angewandt auf ein konkretes reales Problem aus der Produktionsplanung. Dieses Kapitel erläutert die notwendigen Begriffsdefinitionen sowie die Grundlagen zur Planung und Optimierung von Produktion und Logistik innerhalb eines Produktionssystems. Danach werden die mathematischen Grundlagen, Begriffe und Definitionen sowie Konzepte diskutiert, damit diese in den folgenden Kapiteln einsetzbar sind. Nach diesem Exkurs in die Grundlagen der nichtlinearen multikriteriellen Optimierung wird die Arbeit innerhalb des Operations Research entsprechend eingegliedert sowie der Betrachtungsbereich in Bezug auf die Charakteristika der Branche und die relevanten Prozesse der Supply-Chain bzw. die Planungsaufgabe und den Planungshorizont sowie der damit verbundenen Kostenstruktur eingegrenzt.

2.1 Begriffsdefinitionen

Der Fachbegriff des Operations Research wird, verwurzelt im militärischen Bereich, ins Deutsche mit „*Unternehmensforschung*“ oder „*mathematische Optimalplanung*“ übersetzt (Ellinger et al. 2003, S. 1f.). Eine aus betrieblicher Sicht geeignete Definition liefert Müller-Merbach (Müller-Merbach 1992, S. 1), nämlich „*die Anwendung mathematischer Methoden zur Vorbereitung optimaler Entscheidungen*“ bzw. die „*modellgestützte Vorbereitung von Entscheidungen zur Gestaltung und Steuerung sozio-technischer Systeme*“. Zimmermann definiert Operations Research als „... *die Anwendung mathematischer; statistischer und formallogischer Methoden bei der Analyse, Beschreibung und Prognose betrieblicher und zwischenbetrieblicher Vorgänge zum Zwecke der Entscheidungsfindung*“ (Zimmermann 2008, S. 9). Weitere Definitionen enthalten im Wesentlichen die drei folgenden Charakteristika des OR (vgl. Müller-Merbach 1992, S. 2):

1. Entscheidungsvorbereitung für den*die Stakeholder*in.
2. Anstreben einer möglichst optimalen Entscheidung.
3. Verwendung von (zumeist) computerbasiert umgesetzt mathematischen bzw. (meta-) heuristischen Methoden in dazugehörigen mathematischen Modellen.

Der*die Stakeholder*in stellt im Bezug auf das vorliegende Problem die Geschäftsführung im Unternehmen dar. Diese möchte den Planungsprozess digitalisieren. Die Methoden des Operations Research dienen der logischen Entscheidungsvorbereitung und als Unterstützung für die Planungsabteilung, welche ein entsprechendes Feedback bezüglich der Leistungsfähigkeit und Einsatzbereitschaft der Methodik an die Geschäftsführung abgibt.

2.2 Grundlagen zur Planung von Produktion und Logistik

Gemäß Dangelmaier (vgl. Dangelmaier 2009, S. 1) wird die **Produktion** als gezielter betrieblicher Umwandlungs- und Transformationsprozess definiert, wobei eine Menge bzw. Kombination aus materiellen und immateriellen Input-Faktoren für einen andersartigen Output eingesetzt wird, und stellt zusammen mit der Beschaffung den Bereich der betrieblichen Leistungserstellung dar (vgl. Veit 2010, S. 7). Diese **Elementarfaktoren** werden nach der Produktionstheorie in **Produkte** und **Produktionsfaktoren** (eingesetzte Faktoren → *Enabler*) differenziert, wobei die Produktionsfaktoren weiter zerlegt werden in sogenannte *primäre* und *sekundäre* Produktionsfaktoren. Nach dieser Definition finden primäre Produktionsfaktoren von außen Eingang in das Produktionssystem, während sekundäre Produktionsfaktoren innerhalb eines Betriebes erstellte und wiedereingesetzte *Zwischenprodukte* darstellen (vgl. Dangelmaier 2009, S. 34). Die **primären Produktionsfaktoren** werden außerdem als *Verbrauchs-* bzw. *Repetierfaktoren* bezeichnet, während die **sekundären Produktionsfaktoren** zur Gruppe der *Gebrauchsfaktoren* zählen. Eine andere Differenzierung unterscheidet bei den eingesetzten Produktionsfaktoren zwischen Werkstoffen, Betriebsmitteln und Arbeitskräften und fasst die beiden letztgenannten Kategorien zu **Potenzialfaktoren** zusammen (vgl. Dyckhoff und Spengler 2010, S. 17f.).

Ein **Produktionssystem** ist durch eine technisch, organisatorisch und kostenseitig selbständige Allokation von Potenzialfaktoren zu Produktionszwecken definiert und produziert Güter in Form eines ökonomischen Systems innerhalb eines bestimmten Zeitraums (vgl. Dangelmaier 2009, S. 3).

2.2.1 Produktionsplanung und -steuerung

Diese Dissertation beschäftigt sich mit der Entwicklung einer Methodik für eine optimierte Produktionsplanung. Deshalb wird der Begriff Produktionsplanung im Rahmen dieser Arbeit erläutert.

Das Kernziel der **Produktionsplanung** besteht in der optimalen Kombination der Produktionsfaktoren Mensch, Betriebsmittel, Produktionsanlage und Produktionsmethode, um ein gegebenes Formalziel, beispielsweise Gewinnmaximierung oder Liefertermintreue, des Produktionssystems bestmöglich zu erreichen (vgl. Veit 2010, S. 7f.). Während die Kombination der Produktionsfaktoren i.d.R. eindeutig auf Basis einer Stückliste bzw. einem gemäß Arbeitsplan definierten Produktionsablauf dem jeweiligen Endprodukt zugeordnet ist, verbleibt in der definierten Reihenfolge und Terminierung der Durchführung der Produktionsaufträge ein dispositiver Spielraum in Bezug auf die Finalisierung bzw. Fertigstellung der (Zwischen-) Produkte. Außerdem sind diverse Leistungen durch alternative Prozesse (gewählter Grad der Automatisierung), alternative Ressourcen (z.B. Maschinen) oder auch eine alternative Produktionslogik (Serienfertigung, Auftragsfertigung, synchronisierte Produktionskonzepte) erstellbar. Dadurch kann die Ablauforganisation entsprechend organisiert werden. Dementsprechend sind Teile dieser Produktionsfaktoren, z.B. die an Maschinen über Taktzeiten bzw. Bearbeitungs- und Rüstzeiten gebundene Herstellungskosten, Konstanten, während andere Teile bezüglich der räumlichen, zeitlichen und mengenmäßigen Erbringung der betrieblichen Leistungserstellung kostenvariabel sind (vgl. Veit 2010, S. 8).

Auf Grund der Komplexität moderner Produktionssysteme, die einzelne Entscheidungen schwierig in Bezug auf übergeordnete Unternehmensziele messbar machen, werden zumeist (individuell angepasste) abgeleitete Kennzahlen als Indikatoren zur Messung der Effizienz und Effektivität herangezogen. Die Leistungsfähigkeit eines Produktionssystems wird v.a. an der Liefererfüllung bzw. Liefertermintreue sowie der produzierten Qualität gegenüber dem*der Kund*in gemessen. Die Kosten sind v.a. durch Prozess- und Bestandskosten vorbestimmt. Der Zusammenhang zwischen Kosten- und Leistungsseite schlussfolgert ein größeres Leistungspotential und höhere Bestände auf Basis der Verfügbarkeit größerer Kapazitäten. Zwischen den Kostenarten gibt es Wechselwirkungen, z.B. in Form von geringeren Prozesskosten durch besser ausgelastete Maschinen, welche bei gleichem Leistungsniveau zu steigenden Beständen führen (vgl. Veit 2010, S. 8).

2.2.2 Ebenen der Kapazitätsplanung

Für eine möglichst optimale Auslastung der zur Verfügung stehenden Produktionsressourcen ist eine entsprechende Planung notwendig. Zur Erfüllung unterschiedlicher Aufgaben unter Berücksichtigung eines dynamischen Informationsstands haben sich verschiedene Planungsaufgaben entwickelt. Krajewski und Ritzman unterscheiden beispielsweise zwischen Langfrist- und eine Kurzfristplanung. Während sich aus der Langfristplanung (Betrachtungshorizont > 2 Jahre) Investitionsmaßnahmen in Bezug auf neue Anlagen bzw. Werkserweiterungsmaßnahmen ableiten lassen, widmet sich die Kurzfristplanung der optimal notwendigen Mitarbeiter*innenanzahl in Zusammenspiel mit dem dazugehörigen Automatisierungsgrad, Maschinenauslastung, Schichtplänen und Beständen (vgl. Morawetz 2015, S. 13f.). Gemäß des Übergangs von Planung zu Steuerung nach Buzacott ist die Steuerungsebene auch für die Steuerung der Kapazitäten verantwortlich. Nach Lödding verfolgt die Kapazitätsplanung die Durchsetzung eines Produktionsprogramms mit dem im Voraus ermittelten Kapazitätsangebot, während die Kapazitätssteuerung über den tatsächlichen Einsatz darüber entscheidet. Die Kapazitätssteuerung entscheidet allgemeiner das den Arbeitszeiten zu Grunde liegende Schichtmodell, die Zuordnung von Tätigkeiten zu einem*einer konkreten Mitarbeiter*in (Rüsttätigkeiten, Kommissioniertätigkeiten, Ver- und Entsorgungstätigkeiten, etc.) und legt damit Maßnahmen der Kapazitätsflexibilität fest (vgl. Lödding 2008, S. 461). Diese Arbeit unterscheidet die Kapazitätsplanung und -steuerung in die folgenden drei Ebenen (vgl. Heizer und Render 2008, S. 288; Morawetz 2015, S. 14):

- Die **langfristige Kapazitätsplanung** mit strategischem Charakter und mindestens 12 Monate bzw. mehrere Jahre Planungshorizont in der Zukunft. Diese ist Grundlage für Investitionsentscheidungen in (Personal- und Maschinen-) Ressourcen, Fabriken, Logistik- und Distributionsstrukturen und stützt sich auf Markt- oder Absatzprognosen unter Einbezug von aktuellen und zukünftigen technologischen Trends.
- Die **mittelfristige Kapazitätsplanung**, welche i.d.R. ein bis 12 (bzw. maximal 18) Monate Planungshorizont beinhaltet und sowohl auf Vertriebsprognosen und Kund*innenforecasts als auch konkreten Kund*innenbestellungen bzw. konkreten Abrufen (in der Zukunft) besteht. Auf dieser Ebene wird nach einer ersten Kapazitätsbilanzierung über Maßnahmen zur Anpassung der Plankapazität entschieden.
- Die **Kapazitätssteuerung** überwacht die Kapazitäts- und Terminplanung und ergreift nach Notwendigkeit entsprechende Maßnahmen zur (kontinuierlichen) **kurzfristigen Kapazitätsanpassung**.

Die im Zuge dieser Arbeit entwickelte Methodik zielt vor allem auf die Optimierung der mittelfristigen Kapazitätsplanung ab und bedient sich dabei überwiegend einer Kombination aus Methoden des Operations Research, der Simulation und der statistischen Datenanalyse. Die mittelfristige Kapazitätsplanung zielt dementsprechend auf eine Optimierung der wöchentlichen Produktionsplanung der nächsten 78 Wochen ab.

2.2.3 Kapazitätsanpassungsstrategien

Unabhängig von einer quantitativen, zeitlichen oder intensitätsmäßigen Kapazitätsanpassung wird prinzipiell zwischen den drei Kapazitätsanpassungsstrategien Emanzipation, Synchronisation und partielle Emanzipation (vgl. Abele et al. 2008, S. 585ff.) unterschieden. Die Synchronisation stellt dabei eine in feineren Stufen aufgelöste partielle Emanzipation dar. Abbildung 2.1 stellt das jeweilige Maß an Verschwendung bei entsprechender Unterversorgung bzw. Überproduktion in den beiden Strategien Emanzipation und partieller Emanzipation gegenüber, während bei der Synchronisation keine Verschwendung vorgesehen ist.

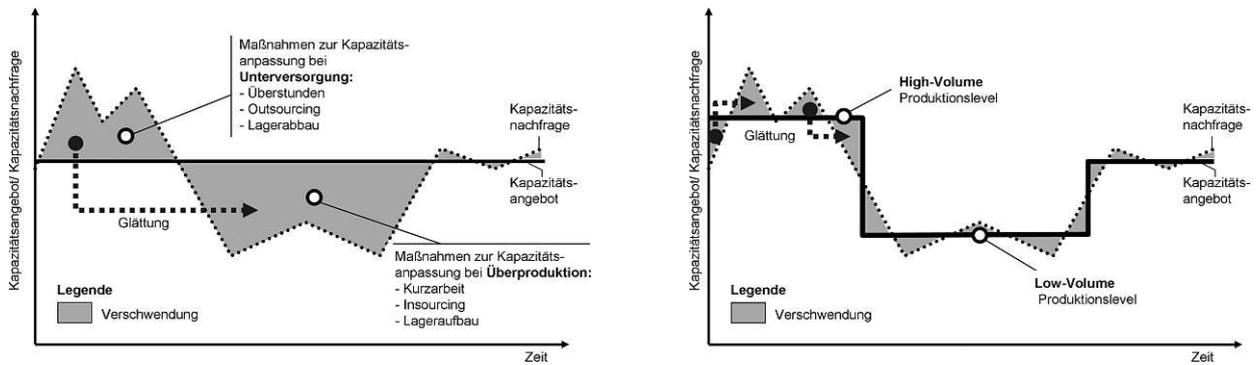


Abb. 2.1: Gegenüberstellung zweier Kapazitätsanpassungsstrategien : (links) Emanzipation bzw. (rechts) partielle Emanzipation (Abele et al. 2008, S. 585-586)

Die Emanzipationsstrategie entkoppelt die Produktionsausstoßmenge von der Nachfragemenge, die Kapazitäten werden bewusst nicht an die Nachfrage angepasst. Die Vorteile beinhalten eine optimale Auslastung sowie nicht vorhandene bzw. geringe Kosten für eine Kapazitätsanpassung (vgl. Morawetz 2015, S. 19f.). Die Nachteile umfassen einen vergleichsweise, durch die Emanzipation gebildeten, relativ hohen Lagerstand bzw. Kapitalbindungskosten sowie entsprechend lange Reaktionszeiten im Zuge einer Produktumstellung.

Die Synchronisationsstrategie passt, im Gegensatz zur Emanzipation, ihre Produktionsausstoßraten unmittelbar an die marktbedingte Nachfragemengen an. Dementsprechend entfallen Lager- und Kapitalbindungskosten für Fertigprodukte. Nachteile dieser stark kund*innenorientierten Produktion sind hohe Produktionsänderungskosten sowie Kosten für laufende Anpassungen und damit verbunden starke Auslastungsschwankungen der Ressourcen, vor allem bei Vorhandensein von saisonalen bzw. aktionsgetriebenen Produkten. Dementsprechend ist diese Strategie vorwiegend bei Märkten mit geringen Nachfrageschwankungen oder teilweise hochpreisigen, kund*innenindividuellen Gütern (z.B. die Anfertigung von Ringen und Juwelen) oder auch (rasch) verderblichen Waren (Lebensmitteln) bzw. modischen Artikeln anwendbar (vgl. Morawetz 2015, S. 19f.).

Die partielle Emanzipation kombiniert Elemente aus der Emanzipation als auch Synchronisation und passt die Produktionsausstoßmenge stufenweise der Nachfrage an. Diese Anpassung kann entweder in zeitlichen Intervallen stattfinden, oder auch durch zuvor definierte Abweichungen der Nachfragemenge von der Produktionsmenge ausgelöst werden und entspricht damit den Anforderungen des Lean Managements. Ein großer Nachteil dieser Strategie liegt - neben den damit verbundenen höheren Lagerständen als bei der Synchronisation - darin, dass Kapazitätseinheiten, beispielsweise Maschinen oder Arbeitskräfte, in der Regel nicht stufenlos skalierbar bzw. zuschaltbar oder deaktivierbar sind. Auf Grund der betrieblichen Realität stellt die Synchronisation eine partielle Emanzipation mit besonders kleinen Abstufungen in den Kapazitätseinheiten dar (vgl. Morawetz 2015, S. 21).

2.2.4 Programmorientierte Bedarfsermittlung

Die programmorientierte Bedarfsermittlung stellt ein Teilproblem der Losgrößenplanung im Zuge der Produktionsplanung dar. Dabei werden die folgenden Informationen benötigt: Das geplante Hauptproduktionsprogramm, der Erzeugniszusammenhang, die geplanten Durchlaufzeiten der Produkte sowie die Entwicklung der Lagerbestände. Das Hauptproduktionsprogramm bestimmt Art und Menge der herzustellenden Endprodukte in den einzelnen Perioden. Der Erzeugniszusammenhang definiert den bereitzustellenden Materialbedarf in einem mehrstufigen Produktionsprozess für die Fertigung des jeweiligen Endprodukts und wird v.a. über Materialstücklisten aufgelöst. Die Durchlaufzeit informiert über die Dauer der Fertigstellung eines Endprodukts in Bezug auf den ausgelösten Produktionszeitpunkt. Der verfügbare Lagerbestand kann häufig dazu verwendet werden, einen Teil der Bedarfsmengen zu decken, und muss deshalb laufend überwacht werden (vgl. Tempelmeier 2005, S. 179ff.).

Das Hauptproduktionsprogramm stellt den Primärbedarf dar. Auf Basis der Produktionsmengen der Endprodukte ergibt sich unter Beachtung der Erzeugnisstruktur der *Sekundärbedarf* aller untergeordneten (Zwischen-) Erzeugnisse. Auf die untergeordneten Fertigungsartikel wird ggf. der dazugehörige Primärbedarf aufgeschlagen, um den *Bruttobedarf* eines Produkts in einer bestimmten Periode zu erhalten. Anschließend wird vom Bruttobedarf der verfügbare Lagerbestand abgezogen, um den *Nettobedarf* des Erzeugnisses zu erhalten. Anschließend wird für die dazugehörige Produktion die sogenannte *Vorlaufzeit* abgezogen, wodurch die periodenspezifischen Beschaffungs- und Produktionsmengen entstehen (vgl. Tempelmeier 2005, S. 187ff.).

2.2.5 Losgrößenplanung und Produktionssystemoptimierung

Auf Basis der periodenspezifischen Nettobedarfsmengen muss die Häufigkeit der Produktion und die dazugehörige Losgröße ermittelt werden. Die Zusammenfassung vieler periodenspezifischer Nettobedarfe zu einer Produktion verursacht entsprechend hohe Lagerkosten, während häufige Produktionen entsprechend hohe Rüstkosten nach sich ziehen. Die Losgrößenplanung beschäftigt sich, unter Berücksichtigung der zu Grunde liegenden Stücklistenauflösung, mit dem Problem der Lager- und Rüstkosten im Zuge der Produktionsplanung und versucht dabei, häufig unter weiteren Nebenbedingungen wie z.B. beschränkt verfügbaren Kapazitäten, ein wirtschaftliches Optimum zu finden (vgl. Tempelmeier 2005, S. 192).

Die Bestimmung der möglichst optimalen Konfiguration eines Produktionssystems verlangt von der Planung, für jede gegebene Konfigurationsalternative die relevanten Kenngrößen vorab zu bestimmen, darauf basierend zu bewerten und anschließend die für die jeweilige Situation angepasste beste Lösung auszuwählen. Die Konfigurationsalternative stellt die Kombination von Entscheidungsvariablen in Bezug auf das vorliegende Problem dar. Die relevanten Kenngrößen stellen i.d.R. ökonomische Ausprägungen der Zielgrößen Liefertreue, Zielbestand, Produktionsdurchsatz unter Berücksichtigung sämtlicher spezifischer Produktionsbedingungen (Maschinen- und Anlagenverfügbarkeit, Schichtpläne, etc.) zum Auffinden einer möglichst kostengünstigen Ressourcenkombination dar. Karmarkar hat beispielsweise, unter Zuhilfenahme der Warteschlangentheorie, analytische Aussagen über den Zusammenhang zwischen Losgröße und Durchlaufzeit getroffen bzw. abgeleitet (vgl. Arnold et al. 2008, S. 113).

Das bekannteste statische Verfahren in der Materialdisposition stellt beispielsweise die *Andler'sche Losgrößenformel* dar. Die Formel kalkuliert die Bestellmenge gleichzeitig minimierend die Summe aus Beschaffungs- und Lagerkosten. Dabei werden jedoch bestimmte Systemvereinfachungen vorausgesetzt (konstante Bedarfsverläufe, stetige Lagerabgänge im Zeitverlauf, etc.), welche in der Realität nicht immer zutreffen. Dadurch werden in der Literatur zunehmend auch dynamische Verfahren betrachtet, welche z.B. Bedarfsschwankungen erlauben (vgl. Arnold et al. 2008, S. 443).

Die produktionslogistische Bewertung eines Produktionssystems erfolgt z.B. über das *Trichtermodell*. Dieses beschreibt jede beliebige Produktionskapazitätseinheit über die Einflussgrößen Zugang, Bestand und Abgang in ihrem Durchlaufverhalten. Die *Trichterformel* drückt die Relation aus mittlerem Bestand und mittlerer Leistung als die mittlere Bestandsreichweite (in Stunden) aus (vgl. Sven Grundmann 2007, S. 12f.).

2.2.6 Produktionsglättung

Die Produktionsglättung stellt die Minimierung der Produktionsauslastungsschwankungen in den Mittelpunkt der Betrachtung. Die inhaltliche Basis für die *Heijunka* genannte Nivellierung von Produktionsaufträgen liefert das Toyota Produktionssystem (TPS). Im Zuge dieser Planung werden maximale Losgrößen, die Produktionsreihenfolge und Produktionstermine, die dazugehörigen Produktionsmengen und Bestände gemeinsam geplant, anstelle einer Zerlegung des Produktionsplanungsproblems in mehrere isoliert zu betrachtende Teilprobleme. Letzteren Ansatz verfolgt die MRP-Planung. Dementsprechend werden die zusammenhängenden Planungsaufgaben in Heijunka transparenter als durch eine hierarchische, partielle Optimierung der Produktionsplanung. Dieser holistische Ansatz ist von zentraler Bedeutung für die Zielstellung dieser Arbeit und spiegelt sich in der Definition des *Fitnesswerts* wider.

Dieser Skalar stellt ein Maß für die Güte einer in einem dazugehörigen Zielfunktionssystem betrachteten Lösung dar. Die *Heijunka-Nivellierung* unterscheidet in der Abfolge grundsätzlich zwei Planungsphasen (vgl. Veit 2010, S. 20f.):

1. Auslegung des Produktionssystems in regelmäßigen, mittelfristigen Intervallen auf Grund des aktuellen Systemzustands und Erfahrungen auf Basis des letzten Auslegungszyklus sowie Prognosen betreffend das zukünftige Intervall.
2. Tägliche Planung im operativen Betrieb sowie dazugehöriges Produktions-Controlling (Soll-Ist-Abweichungen des Produktionsausstoßes).

Die Ergebnisse beider Planungsphasen spiegeln sich einem Heijunka-Board genannten Steckbrett wider, wobei die Planung immer am Engpass ausgerichtet wird. Die Durchlaufzeit nach der beplanten, jeweils für den Plan gültigen, Stufe muss stets eindeutig bestimmbar und kürzer als die geforderte Lieferzeit sein (vgl. Veit 2010, S. 21).

Auf Basis der Identifikation dieses Schrittmacherprozesses sollte der erste Schritt eines schlanken Produktionssystems die Berechnung des Kundentakts \hat{T}_i darstellen, welcher sich auf Basis der verfügbaren Produktionszeit dividiert durch die Nachfrage nach dem jeweiligen Produkt i berechnet. Auf Grund von Leistungsreserven bzw. der Fertigung von Produktionslosen muss die tatsächliche Zykluszeit T_i kleiner sein als \hat{T}_i . Das Ziel von just-in-time Konzepten besteht in einer Produktion möglichst nahe am Kundentakt. Für die Bildung von regelmäßigen Intervallen bietet sich die EPEI-Planung an. Diese definiert den Abstand zweier Produktionslosauflagezeitpunkte in einem Zyklus pro Produkt sowie die maximale Produktionsmenge. Durch die EPEI Berechnung wird also transparent, welche Produkte in welchem Abstand und in welcher Menge unter Berücksichtigung der verfügbaren Produktionszeit gefertigt werden (vgl. Veit 2010, S. 21f.).

Eine effiziente Methodik im Zuge der Anwendung und Ausführung einer Produktionsglättung stellt dabei allgemein das Prinzip der Vorverschiebung von Produktionsaufträgen dar. Der Grad an vorverschiebbaren Aufträgen erhöht den Zielbestand (sowie das Bestandsrisiko) und muss daher auf diese Art und Weise entsprechend in der Zielfunktion bewertet werden (vgl. Kogan und Tell 2009, S. 223ff.).

Für diese Arbeit stellt sich also innerhalb der Methodik auch die Frage nach dem Zyklus pro Produkt sowie der maximal zur Verfügung stehenden Produktionsmenge. Dabei spielen auch Planungsrestriktionen sowie die Auflösung bzw. Granularität der Planung eine entscheidende Rolle. Weiters spielt die Möglichkeit bzw. die Höhe des Anteil an vorverschiebbaren Aufträgen eine entscheidende Rolle. Im operativen Betrieb stellt die Reihenfolgebildung im Zuge der Feinplanung ein zentrales Element dar (vgl. Veit 2010, S. 21f.). Dementsprechend wird im nächsten Abschnitt ein grundlegendes heuristisches Konzept basierend auf Toyota diskutiert, bevor diese Arbeit eingegrenzt wird.

Erstellung geglätteter Produktionsreihenfolgen: Tegel stellt in seiner Dissertation (vgl. Tegel 2012, S. 47ff.) für das sogenannte *Product and Output Rate Variation Problem* (PORV) ein mathematisches Modell zur Bildung geglätteter Produktionsreihenfolgen von Vor- und Endprodukten am Beispiele des Automotive-Sektors auf, welches darauf abzielt, die absolute Abweichung der bis zu einem definierten Zeitpunkt hergestellten Menge eines Endprodukts gegenüber der Sollmenge möglichst gering zu halten. Dabei bedient er sich des Konzepts der *Losteilung*. Die Lösung dieses PORV mittels eines Standardsolvers funktioniert für kleine Instanzen gut, allerdings steigt die Laufzeit auf Grund der exponentiell steigenden Komplexität, die sich aus der Vielzahl an kombinatorisch möglichen Produktionsplänen ergibt, bei höheren Endproduktbedarfen oder einer höheren Anzahl an Endprodukten entsprechend stark an (vgl. Tegel 2012, S. 55). Die Linearisierung der Betragsfunktion des zuvor nichtlinearen Problems ändert nichts an dem Laufzeitverhalten. Neben der exakten Lösung des Problems stellt er auch die vorproduktorientierte *Goal-Chasing-Heuristik* (siehe Anhang, Abschnitt A.5) von Toyota in seiner Arbeit vor. Diese beruht darauf, dass ein Endprodukt i an die t -te Stelle des Plans aufgenommen wird, wenn der Bedarf von i noch nicht vollständig gedeckt worden ist und durch die Hinzunahme von i eine möglichst geringe Abweichung der Vorprodukt-Sollbedarfsmenge von deren kumuliertem Bedarf erreicht wird. Das grundsätzliche Ziel dieser vorproduktorientierten Heuristik besteht in der Findung

einer Endproduktreihenfolge, welche die Schwankungen der Vorproduktbedarfe reduziert (vgl. Tegel 2012, S. 55-60).

Auf Basis der Anwendung dieser Methodik resümiert Tegel, dass insbesondere bei Vorprodukten, die lediglich für ein einziges Endprodukt benötigt werden, eine erhebliche Reduktion der Nachfrageschwankung umgesetzt werden kann, während Vorprodukte, welche in jedes Endprodukt in ähnlicher Menge einfließen, sich ein gleichmäßiger Produktionsplan weniger stark (positiv) auswirkt. Außerdem reduziert sich die Qualität der Heuristik bzw. die Lösungsgüte bei zunehmender Anzahl an Endprodukten (vgl. Tegel 2012, S. 90).

Produktionsglättung im Kontext dieser Arbeit: Das vorliegende MOMPMP Problem erfordert keine Erstellung einer geglätteten Produktionsreihenfolge im Zuge der Wochenplanung auf der Planungsebene eines Produktionsloses, sondern eine möglichst gleichmäßige Produktionsausstoßmenge über einen mittel- bis längerfristigen Zeitraum. Die Produktionsausstoßmenge stellt dabei die Menge pro Produktionsartikel dar, welche das Werk pro Periode, über alle Artikel aufsummiert, als Output (Summe aller Endprodukte pro Periode) generiert. Die Aufteilung der Wochenmenge pro Produkt ist Teil der operativen Feinplanung und hat dementsprechend wenig Gemeinsamkeiten mit der Erstellung einer mittelfristig optimierten wöchentlichen Produktionsplanung, in welcher die konkreten Anliefertage- und Mengen in der Woche nicht im Voraus exakt, d.h. tagesgenau, bekannt sind bzw. auf Grund diverser Abhängigkeiten (Lieferfähigkeit der Lieferant*innen, etc.) nicht endgültig für jede Woche bereits im Voraus feststehen.

Prinzipiell wird an dieser Stelle unterschieden zwischen einer Produktion mit Montageprozessen, welche unterschiedliche Vor- und Endprodukte erfordert, und einer Produktion, welche die Produkte in einem mehrstufigen Produktionsprozess weiterverarbeitet und nicht ineinander eingehen lässt. Die letztgenannte Variante stellt den industriellen Anwendungsfall im Rahmen dieser Arbeit dar. Das Konzept der Losgrößenplanung, siehe 2.2.5, ist methodisch für diese Arbeit relevant, weil für die Summe aller Produkte und deren jeweiligen wöchentlichen Produktionsmengen die maximal verfügbare Produktionsmenge eingehalten werden muss. Dies wird, nach ausführlicher Literaturanalyse, entsprechend in der Methodenentwicklung berücksichtigt.

2.3 Mathematische Grundlagen der Nichtlinearen Multikriteriellen Optimierung

Multikriterielle Optimierung beschäftigt sich mit Problemen, welche mehrere (i.d.R. konkurrierende) Zielsetzungen im Rahmen eines Verfahrens simultan berücksichtigen (vgl. Miettinen Kaisa 2001, S. 1). Diese Probleme verfügen allgemein über die folgende Form:

$$\text{Minimiere } (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \quad (3.1)$$

in Abhängigkeit von $\mathbf{x} \in S$,

wobei $k \geq 2$ für die Anzahl der betrachteten Teilzielfunktionen $f_i : \mathbf{x} \in \mathbb{R}^n \rightarrow \mathbb{R}$ steht, welche gleichzeitig optimiert werden. Der Vektor betreffend die Entscheidungsvariablen $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ gehört zu dem (nichtleeren) zulässigen Suchraum $S \subset \mathbb{R}^n$. Der Bereich zulässiger Lösungen wird wiederum zumeist durch Nebenbedingungen eingeschränkt. Das Ziel besteht darin, optimale Entscheidungsvariablen $\mathbf{x}^* \in X$ zu finden, wobei keiner der Teilzielfunktionswerte weiter verbessert werden kann, ohne dass gleichzeitig (zumindest) ein anderer Teilzielfunktionswert verschlechtert wird. Diese Lösung wird *Pareto-Optimum* genannt (vgl. Miettinen Kaisa 2001, S. 1ff.). Aus mathematischer Perspektive wird ein multikriterielles Problem als gelöst betrachtet, wenn eine Pareto-optimale Lösung bzw. Lösungsmenge gefunden worden ist. Die dazugehörige Optimierung ist auch unter dem Begriff der *Vektor-Optimierung* bekannt. Ein geeignetes Verfahren soll also sowohl schnell zur Pareto-Menge

konvergieren als auch andererseits diese Menge möglichst gleichmäßig approximieren. Aus der Pareto-optimalen Menge wird dann von dem*der Entscheidungsträger*in eine Lösung als beste Lösung ausgewählt. Multikriterielle Optimierungsprobleme werden häufig durch eine Skalarisierung gelöst. Das bedeutet, dass das vorliegende, zu modellierende, Problem in ein singlekriterielles Problem konvertiert wird. Skalarisierung kann z.B. über die folgenden Ansätze erfolgen (vgl. Ehrgott und Gandibleux 2003, S. 77f.):

1. Besetzung der Teilziele mit positiven (unterschiedlichen) Gewichtungen.
2. Mittels der sogenannten ϵ – *constraint* Methode.
3. Über die gewichtete Tchebycheff-Metrik.

Anschließend wird das einkriterielle, lineare bzw. nichtlineare Problem mit einer reellwertigen Zielfunktion mittels kombinatorischer Optimierungsalgorithmen gelöst. Da die Laufzeitkomplexität grundsätzlich von der Effizienz des dazugehörigen Algorithmus zur Problemlösung abhängt, wird zunächst der fundamentale Fachbegriff *Algorithmus* definiert:

Definition 1. Algorithmus

„Ein Algorithmus ist eine vollständige, präzise und in einer Notation oder Sprache mit exakter Definition abgefasste, endliche Beschreibung eines schrittweisen Problemlösungsverfahrens zu Ermitteln gesuchter Datenobjekte aus gegebenen Werten von Datenobjekten, in dem jeder Schritt aus einer Anzahl ausführbarer, eindeutiger Aktionen und einer Angabe über den nächsten Schritt besteht“ (Pomberger und Dobler 2008, S. 33).

Die Fallunterscheidung in Bezug auf die Nichtlinearität wird durch die folgende Definition geklärt:

Definition 2. Multikriterielles (nicht) lineares Problem

„Wenn die gesamte Funktionenmenge, bestehend aus Zielsetzungen und Nebenbedingungen, linear ist, dann wird das Problem als multikriterielles lineares Problem bezeichnet. Wenn zumindest eine der Funktionen nichtlinear ist, dann wird das Problem als nichtlineares multikriterielles Optimierungsproblem bezeichnet. Dementsprechend ist das Problem nicht differenzierbar, d.h. einige der Funktionen sind nicht differenzierbar“ (Miettinen Kaisa 2001, S. 2).

Dementsprechend klärt sich allgemein bei der Formulierung des konkreten Problems sowie des dazugehörigen Modells, ob ein nichtlineares Problem vorliegt und wird bis zu diesem Zeitpunkt mit der entsprechenden Hypothese als nichtlineares Optimierungsproblem bedacht.

2.3.1 Methodische Grundlagen

Eine allgemeine *Nichtlineare Multikriterielle Optimierungsaufgabe* (NLMOA) mit M Zielfunktionen, J Nebenbedingungen und N Entscheidungsvariablen (auch als Optimierungsvariablen bezeichnet), welche im (Entscheidungs-) Variablenvektor $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ zusammengefasst werden, kann wie folgt angegeben werden (Kux 2011, S. 7):

$$NLMOA \left\{ \begin{array}{ll} f_m(\mathbf{x}) \rightarrow \min & m = 1, 2, \dots, M \\ g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^n \end{array} \right\} \quad (3.2)$$

Dabei sind $M, J, N \in \mathbb{N}$, wobei es für das Vorliegen eines nichtlinearen Problems bereits genügt, dass eine Funktion aus der Menge an Zielfunktionen bzw. Nebenbedingungen durch eine nichtlineare Funktion gegeben ist. Die Optimierungsvariablen sind in der Regel durch eine untere Grenze $x_i^{(U)}$ als auch eine obere Grenze $x_i^{(O)}$ beschränkt, andernfalls handelt es sich um eine zumindest teilweise unbeschränkte Optimierungsaufgabe. Prinzipiell genügt es Minimierungsaufgaben mit Ungleichheitsnebenbedingungen zu betrachten, weil die Maximierung von $f(\mathbf{x})$ äquivalent zur Minimierung von $-f(\mathbf{x})$ ist und jede

Gleichung $h(\mathbf{x}) = 0$ durch zwei Ungleichungen $h(\mathbf{x}) \geq 0 \wedge h(\mathbf{x}) \leq 0$ beschrieben werden kann (vgl. Kux 2011, S. 7).

Definition 3. Zulässige Menge

„Die Menge aller $\mathbf{x} \in \mathbb{R}^n$, die in den gegebenen Intervallgrenzen liegt und die Nebenbedingungen erfüllt, wird als zulässige Menge

$$S = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \begin{array}{ll} g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \end{array} \right\} \quad (3.3)$$

bezeichnet.“ (Kux 2011, S. 7-8).

2.3.1.1 Pareto-Optimalität und Dominanz

Auf Grund der (teilweisen) Konkurrenzierbarkeit einzelner Teilziele sowie einer möglichen Unvereinbarkeit der vorliegenden Teilziele ist es in der Regel nicht möglich, eine Lösung zu finden, welche alle Ziele gleichzeitig vollumfänglich verbessert. Daher werden die einzelnen Entscheidungsvariablen oder auch Entscheidungsvektoren als optimal betrachtet, wenn man sie nicht mehr weiter verbessern kann, ohne zumindest eine andere Variable dabei zu verschlechtern.

Definition 4. Pareto-Optimalität

Eine Lösung $\mathbf{x}^* \in S$ wird als **Pareto-optimal** bezeichnet, wenn es kein anderes $\mathbf{x} \in S$ gibt, so dass $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ für alle $i = (1, \dots, k)$ und $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ für zumindest einen Index j gilt. Der Zielfunktionsvektor ist Pareto-optimal, wenn der korrespondierende Entscheidungsvariablen-Vektor Pareto-optimal ist (vgl. Zitzler et al. 2001, S.2; Chiandussi et al. 2012, S. 915).

Der Ausdruck **Pareto-Optimalität (PO)** wird i.d.R. (sofern nicht anders angegeben) in Bezug auf den gesamten Entscheidungsvariablenraum verstanden. Diese Definition besagt, dass \mathbf{x}^* Pareto-optimal ist, wenn es keinen Vektor \mathbf{x} gibt, der ein bestimmtes Kriterium verbessert, ohne dabei gleichzeitig mindestens ein anderes Kriterium zu verschlechtern (vgl. Chiandussi et al. 2012, S. 915).

Definition 5. Pareto-Dominanz

Ein Vektor $\mathbf{u} = (u_1, \dots, u_k) \in \mathbb{R}^n$ dominiert einen zweiten Vektor $\mathbf{v} = (v_1, \dots, v_k) \in \mathbb{R}^n$ (dargestellt durch $u \prec v$) nur dann, wenn \mathbf{u} zumindest in einer Dimension besser ist als \mathbf{v} , ohne in den übrigen Dimensionen schlechter zu sein (vgl. Chiandussi et al. 2012, S. 915; Vaillant 2015, S. 65):

$$u_k \leq v_k \quad \forall k \in [1, n]$$

$$\exists l : u_l < v_l$$

Diese **Dominanz-Relation** besagt, dass \mathbf{u} in allen Dimensionen nicht schlechter ist als \mathbf{v} und in mindestens einer Dimension \mathbf{v} übertrifft (vgl. Vaillant 2015, S. 65). Dementsprechend lässt sich die vorherige Definition auch über die Pareto-Dominanz erklären:

Definition 6. Pareto-Optimalität

„Der Vektor $\mathbf{x}_i = (x_1, \dots, x_m) \in \mathbb{R}^m$ ist pareto-optimal, wenn keine andere Lösung $\mathbf{x}_j = (x_1, \dots, x_m) \in S$ existiert, deren Zielfunktionsvektor $f(\mathbf{x}_j)$ den Vektor $f(\mathbf{x}_i)$ dominiert“ (Vaillant 2015, S. 66).

$$\nexists \mathbf{x}_j : \left\{ \begin{array}{ll} f_k(\mathbf{x}_j) \leq f_k(\mathbf{x}_i) & \forall k \in [1, n] \\ \exists l : f_l(\mathbf{x}_j) < f_l(\mathbf{x}_i) \end{array} \right. \quad (3.4)$$

Pareto-optimale Lösungen werden auch als *effiziente* bzw. *nicht unterlegene* oder *nicht-dominierte* Lösungen bezeichnet. Man spricht dabei von *Nicht-Dominanz* (vgl. Chiandussi et al. 2012, S. 915).

Definition 7. Pareto-Front

Für ein gegebenes Mehrzielproblem, $F(\mathbf{x})$, und einer dazugehörigen Menge (*Set*) an Pareto-optimalen Lösungen P^* wird die Pareto-Front PF^* wie folgt definiert (vgl. Chiandussi et al. 2012, S. 915):

$$PF^* := \{\mathbf{u} = F(\mathbf{x}) \mid \mathbf{x} \in P^*\}$$

Nicht dominierte Vektoren, die im Zielraum aufgetragen werden, werden in ihrer Gesamtheit als Pareto-Front bzw. Pareto-Menge bezeichnet (vgl. Chiandussi et al. 2012, S. 915). Das Ziel eines Verfahrens zur Lösung einer multikriteriellen Optimierungsaufgabe stellt idealerweise die Bestimmung (eines Ausschnitts) der Pareto-Menge dar. Lösungen entlang der Pareto-Front werden für gewöhnlich akzeptiert (vgl. Chiandussi et al. 2012, S. 913). Aus dieser Gesamtheit wird dann eine beste Lösung ermittelt (vgl. Kux 2011, S. 8).

Der Übergang von einer Pareto-optimalen Lösung auf eine andere wird durch einen *Trade-Off* bewerkstelligt. Dabei ändern sich einzelne Teilzielwertwerte bzw. Vektoren entlang der Pareto-Front, ohne dass sich der Wert der gesamten Zielfunktion ändert. Dieser Prozess wird auch als Grenzrate der Substitution bezeichnet (vgl. Zitzler et al. 2001, S. 4).

Neben der Definition der Pareto-Optimalität gibt es ergänzend eine Definitionen der *schwachen Pareto-Optimalität*, welche darauf beruht, dass es keinen Lösungsvektor gibt, welcher die bisherige gefundene Lösung in allen Dimensionen übertrifft. Das bedeutet, dass die Lösung zumindest einer vektoriiellen Dimension gleich bzw. konstant bleibt (vgl. Ehrgott und Gandibleux 2003, S. 229). Schwach Pareto-optimale Lösungen sind im Rahmen der Optimierung relevant, weil diese in der Regel einfacher zu generieren sind als Pareto-optimale Lösungen.

Definition 8. Schwache Pareto-Optimalität

Ein Punkt $\mathbf{x}^* \in S$ wird als **schwach Pareto-optimal** bezeichnet, wenn es kein anderes $\mathbf{x} \in S$ gibt, sodass $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ für alle $i = (1, \dots, k)$ gilt (vgl. Chiandussi et al. 2012, S. 915).

Definition 9. Strenge Pareto-Optimalität

Ein Punkt $\mathbf{x}^* \in S$ wird als **strenge Pareto-optimal** bezeichnet, wenn es kein anderes $\mathbf{x} \in S$ gibt, sodass $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ für alle $i = (1, \dots, k)$ gilt (vgl. Chiandussi et al. 2012, S. 916).

Der Vergleich von Definition 6 und 8 ergibt, dass Lösungen schwach Pareto-optimale Punkte darstellen, wenn es keine andere Lösung gibt, die sie in sämtlichen Dimensionen übertrifft. Für strenge Pareto-Optimalität, siehe auch Definition 9, wird hingegen gefordert, dass keine andere Lösung in allen Dimensionen gleichwertig oder besser ist. Im Vergleich mit Definition 4 wird deutlich, dass Pareto-Optimalität eine Überlegenheit einer Lösung in zumindest einer Vektordimension ausdrückt.

Der über Transformation durch den Funktionszusammenhang der zulässigen Menge S (aus dem Designraum) aufgespannte Zielraum $Z = f(S)$ beinhaltet die Menge aller zulässigen Lösungen, siehe Abb. 2.2. Die Abbildung der Pareto-Menge im Zielraum ist stets Teil der *Zielraumgrenze*. Während die Transformation der Pareto-Menge aus dem Designraum im zweidimensionalen Fall (wie in Abb. 2.2) eine Pareto-Front im Bild- und Zielraum abbildet, entsteht dadurch bei höheren Dimensionen eine *Hyperfläche* (vgl. Vaillant 2015, S. 66f.; Branke et al. 2008, S. X).

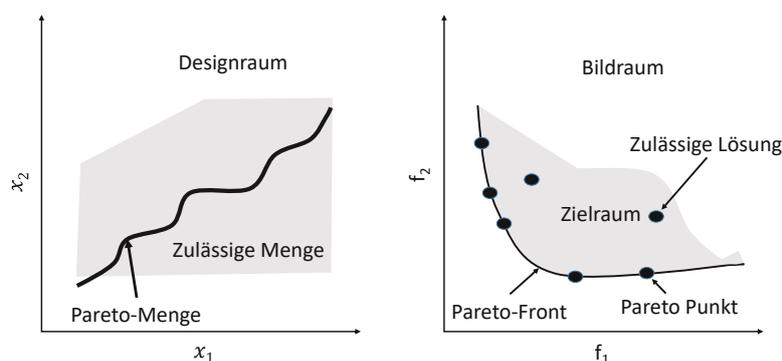


Abb. 2.2: Design- und Bildraum (Quelle: eigene Darstellung, in Anlehnung an Vaillant 2015, S. 67)

Möglicherweise gibt es, je nach Lösungsraum, eine unendlich große Anzahl an Pareto-optimalen Lösungen, die allesamt als gleichwertig zu bewerten sind. Im Einklang mit der Definition in Bezug auf das *globale Minimum* eines multikriteriellen Optimierungsproblems von Chiandussi entsprechen die Lösungen entlang der Pareto-Front den besten verfügbaren Lösungen aus Z und erfüllen damit die Bedingung für diese Definition (vgl. Chiandussi et al. 2012, S. 916).

2.3.1.2 Konvexität

Ein Optimierungsproblem wird als *konvex* bezeichnet, wenn sowohl Pareto-Menge als auch Pareto-Front konvex sind (vgl. Vaillant 2015, S. 67). Die Konvexität wird anhand der Begriffe konvexe Mengen und konvexe Optimierung im nächsten Paragraphen definiert.

Stein definiert den Begriff einer **konvexen Menge** folgendermaßen (vgl. Stein 2018, S. 38):

Eine Menge $X \subset \mathbb{R}^n$ wird als *konvex* bezeichnet, wenn folgende Beziehung gilt: $\forall \mathbf{x}, \mathbf{y} \in X, \lambda \in (0, 1) : (1 - \lambda)\mathbf{x} + \lambda\mathbf{y} \in X$. Die Verbindungsstrecke von je zwei beliebigen Punkten in X gehört also komplett zu X , siehe Abb. 2.3.

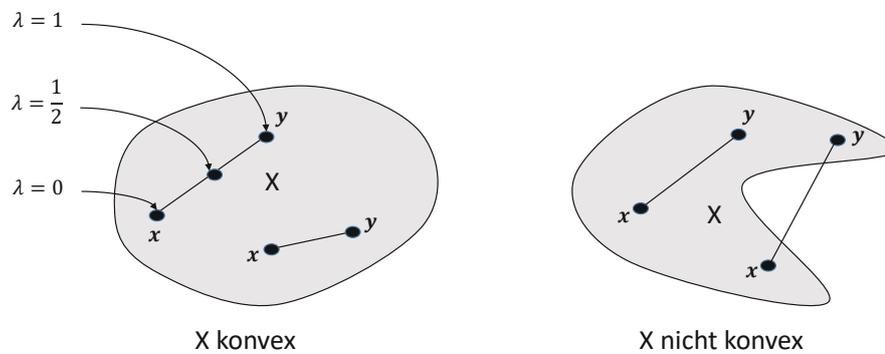


Abb. 2.3: Konvexität von Mengen im \mathbb{R}^2 (Quelle: eigene Darstellung, angelehnt an Stein 2018, S. 39)

Für eine konvexe Menge $X \subset \mathbb{R}^n$ heißt eine Funktion $f : X \rightarrow \mathbb{R}$ konvex (auf X), wenn gilt (vgl. Stein 2018, S. 38):

$$\forall \mathbf{x}, \mathbf{y} \in X, \lambda \in (0, 1) : f(1 - \lambda)\mathbf{x} + \lambda\mathbf{y} \leq (1 - \lambda)f(\mathbf{x}) + \lambda f(\mathbf{y}) \quad (3.5)$$

Das bedeutet, der Funktionsgraph von f verläuft unter jeder seiner Sekanten, siehe auch Abb. 2.4 zur Verdeutlichung. Das Konzept der konvexen Optimierung wird im Anhang (A.9) diskutiert.

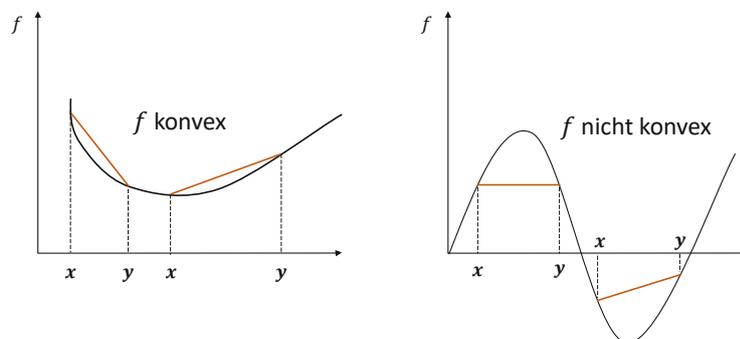


Abb. 2.4: Konvexität von Funktionen (Quelle: eigene Darstellung, in Anlehnung an Stein 2018, S. 39)

2.3.1.3 Komplexitätstheorie

Die Komplexitätstheorie bewertet Probleme anhand der Anzahl erforderlicher Rechenschritte für deren Lösung. Die meisten Probleme, welche in der kombinatorischen Optimierung auftreten bzw. mittels nichtlinearer Optimierungsverfahren gelöst werden, gehören zur Klasse **NP**. Deshalb empfiehlt es sich, die Begriffe **P** und **NP** im Zusammenhang mit der Turingmaschine vorab der eigentlichen Methodenabschnitte zu definieren (vgl. Korte und Vygen 2012, S. 403ff.; Prokop 2014, S. 65ff.).

Definition 10. Entscheidungsproblem: *Ein Entscheidungsproblem stellt ein Paar $P = (X, Y)$ dar, wobei X in polynomieller Zeit bestimmbar und $Y \subseteq X$ ist. Die Elemente von X heißen Instanzen von P . Die Klasse aller Entscheidungsprobleme, für die ein polynomieller Algorithmus existiert, wird mit **P** bezeichnet (vgl. Korte und Vygen 2012, S. 412f.).*

Definition 11. NP: *„Ein Entscheidungsproblem ist genau dann in **NP**, wenn es einen polynomiellen nichtdeterministischen Algorithmus hat. Ein Entscheidungsproblem $P \in NP$ heißt **NP-vollständig**, falls sich alle anderen Probleme in NP polynomiell in P transformieren“ (Korte und Vygen 2012, S. 415f.).*

Die Klasse NP beinhaltet dementsprechend alle Probleme, für die eine Lösung in polynomieller Zeit nachgewiesen werden kann (vgl. Prokop 2014, S. 66).

Polynomielle Transformation bedeutet in Bezug auf zwei Entscheidungsprobleme $P_1 = (X_1, Y_1)$ und $P_2 = (X_2, Y_2)$, dass es eine in polynomieller Zeit berechenbare Funktion $f : X_1 \rightarrow X_2$ gibt, sodass $f(x_1) \in Y_2$ für alle $x_1 \in Y_1$ und $f(x_1) \in X_2 \setminus Y_2$ für alle $x_1 \in X_1 \setminus Y_1$ gilt (vgl. Korte und Vygen 2012, S. 416).

Dementsprechend stellen die NP -vollständigen Probleme jedenfalls die schwersten Probleme aus der Klasse NP dar und können nur dann in polynomieller Zeit gelöst werden können, wenn $P = NP$ gilt. Der *Satz von Cook* beweist die Existenz NP -vollständiger Probleme. Dabei wurde der Beweis erbracht, dass ein bestimmtes Entscheidungsproblem, **SATISFIABILITY (SAT)**, NP -vollständig ist. **SAT** stellt die Frage, ob eine gegebene boolesche Formel erfüllbar ist. Die Menge all jener Probleme, die sich (polynomiell) auf **SAT** reduzieren lässt, ist **NP-hart** (vgl. Korte und Vygen 2012, S. 403-417; Prokop 2014, S. 66-68).

Die Turingmaschine TM basiert wiederum auf dem Forscher Alan Turing, der sich fragte, ob eine automatisierte Beweisführung von Formeln aus der Mathematik möglich ist und erstellte dafür ein maschinelles Konzept. Die Turing-Maschine ist formal betrachtet ein 7-Tupel:

$$TM = (Q, \Gamma, b, \Sigma, \delta, q_0, F) \quad (3.6),$$

wobei Q eine endliche, nicht-leere Menge an Zuständen definiert. Γ bezeichnet eine endliche, nicht-leere Menge an auf einem Band verwendbarer Zeichen. b beschreibt den Initialzustand unbeschriebener Stellen. Σ stellt die auf dem Band vorhandene endliche Menge an Zeichen dar, δ bezeichnet die Übergangsfunktion, q_0 den Anfangszustand und F die Menge an Endzuständen der Turingmaschine (vgl. Prokop 2014, S. 29f.; Korte und Vygen 2012, S. 403-406).

Algorithmen stellen also schrittweise Anleitungen zur Problemlösung dar, welche eine Sequenz von Instruktionen bilden und dabei systematisch vorzunehmende Datenmodifikationen erlauben. Sie sind beliebig aufteilbar, sodass Programme aus einer Reihe derartiger Handlungsanweisungen zur Lösung diverser Teilprobleme bestehen. Häufig wird in den folgenden Abschnitten sogenannter *Pseudocode* verwendet, um den algorithmischen Ansatz, bestehend aus allgemein üblichen Sprachkonstrukten (z.B. if-then-else-end if, for/while Schleifen, etc.), verständlich zu skizzieren (vgl. Prokop 2014, S. 61f.).

2.3.1.4 Landau-Notation

Die *Landau-Notation* bzw. *O-Notation* betrachtet das einer Funktion zugrunde liegende asymptotische Verhalten und gibt ein relatives Maß an, um Algorithmen untereinander in Bezug auf ihre Laufzeitkomplexität in Bezug auf eine obere Schranke (\mathcal{O}) grundsätzlich vergleichen zu können. Während beispielsweise $\mathcal{O}(1)$ für eine Klasse mit konstanter Laufzeit steht, beschreibt $\mathcal{O}(n)$ linear wachsende Funktionen bzw. $\mathcal{O}(n^2)$ quadratisch wachsende Funktionen. Diese Notation ist wichtig bei der grundlegenden Bewertung eines Algorithmus. Nach Möglichkeit sollte stets ein relativ, d.h. in Bezug auf andere verfügbare Algorithmen, laufzeiteffizientes Verfahren zur algorithmischen Lösung eines Problems angewendet werden (vgl. Prokop 2014, S. 49).

Definition 12. Ein Algorithmus mit rationalem Input hat eine **polynomielle Laufzeit** oder ist **polynomiell**, falls es eine ganze Zahl k gibt, so dass der Algorithmus einerseits für einen aus n Zahlen bestehenden Input eine $\mathcal{O}(n^k)$ Laufzeit aufweist und andererseits alle in den Zwischenrechnungen auftretenden Zahlen mit $\mathcal{O}(n^k)$ Bits gespeichert werden können (vgl. Korte und Vygen 2012, S. 6).

Für den Fall, dass $k = 1$ gilt, wird das Verfahren als **Linearzeit-Algorithmus** bezeichnet (vgl. Korte und Vygen 2012, S. 6f.).

Tabelle 2.1 vergleicht hypothetische Laufzeiten von Algorithmen mit unterschiedlicher Zeitkomplexität und verdeutlicht die daraus resultierenden differenzierenden Rechenzeiten. Während die Rechenzeit für ein Problem der Komplexität $\mathcal{O}(10n^2)$ noch quadratisch anwächst und auch für große n in akzeptabler Zeit gelöst wird, erfordern Probleme mit *superlinear* anwachsender Problemkomplexität, z.B. $\mathcal{O}(n^{\log n})$, bereits deutlich längere Laufzeiten, während mit der Fakultät¹ wachsende (z.B. kombinatorische) Probleme mit sogenannter *faktorieller Komplexität* $\mathcal{O}(n!)$ bereits mit einem relativ geringen Wert für n praktisch unlösbar sind (vgl. Korte und Vygen 2012, S. 7).

Tab. 2.1: Gegenüberstellung unterschiedlicher Laufzeitkomplexitäten (Korte und Vygen 2012, S. 7)

n	$100n \log n$	$10n^2$	$n^{3,5}$	$n^{\log n}$	2^n	$n!$
10	3 μs	1 μs	3 μs	2 μs	1 μs	4 ms
20	9 μs	4 μs	36 μs	420 μs	1 ms	76 Jahre
50	28 μs	25 μs	884 μs	4 s	13 Tage	10^{48} Jahre
100	66 μs	100 μs	10 ms	5 Stunden	$4 \cdot 10^{13}$ Jahre	
10^4	13 ms	1 s	28 Stunden			
10^6	2 s	3 Stunden	3169 Jahre			
10^{12}	46 Tage	$3 \cdot 10^8$ Jahre				

Da sich diese Arbeit mit der Entwicklung einer laufzeiteffizienten Methodik beschäftigt, ist es sowohl für die Bewertung als auch die Zukunftsfähigkeit der Methodik zur Lösung realer Probleme von besonderer Bedeutung, dass die Verfahrenslaufzeit mit zunehmender Problemgröße idealerweise nur (näherungsweise) linear oder maximal mit logarithmischer Komplexität $\mathcal{O}(\log n)$ wächst, siehe Tab. 2.1. Die Methodik soll einerseits generalisierbar als auch skalierbar sein. Dementsprechend ist die Laufzeit-Effizienz von entsprechender Bedeutung, wenn die Methodik z.B. auf ein größeres Werk oder einem aus Sicht der Anzahl an Planungsperioden längeren Planungshorizont übertragen wird. Für die Abschätzung der Komplexität wird zumindest der Teil des Algorithmus mit der höchsten *O-Klasse* oder das gesamte Verfahren untersucht. In Bezug auf die Gesamtlaufzeit eines Verfahrens bietet es sich generell an, relativ einfache Datenstrukturen in einer Programmiersprache zu verwenden anstelle von überladenen Objekten, welche einen Bruchteil ihrer Eigenschaften zur Laufzeit verwenden. Weiters soll die Laufzeit-Komplexität des Verfahrens gemäß Landau-Notation möglichst gering gehalten werden oder das Verfahren nach Möglichkeit parallelisiert werden, um eine entsprechende Skalierbarkeit zu gewährleisten.

¹ n – Fakultät = $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ Lösungen

2.3.2 Herausforderungen bei der Modellierung und Lösungssuche

In der Produktionsplanung spielt der Einsatz von effektiv-effizienten mathematischen Optimierungsverfahren insofern eine besondere Rolle, weil die Planung auf Basis von aktualisierten Daten in kurzer Zeit mit einem möglichst guten Ergebnis erfolgen soll. Ggf. können im Zuge der Planung, unterstützt durch Methoden der Parallelisierung, mehrere (Absatz-) Szenarien simuliert werden, dennoch möchte die Planungsabteilung i.d.R. nicht lange auf ein Ergebnis warten, sobald der aktualisierte Datensatz zur Verfügung gestellt worden ist. Das spricht grundsätzlich gegen den Einsatz der Kombination aus (einer komplexen, detaillierten oder auch mit kontinuierlichen und diskreten Anteilen versehenen) Simulation und metaheuristischer Optimierung (vgl. Sobottka et al. 2018, S. 413-420). Die Herausforderungen liegen daher vor allem in der Modellierung des vorliegenden Problems sowie der dazugehörigen Auswahl der notwendigen Parameter und Parametrisierung des verwendeten Verfahrens. Wenn über die zu optimierende Zielfunktion wenige bis keine Informationen vorliegen, werden diese als *Blackbox-Probleme* bezeichnet (vgl. Kux 2011, S. 10ff.).

Modellierung des Problems und der Zielfunktion

Im Rahmen der Modellierung wurde bereits in Kapitel 1 der Operations Research Modeling Approach angesprochen. Dabei kann sich der Einsatz bzw. die Kombination von unterschiedlichen Methoden (integriert oder sequenziell) anbieten, welche ausführlich in Abschnitt 3.4 (Hybride Optimierungsverfahren) beschrieben und diskutiert werden.

Abgesehen davon sei an dieser Stelle erwähnt, dass die Herausforderung in der Wahl des richtigen *Detaillierungsgrades* für das vorliegende Problem liegt. Detaillierte Modelle enthalten viele Komponenten, wodurch ein relativ großer Aufwand für die Erstellung einerseits und in Hinblick auf die laufende Modellpflege andererseits notwendig wird, v.a. im Zuge der Datenbeschaffung in der richtigen Qualität und Menge für die Parametrisierung der einzelnen Modellkomponenten. Einfache Modelle sind häufig zu grob, um die Problemstellung zufriedenstellend zu lösen. Ein weiteres Problem entsteht im Zusammenhang mit der Verwendung von stochastischen Modellkomponenten. Diese werden v.a. zur Vermeidung zu großer Modellkomplexität verwendet, um komplizierte Teilmodelle durch (relativ einfach zu modellierende) stochastische Ersatzmodelle zu ersetzen. Diese Vereinfachungen werden z.B. bei Anlieferschwankungen oder dem Ausfallsverhalten von Anlagen herangezogen. Die Frage nach der passenden Parametrisierung stellt sich auch im Bereich der Modellierung der Stochastik. Im Zweifel müssen Sensitivitätsanalysen, welche den Einfluss von Parameteränderungen auf das Ergebnis aufzeigen, in Bezug auf einzelne Modellkomponenten durchgeführt werden (vgl. März et al. 2011, S. 17).

Im Zuge des Designs der Zielfunktion existieren diverse Möglichkeiten, um diese zu verbessern bzw. eine mögliche Stagnation vorzubeugen. Die Ansätze beziehen sich z.B. darauf, dass möglichst viele Teile einer Lösung zu der aufgestellten Zielfunktion einen Beitrag leisten, damit sich eine Modifikation nach Möglichkeit durch einen neuen Zielfunktionswert entsprechend (signifikant) unterscheidet. Außerdem vermeidet bzw. reduziert dieser Ansatz sprungfixe Stellen durch die verbesserte (Auflösung der) Zielfunktion (vgl. Baresel et al. 2002, S. 1329-1336). Daher wird im Zuge der Validierung des in dieser Arbeit entwickelten Verfahrens auf frühzeitige Stagnation geachtet bzw. reduziert.

Grid Search

Nahezu jede der vorgestellten Methoden im nächsten Kapitel verlangt bestimmte Informationen, welche entweder durch den*die Analyst*in im Einklang mit dem*der Entscheidungsträger*in zu spezifizieren oder empirisch zu bestimmen sind. Neben problemspezifischen Parametern des jeweiligen Verfahrens sind dabei auch allgemeine Parameter wie z.B. die maximale Laufzeit des Verfahrens oder eine zu erreichende Mindestqualität bzw. Güte einer Lösung zu definieren, bevor das Verfahren abgebrochen werden kann. Wichtige Parameter stellen dabei z.B. die Ermittlung der geeigneten Schrittweite (bei Modifikationen einer Lösung) oder auch die Gewichtungen in einem Verfahren dar.

Um diesen Herausforderungen effizient zu begegnen, bietet sich vielfach eine Parameteroptimierung mittels z.B. eines *Grid-Search* Verfahrens an. Hierbei werden die Parameter nach oben und unten in Bezug auf ihren Wertebereich beschränkt und im Rahmen eines vorgegebenen *Rasters* stufenweise und sukzessive systematisch modifiziert, bis eine optimale Einstellung gefunden worden ist. Die Herausforderung liegt hierbei v.a. in der Wahl der geeigneten Schrittweite, welche in einem zweiten Schritt gegebenenfalls in einem bestimmten (vielversprechenden) Bereich verfeinert werden kann. Der einzige Nachteil daran ist, dass diese Methode relativ aufwendig in Bezug auf die geforderte Rechenintensität ist (vgl. Ramadhan et al. 2017, S. 626f.).

Blackbox-Probleme

Bei Optimierungsproblemen mit wenig bis keinerlei Informationen über die Zielfunktion, sogenannten *Blackbox-Problemen*, gehen die Variablen gemäß des ORMA in ein parametrisiertes (z.B. mathematisch-physikalisches) Modell ein, welches auf Basis der konkreten Modellierung entsprechende Ergebnisse zurückliefert. Die zulässige Menge S ist dabei von einfachen als auch komplexen statischen bzw. dynamischen Nebenbedingungen beschränkt (vgl. Kux 2011, S. 10).

Definition 13. Blackbox-Probleme sind in mehr oder weniger großem Ausmaß häufig durch die folgenden Eigenschaften charakterisiert (vgl. Kux 2011, S. 10ff.; Knowles 2006, S. 50ff.):

- Das Problem verfügt über den Definitionsbereich beschränkende, nichtlineare Nebenbedingungen sowie über mehrere kaum in einer Zielgröße miteinander messbare Zielsetzungen, welche teils nicht analytisch beschreibbar und häufig einander konkurrierend sind.
- Die Anzahl an durchführbaren Zielfunktionsbewertungen ist durch finanzielle, zeitliche oder andere ressourcenbedingte Gründe beschränkt, wodurch eine realistische Simulation bzw. vollumfängliche Suche nicht möglich ist.
- Die iterative Ausführung des jeweiligen Verfahrens ergibt (deterministisch bzw. stochastisch bedingt) gleiche oder sehr ähnliche Ergebnisse (geringes *Rauschen*), wobei prinzipiell große Verbesserungen in Bezug auf die Initiallösung möglich sind.
- Die Dimension des Suchraums S ist klein bis mittelgroß mit 5 – 30 Entscheidungsvariablen, die aufgespannte Fitnesslandschaft ist lokal glatt und multimodal. S ist nicht einfach zusammenhängend, im inneren Bereich von S gibt es einfach zusammenhängende, beschränkte Punktmengen, die nicht zum Definitionsbereich gehören (technisch nicht realisierbare Lösungen).
- Nur eine Funktionsbewertung kann gleichzeitig erfolgen, d.h. eine Parallelisierung ist verfahrensbedingt nicht oder nur eingeschränkt möglich.

In der Praxis treten meistens unterschiedliche Kombinationen dieser Eigenschaften auf, wobei diese Auflistung keinen Anspruch auf Vollständigkeit hat. Im Folgenden werden zwei Eigenschaften als Indikation für Blackbox-Probleme näher betrachtet und diskutiert, welche für die vorliegende Arbeit bzw. ein Problem im Bereich der Produktionsplanung von kritischer Bedeutung sind.

Widersprüchliche Zielsetzungen

Problemstellungen aus dem Bereich der Produktionsplanung sind vielfach widersprüchlich, wie auch eine entsprechende Analyse aus Hillier zeigt (vgl. Hillier und Lieberman 2015, S. 548ff.). Einer Produktion zugrunde liegende nichtlineare Funktionen, beispielsweise die Preis-Nachfrage-Kurve, zeigen, dass der Zusammenhang zwischen einem Verkaufspreis und der dazugehörigen Absatzmenge in den verschiedensten Ausprägungen nichtlinear ist. Speziell in der FMCG-Branche tritt dieses Phänomen auf, dass mit einer stufenweisen Preiserhöhung der Absatz unverhältnismäßig stark einbricht, wie im gegenteiligen Effekt die Nachfrage auch unverhältnismäßig stark (*exponentiell*) zunimmt, wenn eine Ware mit einem reduzierten Aktionspreis beworben wird. Grundsätzlich liegt bei der Produktion wie auch beim Transport von industriell hergestellten Produkten, der *economies of scale*-Ansatz zugrunde (vgl. Hillier und Lieberman 2015, S. 39ff.; Hennebel et al. 2017, S. 306ff.). Dieser Ansatz besagt,

dass Industrie- und Transportunternehmen verhältnismäßig deutlich günstiger produzieren können, wenn ihre automationsgestützten Maschinen bei der vollen Auslastung produzieren bzw. Transporte in möglichst großen Einheiten (z.B. auf Containerschiffen) transportiert werden. Der Grund dahinter ist, dass v.a. die Fixkosten im Zuge der Produktion bzw. während des Transports (in einem gewissen Ausmaß) konstant bleiben.

In der Produktion sind weitere widersprüchliche Zielsetzungen zu beobachten, klassischerweise die Funktion der Liefertermintreue gegenüber der Lagerhaltungskostenfunktion. Typischerweise verhalten sich die Maximierung der Termintreue und die Minimierung der Lagerhaltungskosten gegensätzlich, weil jeweils eines der beiden Kriterien auf Kosten des jeweils anderen verbessert wird. Abbildung 2.5 verdeutlicht diesen Zusammenhang (vgl. März et al. 2011, S. 71):

Auf einer Zeitachse sind beide Teilfunktionen der Termintreue in einer gemeinsamen Übertragungsfunktion abbildbar, wobei jedoch die vier in der Abbildung unterschiedlichen Funktionsbereiche nichtlinear verlaufen und *sprungfixe* Stellen beim Erreichen von gewissen Grenzwerten aufweisen (vgl. März et al. 2011, S. 71f.).

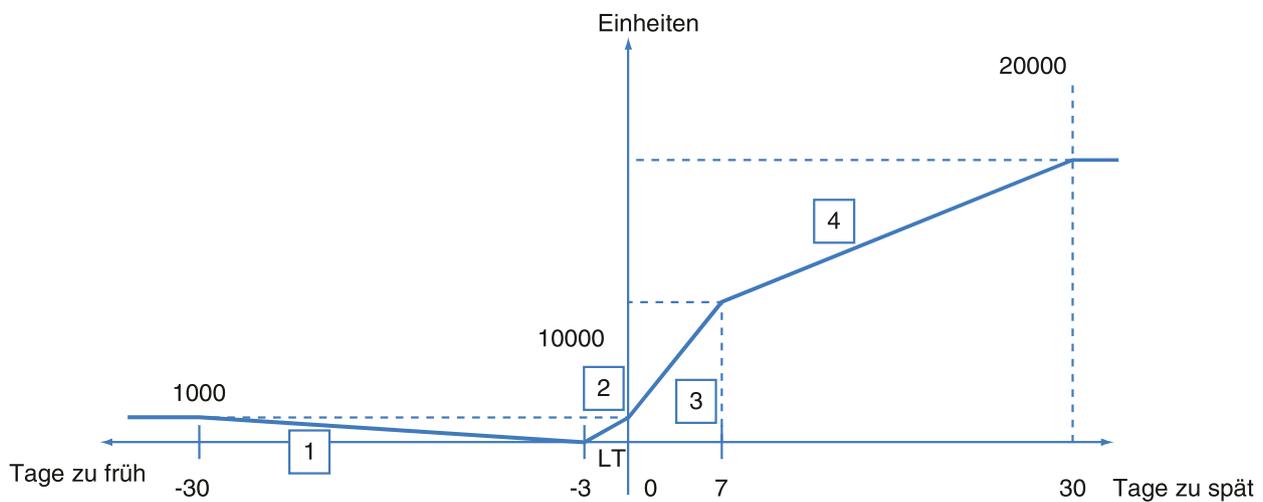


Abb. 2.5: Gegenläufigkeit von Liefertermintreue und Lagerhaltungskosten (März et al. 2011, S. 71)

Eine weitere Eigenschaft bei industrieller Produktion ist die Gegenläufigkeit von Rüstkosten und der Produktionslosgröße, speziell in der Lebensmittelbranche: Während das Risiko einer Charge mit einem spezifischen Ablaufdatum bei einer zunehmend größeren Losgröße mehr Risiko von Abschriften mit sich bringt, birgt eine Reduktion der Losgröße die Gefahr von ansteigenden Rüstkosten. Obwohl sich das Prinzip auch auf andere Industriezweige übertragen lässt (Kapitalbindungskostenrisiko im Falle des Nicht-Absatzes von bestimmten Produkten), ist diese Gegenläufigkeit in der Lebensmittelbranche bei der Fertigung einer jeden Charge allgegenwärtig, weil die mit dem*der Kund*in definierte Restlaufzeit (eine bestimmte Anzahl an Tagen bzw. Perioden) unmittelbar nach Ablauf nicht mehr verkaufbar ist. Dieses Kriterium spricht v.a. bei Produktion mit einer geringen Restlaufzeit für kleinere Chargen, welche gleichzeitig höhere Rüstkosten mit sich bringen. Ein weiteres Beispiel im industriellen Kontext ist der Zusammenhang zwischen der Qualität eines Produktes und seiner (Herstell-) Kosten. So lassen sich bei Reduktion entsprechender Qualitätsmaßnahmen die Kosten reduzieren, andererseits steigt dafür die Fehleranfälligkeit der Produkte und damit die Nachlaufkosten. Ein für diese Arbeit offensichtlich problemrelevantes Beispiel einer widersprüchlichen Zielsetzung liefert also die Gegenläufigkeit der Rüstkostenminimierung und einer Teilung von Produktionslosen.

Dellagi versteht unter dem *smoothing penalty* den Kompromiss bzw. die (Straf-) Kosten für erhöhte Rüstaufwände und untersucht in seinem Beitrag die entsprechenden Widersprüchlichkeiten (vgl. Dellagi et al. 2017, S. 262-270).

Analyse der zugrunde liegenden Fitnesslandschaft

Grundsätzlich spannen Design- und Zielraum bzw. Definitions- und Wertebereich einer Zielfunktion einen $n + 1$ -dimensionalen Suchraum auf, wobei n die Anzahl an Entscheidungsvariablen darstellt, von denen die Zielfunktion abhängt. Diesen Suchraum wird auch als Fitnesslandschaft bzw. Gebirge bezeichnet. Nebenbedingungen schränken den Suchraum derart ein, sodass der nächste diskrete Punkt in der Fitnesslandschaft möglicherweise weit vom letzten Punkt entfernt liegt. Der Abstand bezieht sich dabei auf die Ähnlichkeit im Designraum, wobei durch Nebenbedingungen *Sprünge* entstehen, welche sich auch im Zielraum entsprechend (anders) übertragen. Der*die Analyst*in kann diese Sprünge bei der Modellierung des Problems kaum vermeiden. Dementsprechend muss das Verfahren in der Lage sein, trotz dieser Sprünge die globale Sicht aufrechtzuerhalten. In Abbildung 2.6 ist ersichtlich, dass der Weg des steilsten Gradienten vom Ausgangspunkt A in Richtung B (durchgezogene Linie) nicht das globale Optimum findet, sondern lediglich ein *lokales* Optimum. Der längere Weg von A nach C (gestrichelte Linie) findet hingegen das *globale* Optimum. Der Punkt C ist allerdings nur durch das Überwinden eines Tals erreichbar. Die erste Route von A nach B ist deutlich kürzer und scheint zielstrebig und wird daher auch als *Bergsteigeralgorithmus* bezeichnet. Die andere Route von A nach C ist deutlich länger, erscheint weniger zielstrebig, und erlaubt neben Aufstiegen in Zwischengipfel auch Abstiege in Täler, bevor schlussendlich der höchste Gipfel bei Punkt C gefunden wird. Daher ist diese Strategie deutlich *robuster*, weil sie nicht im *lokal* gefundenen Optimum bei Punkt B verharret. Der Ausdruck *lokal* bezieht sich dabei v.a. auf die entsprechende Nähe von Punkt B zu Punkt A. Diese Distanz äußert sich, repräsentiert über eine bestimmte Metrik, durch das Erreichen weniger Optimierungsschritte (vgl. Schüller 2009, S. 34f.; Branke et al. 2008, S. 15f.).

Daher empfiehlt sich die Verwendung eines *robusten Optimierungsverfahrens*. Dieses ist in der Lage, mit großer Wahrscheinlichkeit mehrere Täler nach erreichten Zwischengipfeln überwinden zu können. Eine Fitnessfunktion mit mehreren lokalen Optima wird auch als *multimodal* bezeichnet und stellt die Optimierung vor größere Herausforderungen als eine *konvexe Funktion*, welche auch durch einen Bergsteigeralgorithmus bewältigbar ist. Am schwierigsten zu optimierende Fitnesslandschaften ohne erkennbare Systematik in der Anordnung der Gipfel verfügen über eine auf Grund von Nebenbedingungen stark zerklüftete, multimodale Fitnesslandschaft mit mehreren großen Tälern und wenigen herausragenden *Plateaus* bzw. Gipfelpunkten. Abgesehen von der Hinzunahme von problemspezifischem Wissen kann hier das Verhalten von Algorithmen ausgenutzt werden, welche ausgehend von mehreren Lösungen den Suchprozess durchführen (vgl. Schüller 2009, S. 35f.).

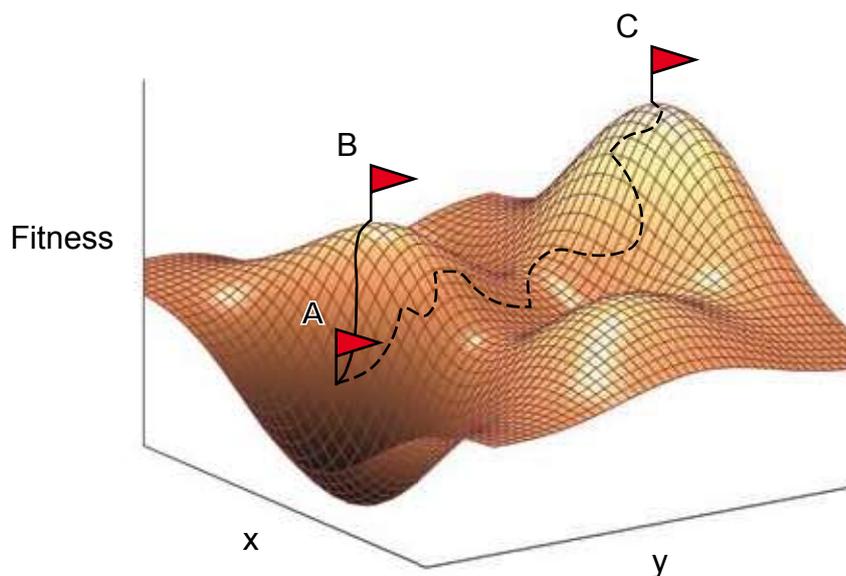


Abb. 2.6: Lokale und globale Optima in einer multimodalen Fitnesslandschaft (Schüller 2009, S. 34)

2.3.3 Fazit

Zusammenfassend lässt sich festhalten, dass nichtlineare multikriterielle Optimierungsprobleme, vor allem mit ihren Eigenschaften als Blackbox- bzw. NP-schwere Probleme, gemäß der betrachteten Problemgröße und Komplexität, schwierig lösbar sind und der Analyse und Modellierung des Problems (vgl. Hillier und Lieberman 2015, S. 10) bzw. der Parametrisierung des Verfahrens die höchste Priorität zukommt. Im Zuge der Verfahrensauswahl und Entwicklung wird auf die Laufzeitkomplexität des Verfahrens sowie die Multimodalität der Fitnesslandschaft, bedingt durch die Widersprüchlichkeit der einzelnen Zielsetzungen (vgl. März et al. 2011, S. 71f.), gezielt Rücksicht genommen, sodass anstelle von lokalen Optima auch Pareto-optimale Lösungen, entsprechend globalen Optima, gefunden werden. Eine Parameteroptimierung wird nach Möglichkeit mittels Grid-Search integriert (vgl. Ramadhan et al. 2017, S. 626f.). Darüber hinaus empfiehlt es sich, nach Möglichkeit eines der angesprochenen Verfahren zur Skalarisierung zu verwenden, um das multikriterielle Problem in ein einkriterielles Problem zu transferieren und anschließend mit einem robusten Optimierungsverfahren zu lösen (vgl. Schüller 2009, S. 35f.). Die entsprechenden Grundlagen diesbezüglich als auch dazugehörige, ausgewählte Optimierungsverfahren werden in Kapitel 3 ausführlich diskutiert. Im Zuge der Produktionsproblembeschreibung wird in Kapitel 4 bestimmt, ob ein Blackbox-Problem, so die Hypothese, vorliegt.

2.4 Thematische Eingliederung innerhalb des Operations Research

Die thematische Einordnung der vorliegenden Kombination eines MOMPMP und MICLS Problems ist, basierend auf der allgemeinen Problemstellung, in Abschnitt 1.2 erfolgt. Der Zusammenhang zwischen Just-In-Time (JIT) und geglätteten Losgrößen wird in der Literatur häufig diskutiert, wobei der JIT-Ansatz die folgenden Anforderungen proklamiert (vgl. Karimi-Nasab und Konstantaras 2012, S. 479f.):

1. Minimale Produktionsabweichungen gegenüber der Nachfrage.
2. Verfügbarkeit eines betreffend den Engpass geglätteten Produktionsplans über den gesamten Planungshorizont.
3. Verfügbarkeit einer geglätteten Kapazitätsauslastung in Bezug auf das betrachtete Produktionssystem.
4. Erreichung minimaler Bestände bzw. Zielbestandsniveaus.

Diese seitens JIT gestellten Anforderungen gelten großteils auch für diese Arbeit aus Sicht der Mittel- bis Langfristplanung, wobei die Anforderungen in Bezug auf den speziellen Anwendungsfall angepasst bzw. relaxiert, d.h. mit einer Toleranz versehen, werden. Die Zielsetzungen können für den jeweiligen Anwendungsfall erweitert bzw. reduziert werden und erfordern die Aufstellung eines mathematischen Modells, welches der betrieblichen Realität möglichst nahe kommt und verschiedene generierte bzw. veränderte Produktionspläne entsprechend realitätsnah und laufzeiteffizient bewerten kann. Auf Grund der in Abschnitt 1.2 hergeleiteten Komplexitätsklasse erfordern Probleme aus dieser Klasse grundsätzlich multikriterielle Lösungsverfahren (vgl. Karimi-Nasab und Konstantaras 2012, S. 479ff.; Wari und Zhu 2016, S. 328ff.).

Die Vielzahl an geeigneten Lösungsverfahren, sowie deren jeweiligen Vor- und Nachteile, werden in Kapitel 3 diskutiert. Darauf basierend wird eine Zusammenfassung für die Anforderungen der im Rahmen dieser Arbeit entwickelten Methodik erstellt.

2.5 Systematische Eingrenzung des Betrachtungsbereichs

Auf Basis der in den vorangegangenen Teilkapiteln erfolgten Strukturierungen und Begriffsdefinitionen, lässt sich der Untersuchungsbereich im Zuge dieser Arbeit methodisch auf die folgenden Schwerpunkte und Themenkomplexe eingrenzen:

1. Erreichung eines rollierend (kontinuierlich, z.B. wöchentlich, angepassten) **optimierten Produktionsplans** mit Fokus einer Harmonisierung der Produktionsausstoßrate auf Produktionsartikel- und Werksebene.
2. Umsetzung und Erreichung einer **gesamtheitlich betrachtet optimierten Kapazitätsplanung und -auslastung** (Anlagen und Maschinen sowie Personal) im betrachteten Planungshorizont.
3. Integration einer Methodik zur Erstellung und Bewertung einer möglichst realistischen **Vorhersage zur Entwicklung der Kapazitätsauslastung** zur Abschätzung der prinzipiellen Umsetzbarkeit einer Lösung, abgestimmt auf den vorliegenden Planungshorizont.
4. Integration einer Methodik zur **rollierenden Stammdatenoptimierung**.

Abb. 2.7 gibt eine Übersicht in Bezug auf die systematische Eingrenzung in Bezug auf die verschiedenen Dimensionen des Betrachtungsbereichs dieser Arbeit in Form von grünen Kästchen. Der betrachtete Teil der Supply-Chain stellt den Produzenten dar, wobei der betrachtete Ausschnitt der zur Leistungserstellung notwendigen Produktionsprozesse umfangmäßig betrachtet wird. Die Planungsaufgabe besteht v.a. in der Harmonisierung der Produktionsausstoßrate in Bezug auf den vorliegenden mittelfristigen Planungshorizont.

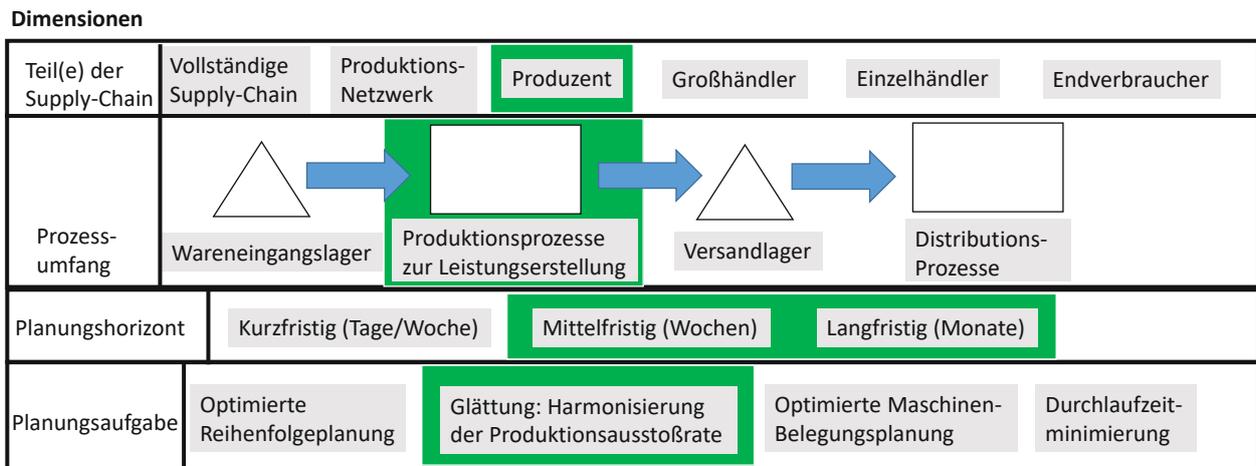


Abb. 2.7: Systematische Eingrenzung des Betrachtungsbereichs (Quelle: eigene Darstellung)

In weiterer Folge wird eine Diskussion zur Eingrenzung der einzelnen Dimensionen des Betrachtungsbereichs vorgenommen, bevor ein Vergleich mit dem aktuellen Stand der Technik erfolgen kann. Diese Eingrenzung erfolgt im Zuge der Reduktion auf die genannten wesentlichen Merkmale und Charakteristika für die Anwendbarkeit der Methodik in den Merkmalsausprägungen Produzent (Branche) und Produktionsprozesse zur Leistungserstellung sowie Planungshorizont und Planungsaufgabe.

2.5.1 Eingrenzung auf Grund der Charakteristika des Produzenten (Branche) und Produktionsprozesses

Diese Arbeit fokussiert sich auf charakteristische **Produktionsabläufe von lebensmittelverarbeitenden Unternehmen**. Diese Unternehmen, welche i.d.R. KMU darstellen², können einen **variabilisierbaren**³ **Produktionsprozess** aufweisen, z.B. das Heranreifen von Produkten in einem definierten Zeitfenster, welcher für alle Produkte in einem bestimmten Ausmaß gilt. Typische Vertreter dieser **spezifischen Branche** stellen dementsprechend Produzenten von Speck, Käse oder auch Wein dar. Der Mehraufwand durch einen (in der zeitlichen Ausdehnung teils) variablen Produktionsprozess muss in Kosten bewertbar sein. Diese Einschätzung muss dabei für jeden konkreten industriellen Anwendungsfall (bzw. Vertreter der Branche) individuell vorab der Modellierung erfolgen. Eine Relaxation dieser Anforderung führt zu einer (zumindest partiellen) Emanzipation (siehe auch Abb. 2.1.), welche jedoch nicht mehr den Produktionsprozess selbst variiert, sondern über das nachgelagerte Fertiglager eine entsprechende Mengenbevorratung vornimmt. Der zweite damit verbundene Unterschied ist, dass die Mengenbevorratung über das Lager fertige Produkte beinhaltet, während ein variabler Produktionsprozess eine differenzierte Produktionsausstoßrate von (i.d.R., mit Ausnahme des letzten Produktionsschrittes) Zwischenprodukten darstellt. Obwohl die in dieser Arbeit entwickelte Methodik **nicht branchenspezifisch** ist, umfasst sie nicht die **Komplexität einer Methodik für die Optimierung von Produktionsabläufen mit Stücklisten bzw. einer industriellen Vor- und Endmontage**.

Der konkrete industrielle Anwendungsfall für die in dieser Arbeit entwickelte Methodik für Produktionsglättung begünstigt den Umstand, dass die anfallenden **(Um-) Rüstkosten** (z.B. am Engpassaggregat) in Relation zu den übrigen Kostenblöcken (Herstellkosten, Materialkosten, etc.) mit anderen Branchen (Metall- und kunststoffverarbeitende Industrie) vergleichsweise günstig bzw. **idealerweise vernachlässigbar** sind. Obwohl die Anzahl an Rüstvorgängen bzw. die damit verbundenen Rüstkosten immer bewertbar sein müssen, ist es eine methodische Einschränkung, wenn eine Losteilung, welche die Anzahl an Umrüstkosten i.d.R. erhöht, nicht sinnvoll anwendbar ist. Dementsprechend wichtig ist die vorherige Klärung mit der Planungsabteilung, ob im Zuge der Herangehensweise eine Losteilung über die Planungsperioden eines **MOMPMP** hinweg, siehe Abschnitt 1.2, möglich ist und sich für die Modellierung des Problems dadurch ein zusätzlicher Freiheitsgrad betreffend die Anzahl an Produktionslosen ergibt (vgl. Karimi-Nasab und Konstantaras 2012, S. 481). Im Falle des vorliegenden Problems ist dieser Freiheitsgrad gegeben.

Darüber hinaus fokussiert sich die Methodik speziell auf die **Reduktion der Volatilität von Absatz- und Produktionsschwankungen** im Rahmen des Produktionsprozesses. Diese Schwankungen ergeben sich einerseits auf Grund des, durch Aktionen verstärkt, **konsumentengetriebenen Einkaufsverhaltens** als auch andererseits durch die **Saisonalität des Absatzverhaltens der Artikel** mit den jeweiligen Auswirkungen auf das gesamte Produktionsspektrum.

In Bezug auf den Auftragsstyp wird an dieser Stelle darauf verwiesen, dass die zu entwickelnde Methodik auf die Glättung einer (Groß-) **Serien- oder Massenfertigung** abzielt, weil bei einer Einzelfertigung oder Kleinserienfertigung einerseits zumeist der zeitlich dispositive Charakter entfällt als auch eine Losteilung nicht ökonomisch sinnvoll ist. Die Anzahl an verschiedenen (zu fertigenden) Produktionsartikeln ist für die Methodik prinzipiell gleichgültig bzw. vernachlässigbar, wenngleich Auswirkungen auf die Problem- und Laufzeitkomplexität betrachtet und möglichst reduziert werden müssen. Die **Methodik soll demnach einerseits für wenige als auch viele verschiedene Produktionsartikel mit möglichst geringem Zuwachs an (Problem- und Laufzeit) Komplexität** im Sinne einer Änderung bzw. Ausdehnung des Produktspektrums oder auch Planungshorizonts zurecht kommen.

²In Bezug auf Großunternehmen existieren bereits meistens APS-Lösungen, welche im gesamten Konzernverbund verwendet werden.

³Die Eigenschaft *variabilisierbar* bedeutet im Kontext von Produktion und Logistik, dass dieser (Produktions) Prozess noch nicht variabel ist, aber prinzipiell (teils) variabel gestaltet werden kann.

2.5.2 Eingrenzung auf Grund der Anforderungen an den Planungshorizont und die Planungsaufgabe

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung einer multikriteriellen Methodik für eine optimierte Mittel- bis Langfristplanung. Die Produktionsplanung erfolgt dabei über den gesamten Planungshorizont gegen begrenzt verfügbare Kapazitäten. Die konkrete Planungsaufgabe erfolgt rollierend. Der mittel- und langfristige Planungscharakter bedeutet, dass sich diese Arbeit nicht der täglichen Produktionsfeinplanung, sondern der wöchentlichen Grobplanung widmet. Die Anforderungen dieser optimierten Mittel- bis Langfristplanung beinhalten einerseits eine Glättung aller Produktionsartikel auf Artikel- als auch auf Werksebene. Eine Glättung auf Artikelebene bedeutet, dass jeder einzelne Artikel über eine möglichst gleichmäßige Produktionsausstoßrate verfügt. Eine Glättung auf Werksebene bedeutet hingegen, dass das gesamte Werk über eine im Idealfall gleichmäßige Produktionsausstoßrate verfügt. Diese Zielstellungen können, je nach dem konkreten industriellen Anwendungsfall, konkurrierender als auch einander unterstützender Natur sein.

Außerdem wird die Planungslösung einen Beitrag im Bereich einer möglichst gleichmäßigen Kapazitätsauslastung leisten, d.h. die Aggregate und Produktionsräumlichkeiten werden nach Möglichkeit bestmöglich ausgelastet. Dies kann vor allem durch eine möglichst gleichmäßige Auslastung erreicht werden. Dies bedingt eine größtmögliche Langlebigkeit mit den vorhandenen Werksstrukturen als auch eine minimierte Gesamtauslastung aller Aggregate. Dieser Umstand resultiert in minimierten Energie- und Materialflusskosten bedingt durch eine geringere in Betrieb befindliche Anzahl an Produktionsaggregaten.

Darüber hinaus verbessert die mittel- bis langfristige Methodik auch die Bestandsniveaus des gesamten Produktionsartikelspektrums.

Der konkreten Planungsaufgabe wird die Annahme zugrunde gelegt, dass die zur Produktion notwendigen Input-Faktoren (v.a. zur Produktion erforderliche Rohstoffe) auch in der entsprechenden zeitlichen Dimension (z.B. Woche) in der erforderlichen Menge von den Lieferant*innen entsprechend beschafft werden können. Andernfalls ist das Ergebnis der Methodik, der konkrete Produktionsvorschlag, nicht umsetzbar. Diese Anforderung gilt prinzipiell auch für die Planung, unabhängig von der Ausführung der in dieser Arbeit entwickelten Methodik, mit der folgenden Ausnahme, dass sich die Planung und Produktion an die Lieferant*innen beschaffungsseitig systematisch angepasst haben. Bei Vorliegen einer derartigen Konstellation muss diese Nebenbedingung (NB) in die Optimierung des Produktionssystems aufgenommen werden.

Die Eingrenzung dieser Arbeit auf eine optimierte Mittel- bis Langfristplanung (rollierende aggregierte Wochenproduktionsmengen) gegenüber einer Feinplanung (Reihenfolgeplanung) auf Auftragsebene proklamiert, dass die wöchentlich (festzulegende) Produktionsreihenfolge im Rahmen dieser Arbeit nicht betrachtet bzw. optimiert wird. Diese Herausforderung ist Aufgabe einer Feinplanung, die vom Unternehmen selbst auf Basis der optimierten Mittel- bis Langfristplanung erstellt wird.

Die im Rahmen dieser Arbeit entwickelte Methodik für die mittelfristige Grobplanung glättet bzw. optimiert die in jeder Periode, d.h. Woche, zu produzierenden Mengen in aggregierter Form und beschränkt ihren Beitrag dadurch auf die in einer jeden Planungsperiode zur Verfügung gestellten, auf Produktionsartikelebene kumulierte, Produktionsvorschläge. Es erfolgt keine explizite Losgrößenoptimierung, obwohl die Losteilung in die Methodik integriert und über die Zielfunktion bewertet wird. Die Feinplanung (Reihenfolge und Terminierung sowie endgültige Losgröße pro Tag) innerhalb einer jeden Periode übernehmen die Mitarbeiter*innen der Planungsabteilung selbst. Diese verfügen über (tages-) aktuelles und implizites Wissen (z.B. Änderung von Anliefermengen), welches im Zuge der Methodik für eine Mittelfristplanung nicht sinnvoll abbildbar ist.

2.6 Methodische Problemabgrenzung

Die allgemeine Problemstellung, siehe Abschnitt 1.2, erlaubt eine methodisch-inhaltliche Abgrenzung. Das Problem wird dabei einerseits gegenüber Lösungsverfahren aus der Absatzplanung zur Absatzprognoseglättung sowie andererseits gegenüber Kapazitätsflexibilitätsstrategien und kommerziellen Planungssystemen abgegrenzt. Dies wird in den nächsten Unterabschnitten erläutert und diskutiert.

2.6.1 Abgrenzung gegenüber Absatzprognoseglättung

Absatzprognoseoptimierung, auch als *Absatzprognoseglättung* bezeichnet, beschäftigt sich grundsätzlich mit der Erstellung möglichst genauer Absatzvorhersagen (vgl. Morawetz 2015, S. 174) bzw. damit verbunden, der Reduktion des dazugehörigen gesamten Absatzprognosefehlers (vgl. Tratar et al. 2016, S. 162ff.), welcher sich über alle Perioden kumuliert berechnet. Von den zur Verfügung stehenden Methoden werden dabei in der Literatur als auch in der betrieblichen Praxis, z.B. im Rahmen der Anwendung eines ERP-Systems, die **exponentielle Glättung erster Ordnung** (vgl. Morawetz 2015, S. 182f.) oder auch die **Holt-Winters Methode** (vgl. Tratar et al. 2016, S. 163f.) verwendet. Die erste Methode ist Standard in diversen ERP-Systemen, beispielsweise im Advanced Planner and Optimizer (APO) Modul von SAP (vgl. Bothe und Nissen 2003, S. 53ff.), während die zweite Methode, auf Grund der Mehrzahl an Prognoseparametern, komplizierter ist und vielfach nicht angewendet wird bzw. in einer ERP-Umgebung nicht verfügbar ist.

Die exponentielle Glättung ist für die Ermittlung von Absatzprognosewerten auf Basis von Vergangenheitsdaten ohne systematischem Muster geeignet, wobei in der jüngeren Vergangenheit liegenden Datenpunkten eine höhere Gewichtung zugeordnet wird (vgl. Morawetz 2015, S. 182f.). Die Absatzprognoseoptimierung beschäftigt sich mit dem Umstand, wie eine möglichst gute Gewichtung in Bezug auf alle vorliegenden Datenpunkte aussehen kann, damit der Gesamtprognosefehler minimiert wird. Daher muss der für die erfolgreiche Verwendung der Methode, für einen jeden Artikel, der sogenannte *Glättungsfaktor* α möglichst optimal bestimmt werden. Diese Glättung hat, mit Ausnahme des Namens und damit verbunden einer grundsätzlichen mathematischen Optimierung, keine inhaltlichen Gemeinsamkeiten mit einer Methodik zur Glättung von Produktionsmengen.

Debnath gibt einen umfassenden Überblick über den aktuellen Stand der Technik bezüglich Absatzprognosemethoden und hält, unter der Auswertung von 483 Prognosemodellen, den folgenden Sachverhalt fest (vgl. Debnath und Mourshed 2018, S. 297-325):

- Artificial Intelligence (AI) Ansätze sind rein statistischen Ansätzen zumeist überlegen.
- Hybride Methoden sind reinen AI-Ansätzen überlegen.
- Bei Vorliegen nichtkontinuierlicher oder nichtlinearer Daten sind ML-Methoden den statistischen Methoden vorzuziehen.
- Bei unklarem oder nicht gegebenem Zusammenhang zwischen den Variablen sind ML-Methoden statistischen Methoden überlegen.

Eine optimierte Absatzprognose mit einem minimalen Prognosefehler bedeutet nicht, dass diese Prognose in der Zukunft derart eintritt, sondern ist lediglich ein mathematisch fundiertes Modell, auf dessen Basis die Absatzplanung die jeweilig verwendete Methode bzw. die dazugehörige Parametrisierung rechtfertigt. Für eine fundierte Absatzplanung sind viele weitere Faktoren und Informationen, wie z.B. die Abrufinformationen von Kund*innen, Aktions- und Kurantmengenanalysen auf Plankundenebene, (Aktions-) Preisinformationen, Veränderungen in der Leistungsbreite, allgemeine Vertriebsinformationen als auch Informationen über die Konkurrenz oder das Wetter, notwendig. Die Gesamtheit dieser Informationen liegt zumeist nur unvollständig vor. Deshalb ist die Erstellung akkurater Absatzpläne auf Artikel- und Plankundenebene auch entsprechend herausfordernd.

Eine inhaltliche Verbindung zwischen Absatzplanung und Produktionsplanung ergibt sich insofern, als die Ergebnisse der (optimierten) Absatzplanung den Input für die Produktion bzw. die Produktionsglättung darstellen. Dementsprechend ist vor der Entwicklung einer geeigneten Methodik für eine

multikriterielle Produktionsglättung einer systematisch durchgeführte Absatzprognose umzusetzen. Der Input aus der Absatzplanung stellt die Grundlage für die nachfolgende Produktionsglättung dar.

2.6.2 Abgrenzung gegenüber Kapazitätsflexibilitätsstrategien

Auf Grund des dispositiven Charakters der Methodik Produktionsglättung wird eine Kapazitätsflexibilitätsbewertung im Zuge dieser Arbeit explizit nicht durchgeführt. In der Literatur existieren zahlreiche Ansätze zur Messung der Kapazitätsflexibilität (vgl. Morawetz 2015, S. 45ff.), konkret z.B. die Flexibilitätsbewertung nach Rogalski (vgl. Rogalski 2009, S. 61-134) bzw. Gottschalk (vgl. Morawetz 2015, S. 54; Gottschalk 2005, S. 72f.).

Rogalski unterscheidet zwischen drei Flexibilitätsarten und weist einer jeden daraus quantifizierbare Beschreibungsgrößen zu. Der grundsätzliche Bewertungsansatz definiert beispielsweise, dass die Mengenflexibilität eines Produktionssystems im Verhältnis zur Abweichung zwischen Break-even Punkt und wirtschaftlicher Maximalkapazität steht (vgl. Rogalski 2009, S. 61-134).

Gottschalk entwickelt sein Konzept der Flexibilitätsprofile, erweitert es um die Dimension Kosten und leitet daraus eine mengen- als auch kostenbasierte Kennzahl zur Bewertung der Flexibilität ab. Diese aufwendig zu erstellenden Profile bieten einen deutlichen Mehrwert, weil sie Auswirkungen unterschiedlicher Maßnahmen berücksichtigen, visualisieren und in ihrem Ausblick zukünftige Optionen als Entscheidungsunterstützung anbieten. Eine nähere Betrachtung der von Gottschalk verwendeten Fallstudien am Beispiele einer Kapazitätsanpassung bei Hilti zeigt, dass Gottschalk dabei bestimmte Szenarien bzw. Strategien (Zusatz- und Freischichten, kurzfristige Änderung des Schichtbetriebs, Nutzung des nachgeordneten Bestandpuffers, Zukauf eines zusätzlichen Umformwerkzeugs, etc.) definiert, visualisiert und anschließend bewertet gegenüberstellt und die jeweils beste Strategie empfiehlt (vgl. Gottschalk 2005, S. 131ff.).

Den Flexibilitätsstrategien ist gemeinsam, dass sie auf die Erstellung einer einfach für eine*n Entscheidungsträger*in zu interpretierenden absoluten oder relativen Kennzahl abzielen. Auf Basis der Kennzahl wird eine Grundlage für eine Verbesserung geschaffen. Entscheidender ist allerdings, wie viel Flexibilität wirtschaftlich sinnvoll ist, und wie viel Anpassungskosten die Flexibilität grundsätzlich kostet. Morawetz hält außerdem fest, dass diverse in seiner Dissertation vorgestellte Ansätze zur Messung der Kapazitätsflexibilität geeignet sind, jedoch nur der visualisierende Ansatz von Gottschalk einen tatsächlichen Mehrwert bei der Ermittlung der kostenoptimalen Kapazitätsanpassungsstrategie bietet. Seine Analyse zeigt auf, dass es kein Verfahren gibt, welches dem*der Planer*in konkret und branchenübergreifend bei der Ermittlung von optimalen Kapazitätsanpassungsstrategien unterstützt, wobei einige aggregierte Produktionsplanungs Modelle zumindest eine Entscheidungsunterstützung anbieten, und damit prinzipiell die laufende praktische Anwendung ermöglichen, wenn sie den Anforderungen eines Industriebetriebs entsprechen würden (vgl. Morawetz 2015, S. 73f.).

Grundsätzlich lässt sich schlussfolgern, dass Kapazitätsflexibilitätsstrategien ein zusätzliches oder auch alternativ geeignetes Instrument darstellen, um Volatilität in einer Produktion zu begegnen. Allerdings sind die Strategien immer im konkreten Anwendungsfall (neu) zu bestimmen. Dieser zusätzliche Aufwand sowie der Umstand, dass diese Strategien immer direkt mit (Flexibilitäts-) Anpassungskosten behaftet sind, führt dazu, dass diese Ansätze im Rahmen dieser Arbeit nicht mehr weiter betrachtet werden. Bei der im Zuge dieser Arbeit entwickelten Methodik fallen nur einmalig Kosten für die Entwicklung und Validierung bzw. Umsetzung an. Außerdem weisen die meisten dieser Strategien lediglich einen kurz bis mittelfristigen Charakter auf und lösen daher nur die aktuelle Situation, ohne jedoch das Problem an der Wurzel, also der Behebung des Bullwhip-Effekts, anzugreifen.

2.6.3 Abgrenzung gegenüber kommerziellen Advanced Planning and Scheduling Systemen (APS) und Solvern

Abschnitt A.10 des Anhangs beschreibt die Entwicklung von kommerziellen ERP- und APS-Systemen. Die vorhandenen Module werden um spezielle Funktionen erweitert, um den jeweiligen industriellen Anforderungen gerecht zu werden. Dieses Problem wird deutlich größer, wenn keine konzernähnliche Netzwerkstruktur vorhanden ist, und ein KMU die gesamte SAP-Funktionalität inklusive APO erwirbt, und sich anschließend noch die ganzen speziellen Funktionen individualisiert zur Verfügung stellen lässt, um die (branchen-) spezifischen Anforderungen abzudecken (vgl. Bothe und Nissen 2003, S. 70f.). Prinzipiell verfügt ein APS-System wie z.B. SAP APO über keine auftragsindividuelle Losgrößen-Optimierung, sondern teilt eine bestehende Auftragsmenge in so viele (auf Losgrößen basierende) Aufträge ein, wie sich aus der Berechnung ergibt (vgl. Bothe und Nissen 2003, S. 56). Es existieren neben starr hinterlegten Losgrößen weitere Losgrößenmethoden, z.B. die exakte Losgröße, die exakt auf den Verbrauch reagiert, oder sogenannte Mindest- und Maximallosgrößen⁴ (vgl. Dickersbach 2009, S. 245f.). Dementsprechend fehlt eine problemspezifische Zielfunktion, welche die Gleichmäßigkeit der Werksauslastung an der Engpass-Anlage bzw. für alle Anlagen eines Werksverbands bewertet.

SAP (APO) plant die Aufträge auf Basis der Rückwärtsterminierung entsprechend ein (vgl. SAP Help Portal 2021b). Darüber hinaus bietet die Software keine explizite dynamische Simulation oder eine entsprechende datenbasierte Auswertung einer Optimierung an. Das Thema Simulation wird nur im Zuge der Bewertung verschiedener Planungsszenarien aufgegriffen (vgl. Bothe und Nissen 2003, S. 50f.). Grundsätzlich ist es möglich, die in dieser Arbeit vorgestellte Glättungsmethodik auch in SAP (APO) zu implementieren. Allerdings ist dies nur dann sinnvoll, wenn das jeweilige Unternehmen SAP bereits verwendet. Neben der laufenden Pflege des Systems, in Analogie zu einer Datenbank, müssen auch die Mitarbeiter*innen entsprechend im Umgang mit dieser Software geschult werden⁵.

Auf Grund der hohen Lizenzgebühren sowie der aktuell mangelnden Verfügbarkeit eines maßgeschneiderten Optimierungsmoduls werden Vertreter von APS-Systemen in dieser Arbeit ausgegrenzt. In der Regel verfügen APS-Systeme lediglich über single-kriterielle Planungsheuristiken, die ohne Losteilung von expliziten Aktionsobjekten auskommen. Die Planungsheuristiken von SAP sind in der Dokumentation (vgl. SAP Help Portal 2021a) beschrieben und umfassen beispielsweise eine Reihenfolgeplanung, eine Durchlaufzeitreduktion oder auch eine kapazitätsgetriebene heuristische Planung (vgl. Bothe und Nissen 2003, S. 60f.). Im Zuge der Produktionsplanung in APO gibt es 2 standardisierte Heuristiken, allen voran die MRP-Heuristik (vgl. Dickersbach 2009, S. 300-308). Die verschiedenen APO-Anwendungen verfügen über die (teilweise über Upgrade eingeschränkte) Möglichkeit der begrenzten Planung bzw. alternativen Ressourcenbelegung (vgl. Dickersbach 2009, S. 239-245).

Kallrath beschreibt im Rahmen der Optimierung mit APO, dass intern der 'CPLEX' Optimizer von IBM ILOG verwendet wird (vgl. Kallrath und Maindl 2006, S. 61ff.; IBM ILOG 2021). Für reale Probleme kann also auch die Meldung zurückgegeben werden, dass das gegebene Problem nicht lösbar ist. Obwohl CPLEX, neben FICO Xpress, als weltweit bester Solver bekannt ist, muss die Optimierung daher auf dieser Solver-Umgebung definiert und modelliert werden. Im Rahmen eines Anwendungsfalls aus der Prozessindustrie wird die Frage gestellt, ob ein bereits zuvor industriell gelöstes multi-location multi-period (MLMP) Problem auch im Rahmen einer Einführung von SAP APO gelöst und damit ersetzt werden kann oder ob das bestehende (bereits funktionierende) Optimierungs-Modul über ein Interface angesprochen wird (vgl. Kallrath und Maindl 2006, S. 143ff.). Dem APO Optimierungshandbuch ist zu entnehmen, dass prinzipiell zwei Szenarien eine multikriterielle Zielsetzung beinhalten (vgl. Kallrath und Maindl 2006, S. 187ff.). Allerdings gibt es für das zu formulierende Optimierungsmodell aktuell keine entsprechend verfügbare Methode in APO. Als grundlegende Einschränkungen seitens des Optimizers in APO nennt Kallrath (vgl. Kallrath und Maindl 2006, S. 199-201), in Bezug auf ein Beispiel aus der Prozessindustrie, folgende Punkte:

⁴Der Autor dieser Dissertation hat als SAP Key-User z.B. in APO die Mindestlosgrößen pro Artikel festgelegt sowie darüber hinaus die jeweiligen Inkrement-Beträge. Diese Inkremente wurden i.d.R. auf Basis ganzer Paletten festgelegt.

⁵Der Autor dieser Dissertation wurde z.B. bei der Einführung von SAP-APO im Zuge seiner früheren beruflichen Tätigkeit als Absatzplaner ca. 1/2 Jahr professionell im Umgang mit SAP-APO als Key-User eingeschult.

- Die Hinterlegung von probabilistischen (Transport-) Zeiten wird nicht unterstützt, d.h. Zeiten werden ausschließlich konstant modelliert.
- Die Kapazitätsbelegung inkl. Nebenbedingungen erfolgt nur in einer Auflösung von Zeitfenstern.
- Reihenfolgeabhängige Rüstzeiten werden nicht berücksichtigt, darüber hinaus werden auch zustandsabhängige Rüstzeiten nicht modelliert: Rüstzeiten werden gemäß einer Rüstzeitenmatrix stets nur als Konstante definiert (vgl. Kallrath und Maindl 2006, S. 139).
- Minimale Produktionslosgrößen auf Artikelebene werden nicht exakt modelliert, sondern über *Workarounds* auf Basis der Kostenstruktur bzw. definierter Mindestlosgrößen.

Ein weiterer, auf Grund seiner Benutzerfreundlichkeit populärer Solver für Probleme aus dem Bereich OR ist LINDO (vgl. Hillier und Lieberman 2015, S.8-151). Eine gute Übersicht betreffend die Leistungsfähigkeit der unterschiedlichsten Solver befindet sich auf der General Algebraic Modeling System (GAMS) Homepage (vgl. GAMS Software GmbH 2021). Der erste von den genannten Punkten in der obigen Aufzählung ist entscheidend, weil damit ausgesagt wird, dass die laut Arbeitsplan (fest) definierten Durchlaufzeiten konstant und daher in jedem Fall einzuhalten sind. Dies deckt sich ggf. nicht mit den Anforderungen an die in dieser Arbeit zu entwickelnde Methodik, wenn zumindest ein **variabilisierbarer Prozess als Anforderung für die Anwendung der Produktionsglättungsmethodik genannt wird**. Dadurch bleiben Möglichkeiten, welche auf Grund der Anwendungsumgebung eingeschränkt werden, ungenutzt. Da in den Anwendungsbeispielen ausschließlich statische Nebenbedingungen formuliert werden, ist die Integration von dynamischen Nebenbedingungen nicht belegt.

2.7 Fazit

Dieses Kapitel gibt einen Überblick über die Grundlagen von Produktion und Optimierung und legt den Fokus dabei auf die Betrachtung der Produktionsglättung. Abschnitt 2.3 erläutert die grundlegendsten mathematischen Begriffe und Konzepte aus der nichtlinearen Optimierung sowie Herausforderungen bei der Modellierung von Blackbox-Problemen. Neben der thematischen Eingliederung innerhalb des OR erfolgt eine systematische Eingrenzung des Betrachtungsbereichs auf Grund der Charakteristika des Produzenten bzw. seiner Branche sowie der Fokus auf Produktionsprozesse zur Leistungserstellung. Außerdem erfolgt eine Eingrenzung basierend auf Planungshorizont- und Aufgabe.

Hinsichtlich der Entwicklung und Anwendung der Produktionsglättung im Kontext dieser Arbeit wird in Bezug auf deren Eignung gefordert, dass mindestens ein ökonomisch sinnvoll (teilweise) vorverschiebbarer oder variabilisierbarer Prozess, idealerweise für (möglichst) viele bzw. alle Artikel besteht. Dieser variiert in seiner Ausführungsdauer idealerweise relativ stark und bleibt dabei, trotz einer möglichen Verlängerung der Prozessdurchlaufzeit, möglichst kostengünstig. Je mehr dieser Prozess in Relation zu seiner Durchlaufzeit variieren kann, desto größer ist das damit verbundene Potential der Methodik. Durch den mittel- bis langfristigen Charakter der Planungsaufgabe ist eine explizite Optimierung der täglichen Produktionsreihenfolge nicht Teil dieser Arbeit. Dies ist und bleibt Aufgabe der Feinplanung, welche die konkreten Ereignisse (Abweichungen von Liefermengen bzw. Liefertag, Maschinenausfall, Krankenstände, etc.) besser und zeitnahe berücksichtigt.

Abbildung 2.8 visualisiert diese zentralen Schlussfolgerungen, wobei der grüne Bereich die Anforderungen hinsichtlich Prozess und Planungshorizont erfüllt. Mit zunehmendem Grad der Variabilisierbarkeit ergibt sich, in Kombination mit einem zumindest mittelfristigen Planungshorizont, eine bestimmte Eignung der in dieser Arbeit entwickelten Methodik.

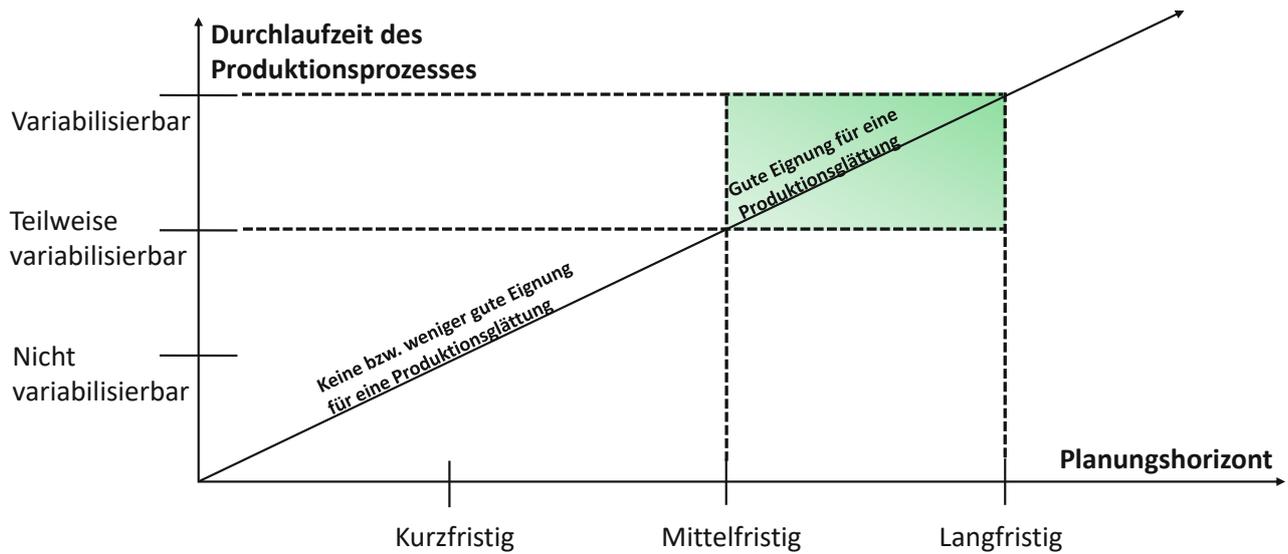


Abb. 2.8: Anforderungen an den Prozess und Planungshorizont (Quelle: eigene Darstellung)

Die methodische Abgrenzung der Produktionsglättungsmethodik gegenüber einer Absatzprognoseglättung ist sinnvoll und notwendig, weil die Glättung der Produktion eine vom Unternehmen selbst beeinflussbare Aufgabe darstellt. Die Optimierung von Prognoseparametern basierend auf der Vergangenheit liefert hingegen keine Sicherheit, dass die auf Basis dieser *Optimierung* ermittelten Parameter auch für die prognostizierte Zukunft ihre Gültigkeit haben. Erwähnenswert bleibt in diesem Zusammenhang jedenfalls, dass die Produktionsplanung notwendigerweise stets auf einer akkuraten Absatzplanung aufsetzt.

In Bezug auf die methodische Abgrenzung zu APS-Systemen wird festgehalten, dass eine auf das jeweilige Optimierungsproblem zugeschnittene Optimierung nicht in einem dieser Systeme ab- oder nachgebildet werden muss, sondern über eine Schnittstelle daran angebunden wird (vgl. Kallrath und Maindl 2006, S. 114). Diverse Einschränkungen seitens der Optimierung sowie die hohen Lizenzkosten, speziell für KMU ohne Nutzung der Netzwerkfunktionalität, lassen den Schluss folgern, dass die zu entwickelnde Methodik sich gegenüber APS-Systemen für die weitere Betrachtung abgrenzen lässt. Traditionelle ERP- und APS-Systeme nehmen basierend auf starren Losgrößen (vgl. Dickersbach 2009, S. 245f.) und festen Produktionskapazitäten die Planung vor. Anstelle von APS-Systemen erscheint es ggf. zweckmäßiger, die dahinter implementierten kommerziellen als auch nicht kommerziellen Optimierungs-Solver (vgl. Kallrath und Maindl 2006, S. 61ff.) zu betrachten und auf deren Vor- und Nachteile einzugehen. Diese Solver lassen im Rahmen der akkuraten Modellerstellung bzw. der Transparenz des Lösungsweges grundsätzlich nicht das ggf. notwendige Feintuning (dynamische Form der Losteilung, saisonal flexibel einstellbare dynamische Kapazitätsgrenzen bzw. anpassbare Melde- und Zielbestandsniveaus, etc.) zu. So lässt sich ein mathematisch ausformuliertes Modell ggf. in eingeschränkter Form mittels eines Solvers lösen, allerdings erlaubt dieser auch nicht die Umsetzung von Phase 6 des ORMA (Implementation als lauffähige Software für die rollierende Planung). Die marktüblichen Solver erlauben dementsprechend keine rollierende Planung eines realen Produktionssystems. Weiters ist in kommerziellen Solvern die Anbindung von integrierter Simulation bzw. statistischer Datenanalyse nicht vorgesehen (vgl. FICO 2021).

3

Kapitel 3

Nichtlineare Optimierungsverfahren: Stand der Technik

Der aktuelle Stand der Technik in dieser Arbeit bezieht sich auf deterministische sowie v.a. stochastische Optimierungsverfahren im Bereich der kombinatorischen Optimierung. Vertreter aus beiden Kategorien sind prinzipiell geeignet, Probleme aus dem Bereich der nichtlinearen multikriteriellen Optimierung im Allgemeinen bzw. das vorliegende MOMPMP Problem aus der Produktionsplanung im speziellen Fall zu lösen. Darüber hinaus werden unterschiedlichste Strategien im Bereich der Hybridisierung, das sind Kombinationen aus unterschiedlichen Verfahren, diskutiert. Zuerst wird ein kurzer Überblick über die verschiedenen Kategorien gegeben, bevor aus jeder Kategorie die für diese Arbeit wichtigsten Vertreter im Detail erläutert und v.a. im Hinblick auf deren Eignung für das Optimierungsproblem dieser Arbeit diskutiert werden. Der Stand der Technik ist v.a. geprägt von zeitgemäßer Literatur auf dem Gebiet der multikriteriellen (nichtlinearen) Optimierung, allen voran von Kalyanmoy Deb, Eckart Zitzler, Kaisa Miettinen, El-Ghazali Talbi¹, Ralph E. Steuer, Joshua Knowles und Carlos Fonseca sowie Matthias Ehrgott (vgl. Branke et al. 2008, Vorwort: S. V-XVIII).

3.1 Überblick

Abbildung 3.1 zeigt die Taxonomie von Optimierungsverfahren. Trotz der Unvollständigkeit dieser Abbildung zeigt sie die wesentlichen Sub-Kategorien und Vertreter der jeweiligen Klassen sowie deren Relevanz im Kontext der Problemstellung. Der Aufbau der weiteren Abschnitte von Kapitel 3 spiegelt sich in der Relevanz der jeweiligen Kategorien. Die klassischen deterministischen numerischen Verfahren werden in dieser Arbeit großteils im Anhang A.2 und A.3 besprochen und nicht näher diskutiert. Präferenz artikulierende Ersatzoptimierungsaufgaben, Heuristiken und populationsbasierte sowie trajektorienbasierte Metaheuristiken werden im Rahmen dieser Arbeit anschließend in Kapitel 3 diskutiert, weil sie im Zuge der Entwicklung der Methodik in dieser Arbeit für die Hybridisierung sowie Verarbeitung der heuristischen Informationen eingesetzt werden. Hybride Strategien erhalten aus Gründen der Übersichtlichkeit eine eigene Taxonomie in ihrem Abschnitt, welcher einerseits die Möglichkeiten der Hybridisierung als auch den Einsatz problemspezifischer Strategien diskutiert.

¹Prof. E. Talbi (Univ. Lille, INRIA, Frankreich) hatte bei der META'2018 den Konferenzvorsitz. Hr. Kamhuber hat dort die im Zuge dieser Arbeit entwickelte Heuristik konzeptionell im Rahmen eines Konferenzbeitrags vorgestellt.

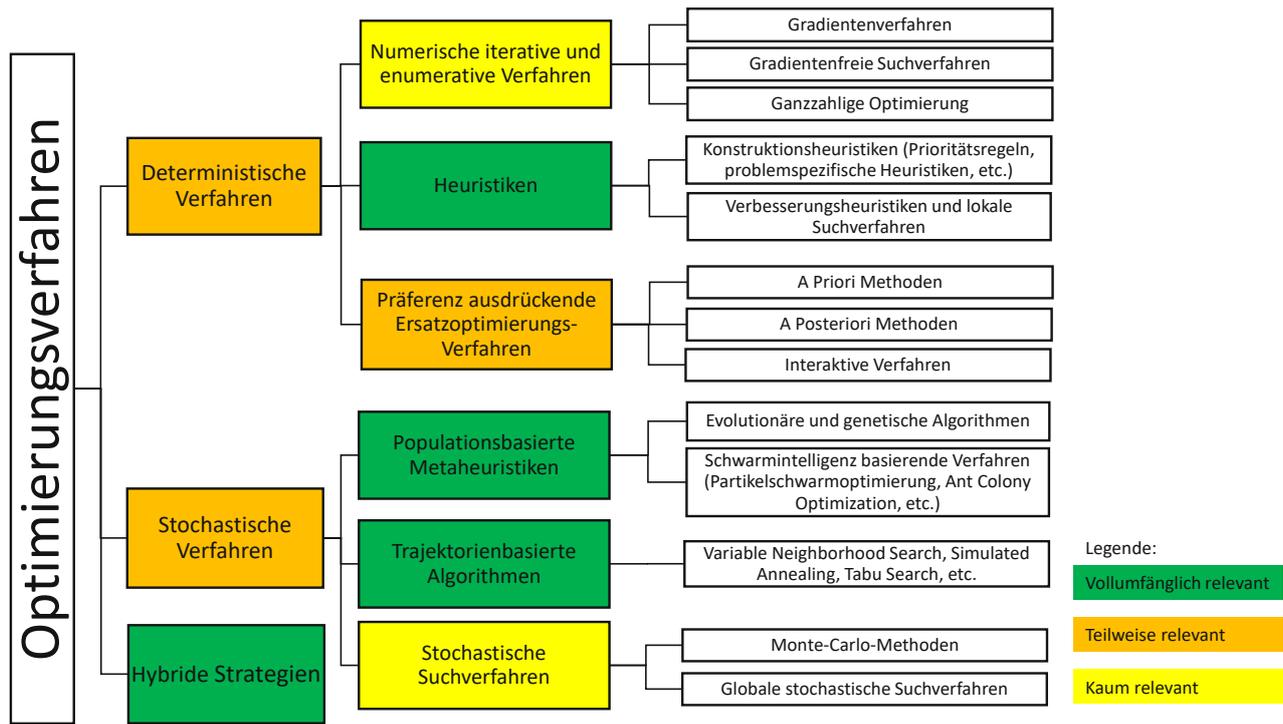


Abb. 3.1: Übersicht: Taxonomie von Optimierungsverfahren (Quelle: eigene Darstellung, in Anlehnung an Kux 2011, S. 15f. und Stöcker 2007, S.14)

Eine weitere Übersicht liefert beispielsweise Meywerk, welcher die Optimierungsverfahren zur Lösung nichtlinearer Probleme in Suchstrategien und Gradientenstrategien gliedert. Die Kategorie der Suchstrategien umfasst dort deterministische Suchstrategien (Jacob-Verfahren, Nelder-Mead-Simplex) sowie stochastische Suchstrategien (Genetische Algorithmen, Simulated Annealing, etc.). Die Gruppe der Gradientenstrategien umfasst Quasi-Newton und Gauß-Newton Verfahren. Diese Übersicht wurde für diese Arbeit nicht herangezogen, weil sowohl die Klasse der Präferenz artikulierenden Ersatzoptimierungsverfahren als auch die Gruppe der Hybriden Strategien fehlen. Meywerk liefert außerdem eine Übersicht in Bezug auf die Einflussfaktoren, welche man bei der Auswahl der jeweiligen Optimierungsstrategie berücksichtigen sollte (vgl. Meywerk 2007, S. 271ff.):

- *Anzahl der Parameter:* Die Anzahl der Parameter entscheidet v.a. darüber, ob Algorithmen verwendet werden, welche in jedem Schritt Informationen, Steigungen oder Krümmungen der Zielfunktion benötigen. Im positiven Falle wird eine große Anzahl an Funktionsauswertungen benötigt. Die Anzahl der Parameter bestimmt außerdem die Konvergenzgeschwindigkeit. Nach Möglichkeit werden Gleichheitsbedingungen nach einem Parameter aufgelöst, um den Suchraum zu reduzieren.
- *Glattheit der Zielfunktion:* Die Glattheit (Unstetigkeit, Stetigkeit oder stetige Differenzierbarkeit) hat Einfluss auf die Auswahl eines Optimierungsverfahrens, da bestimmte Algorithmen, welche z.B. Gradienteninformationen benötigen, nicht anwendbar sind, wenn die Zielfunktion nicht differenzierbar ist.
- *Numerisches Rauschen:* Dieses Kriterium beeinträchtigt Verfahren, welche Gradienteninformationen ausnutzen. Wenn bei der Gradientenbestimmung relativ weit auseinanderliegende Punkte verwendet werden, hat das Rauschen keinen erheblichen Einfluss, sodass die genannten Verfahren verwendet werden.
- *Zeitaufwand/Rechnerressourcen:* Während sich einige Verfahren (GA, EA, etc.) gut parallelisieren lassen, ist dies bei einigen anderen Verfahren erheblich schwieriger (Nelder-Mead-Simplex, etc.). Der Zeitaufwand für eine Zielfunktionsauswertung bestimmt zusammen mit der Parallelisierbarkeit die Antwortzeit für unterschiedliche Verfahren, die jedoch allesamt ohne Simulation auskommen.

Schlüsselwörter und Suchterme im Zuge der Literaturanalyse: Die nächsten Abschnitte widmen sich der Diskussion des relevanten Teils aus dem Stand der Technik in Bezug auf die vorliegende Problemstellung. Die Schlüsselbegriffe im Sinne der systematischen Literaturrecherche und -analyse auf Literaturdatenbanken wie ScienceDirect, Scopus und Google Scholar umfassen dabei betreffend die Primärliteratur gemäß Abb. 3.1 v.a. die folgenden Themenbereiche (vgl. Elsevier B.V. 2021; Elsevier B.V. 2021; Google LLC 2021):

- **Methode:** „*Heuristics*“, „*Metaheuristics*“, „*Simheuristics*“, „*Hybrid Optimization*“, „*Simulation Based Optimization*“, „*Genetic Algorithms*“, „*Simulated Annealing*“.
- **Produktionsplanung, Produktionsglättung:** „*Production Planning*“, „*Production Smoothing*“, „*Production Plan Smoothing*“, „*Optimized Rolling Horizon Planning*“, „*Production Leveling*“, „*Scheduling*“, „*Lot Splitting*“, „*Lot Size Optimization*“.
- **Multikriterielle Optimierung und Nichtlineare Optimierung:** „*Multi-Criteria Optimization*“, „*Multi-Criteria Decision Making*“, „*Multi-Objective Optimization*“, „*Multi-Objective Multi-Period Multi-Product*“, „*Nonlinear Multiobjective Optimization*“.

Auf Grund der kapazitiven Beschränktheit des vorliegenden Problems werden außerdem die Begriffe „*Capacitated*“, „*Compressibility*“ und „*Compressible Process Times*“ bzw. „*Compressible Operation Times*“ ergänzend in der Suche verwendet. Im Zuge der Literaturrecherche bezüglich Präferenz ausdrückender Methoden werden hauptsächlich die Begriffe „*Preference*“, „*A Posteriori Methods*“, „*A Priori Methods*“ sowie „*Interactive Methods*“ verwendet. Im Zuge der erweiterten Recherche für Hybrid Optimization werden auch die Schlüsselwörter „*Simulation*“, „*Hybrid Simulation*“ sowie „*Machine Learning*“, „*Data Mining*“, „*Statistical Learning*“ sowie „*Adaptive*“ bzw. „*Adaptive Learning*“ ergänzend bzw. in Kombination verwendet. Bezüglich der Anwendungsdomäne werden auch die Terme „*Food Industry*“ bzw. „*FMCG*“ oder „*Fast Moving Consumer Goods*“ verwendet.

3.2 Deterministische Verfahren

Die Kategorie der deterministischen Verfahren, welche auch als *Exakte Methoden* bezeichnet werden, verfügen als gemeinsames Merkmal über einen klar definierten algorithmischen Ablauf und benötigen i.d.R. keine stochastischen Einflüsse oder Heuristiken (vgl. Kux 2011, S. 17).

3.2.1 Numerische Verfahren

Die Gruppe der numerischen Verfahren lässt sich weiter untergliedern in die (vollständige) Enumeration sowie Verfahren, welche Gradienteninformationen ausnutzen (Gradientenverfahren, Newton-Verfahren), Gradientenfreie Verfahren, zu denen der Nelder-Mead-Simplex zählt, und Entscheidungsbaumverfahren (vgl. Kux 2011, S. 16).

Gradientenbasierte Verfahren: Die Kategorie der Gradientenverfahren beinhaltet neben dem gleichnamigen Gradienten-Verfahren v.a. das Newton-Verfahren (vgl. Meywerk 2007, S. 282-284). Gradientenstrategien verfügen im Vergleich zu gradientenfreien (Such-) Strategien über eine bessere Konvergenz, die auch mathematisch streng überprüfbar ist. Gradientenverfahren weisen eine höhere Anforderung an die Glattheit der Zielfunktion auf (vgl. Meywerk 2007, S. 277). Der Anhang A.2 beschreibt das Gradientenverfahren inklusive der Bestimmung der dazugehörigen Schrittweiten als auch das Newton-Verfahren. Einige weitere Methoden (Das Verfahren der konjugierten Gradienten, Sequential Quadratic Programming) stehen vor derselben Herausforderung, dass die ersten bzw. teilweise zweiten Ableitungen existieren müssen und effizient bestimmt werden können. Dies stellt für multikriterielle praktische Probleme im Bereich der Produktionsplanung insofern ein Problem dar, weil Gradienteninformationen nicht zur Verfügung stehen (vgl. Kux 2011, S. 17f.).

Diese Gründe zusammen mit weiteren in Unterabschnitt 2.3.2 genannten Herausforderungen sind dafür verantwortlich, dass gradientenbasierte Methoden in dieser Arbeit grundsätzlich nicht in Betracht

gezogen werden, wobei die Idee des Gradienten für die Bestimmung der Glattheit einer Produktion für bestimmte Teilzielfunktionen übernommen werden kann. Diese Gradienteninformationen müssen im Zuge eines globalen Optimierungsalgorithmus bzw. einer hybriden Strategie berücksichtigt werden. Diverse Newton-ähnliche Verfahren, z.B. das Jacob-Suchverfahren, arbeiten auf Basis von Gradienten- und Krümmungsinformationen (vgl. Meywerk 2007, S. 277f.), oder beruhen auf einer Kombination aus einem nicht exakten Newton ähnlichen Verfahren und einer Conditional Gradient (Frank-Wolfe Algorithmus) Methode (vgl. Gonçalves und Oliveira 2018, S. 22-34).

Gradientenfreie Verfahren: Diese verfügen über eine häufig schlechtere Konvergenz verglichen mit Gradientenverfahren, weisen dafür auch sehr geringe Anforderungen an die Glattheit der Zielfunktion auf. Diese Verfahren benötigen keinerlei Gradienteninformationen. Die Methoden aus dieser Kategorie kommen v.a. bei Zielfunktionen mit nicht durchgängig differenzierbaren Ableitungen zum Einsatz. Der *Nelder-Mead-Simplex*, auch unter dem Namen *Downhill-Simplex* oder *Polyeder-Verfahren* bekannt, ist für Probleme mit kleinen Unstetigkeiten oder numerischem Rauschen anwendbar und benötigt eine stetige Zielfunktion. Ein Simplex ist grundsätzlich ein geometrisches Objekt im n -dimensionalen Raum und wird durch $n+1$ Punkte beschrieben. Das im Anhang auf A.3 näher beschriebene *Nelder-Mead-Verfahren* startet mit einem Anfangssimplex und wird schrittweise gemäß bestimmten Konstruktionsprinzipien bzw. Operatoren (*Reflexion, Expansion, Kontraktion*) an seinen Eckpunkten ausgewertet und ersetzt dabei den jeweils schlechtesten Punkt. Der Verfahrensablauf bedingt, dass das Verfahren im n -dimensionalen Raum trotz seiner Möglichkeiten ggf. lokal konvergiert (vgl. Kux 2011, S. 18-20).

Ganzzahlige Optimierung: Das allgemeine Mixed-Integer-Problem (MIP) hat die folgende Form (vgl. Hillier und Lieberman 2015, S. 513):

$$\text{Maximiere } Z = \sum_{j=1}^n c_j x_j, \quad (4.2)$$

$$\text{wobei gilt : } \sum_{j=1}^n a_{ij} x_j \leq b_i \text{ für alle } i = 1, 2, \dots, m,$$

und $x_j \geq 0$ für alle $j = (1, 2, \dots, n)$, wobei x_j ganzzahlig ist für alle $j = 1, 2, \dots, I; I \leq n$.

Für den Fall, dass $n = I$ annimmt, wird das Problem als reines Integer Problem (IP) bezeichnet (vgl. Hillier und Lieberman 2015, S. 513). Zumindest ein Parameter muss ganzzahlige Werte annehmen, damit das Problem als MIP bezeichnet werden kann. Das BaB Prinzip ist in Abb. 3.2 dargestellt.

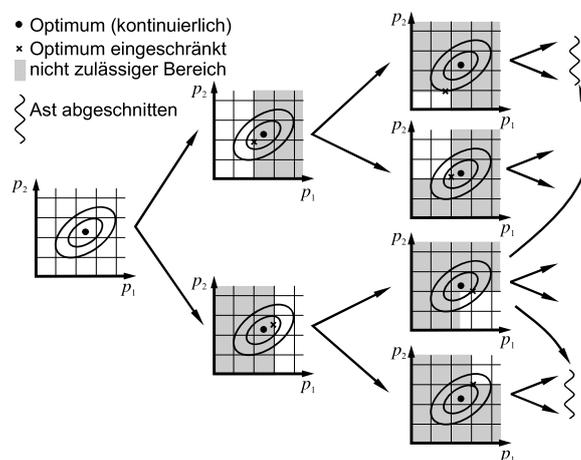


Abb. 3.2: Prinzip des (MIP) Branch-and-Bound-Verfahrens (Quelle: Meywerk 2007, S. 289)

In Abb. 3.2 ist das Optimum des kontinuierlichen Problems durch den schwarzen Punkt skizziert. Falls das Optimum zufällig einen ganzzahligen Wert darstellt, ist das Verfahren bereits nach der ersten Iteration gelöst. Die Zielfunktion hängt allgemein von mehreren Parametern ab, wobei sich Abb. 3.2 auf zwei Parameter (p_1, p_2) beschränkt. Zumeist liegt das Optimum weder auf einem ganzzahligen Wert für p_1 noch für p_2 . Entsprechend wird das Optimierungsproblem in zwei Subprobleme aufgeteilt. Diese Unterprobleme beschränken p_1 in der Form, dass das Optimum nicht mehr im zulässigen (weiß dargestellten) Bereich liegt und, dass die Parameterschranken für p_1 einen ganzzahligen Wert annehmen. Die Idee des Algorithmus besteht darin, die Parametergrenzen auf ganzzahlige Werte zu legen, um so Randextrema zu finden, die auf den diskreten Werten liegen. Im nächsten Schritt wird jedes Suboptimierungsproblem in der Form aufgelöst, dass sich das im zweiten Schritte jeweils ermittelte Optimum im unzulässigen Bereich befindet und die Parametergrenzen für p_2 ganzzahlig sind. Im dritten Schritt wird für die zulässigen weißen Parameterbereiche wieder ein kontinuierliches Optimierungsproblem gelöst. Die Optima werden durch Kreuze dargestellt. Der Algorithmus fährt ggf. mit weiteren Parametern fort (vgl. Meywerk 2007, S. 289-291).

Prinzipiell gibt es zwei Möglichkeiten den Baum aufzuspannen (vgl. Meywerk 2007, S. 290f.):

1. Breitensuche: Ebene für Ebene (in Abb. 3.2 senkrecht dargestellt) behandeln und alle Optimierungsprobleme einer Ebene zu lösen.
2. Tiefensuche: Einen beschrifteten Ast zu Ende gehen, um dann weitere Äste zu betrachten.

Meistens wird nicht der komplette Baum aufgespannt, sondern man konzentriert sich auf günstige Bereiche. Das Aufspannen von Ästen wird als Branch(ing) bezeichnet, das Abschneiden als Bound(ing). Viele Algorithmen aus dem Bereich der ganzzahligen Optimierung basieren auf der Branch-and-Bound (BaB) Idee. Hillier beschreibt einige Beispiele, wo im Zuge der Lösung von Subproblemen teilweise auch keine zulässigen Lösungen auftreten (vgl. Hillier und Lieberman 2015, S. 516ff.).

Diskussion und Fazit: Selbst bei häufig angewendeten BaB Ansätzen ist die Durchführung einer Enumeration nicht ausgeschlossen, wenn die BaB-Regeln für das Problem ungünstig gewählt werden oder in der Problem Instanz nicht effizient sind. Damit ist mit ansteigender Problemgröße eine exponentiell ansteigende Rechenzeit möglich. Im Gegensatz zu polynomial lösbaren Problemen, für welche ein exaktes Lösungsverfahren Sinn ergibt, steigt der Rechenaufwand bei der Lösung NP-schwerer Probleme exponentiell mit der dazugehörigen Problemgröße. Deshalb können exakte Lösungsansätze für praxisrelevante Probleme i.d.R. nicht sinnvoll angewendet werden (vgl. Petri 2006, S. 40f.).

Jedenfalls kann ein zuvor aufgestelltes allgemeines nichtlineares Optimierungsmodell in ein MIP transformiert werden, wenn bestimmte Werte als ganzzahlig beschränkt angenommen werden. Sel und Bilgen stellen eine MIP- und simulationsbasierte Planungsheuristik für die Getränkeindustrie vor, welche die Produktionsmengen an den einzelnen Linien möglichst optimal zuweist (vgl. Sel und Bilgen 2014, S. 385-399). MIP-basierende Heuristiken stellen eine effiziente Möglichkeit zur Lösung großer Problem Instanzen, siehe auch Tab. 2.1, dar (vgl. Absi und Kedad-Sidhoum 2007, S. 171-192). Im Zuge der Kombination einer MIP-basierten (Meta-) Heuristik mit hybrider Simulation kann z.B. der Suchraum anhand der in Testläufen ermittelten Simulationsauflösung entsprechend reduziert werden (vgl. Sobottka et al. 2018, S. 413-420).

Dementsprechend orientiert sich der aktuelle Stand der Technik in konzentrierter Form auf Heuristiken und Metaheuristiken, Präferenz ausdrückende Ersatzoptimierungsverfahren sowie aktuelle hybride Strategien. Für jedes Optimierungsproblem reduziert sich jedenfalls der Suchraum entsprechend der Definition von bestimmten vorab definierten ganzzahligen Werten, welche in Form von Nebenbedingungen formuliert werden können. Dadurch wird das zuvor nicht ganzzahlig definierte Optimierungsproblem in ein MIP transformiert. Dieser Schritt kann jedoch nur in Abstimmung mit dem*der Stakeholder*in erfolgen. Moderne Variablenreduktionstechniken² (vgl. Hua et al. 2008, S. 224-231) oder Ansätze zur Spaltengenerierung³ (vgl. Ramos et al. 2018, S. 155-164) werden, bedingt durch den begrenzten Umfang dieser Arbeit, nicht diskutiert.

²engl.: variable reduction techniques

³column generation (based) approaches

3.2.2 Heuristiken

Heuristische Verfahren werden zur näherungsweise Lösung von komplexen Entscheidungs- und Optimierungsproblemen verwendet. Diese Methoden bedienen sich bestimmter Regeln zur Lösungsfindung, welche das vorliegende mathematisch abgebildete Modell auf Erfolg versprechende Weise ausnützen und einen angemessenen Rechenaufwand erfordern. Während sogenannte *Eröffnungsverfahren* eine zulässige Ausgangslösung konstruieren, gelangen *Verbesserungsverfahren* durch sukzessive Lösungstransformation zu (meist lokal) verbesserten Lösungen. Wohingegen Optimierungsverfahren grundsätzlich eine optimale Lösung eines zugrunde liegenden Modells finden bzw. garantieren (müssen), begnügen sich Heuristiken mit dem Auffinden einer als hinreichend gut bewerteten Lösung (vgl. Domschke und Scholl 2006, S. 1f.), welche auch als *heuristisch effiziente Lösung* bezeichnet wird. Dementsprechend kommen sie v.a. bei Problemen zum Einsatz, wo Optimierungsverfahren auf Grund des zu hohen Rechenaufwands oder Suchraums bei praxisrelevanten Problemen überfordert sind (vgl. Petri 2006, S. 41).

Der Begriff „Heuristik“ entstammt der griechischen Sprache und bedeutet, bezogen auf die Ermittlung einer Lösung, „finden“ bzw. „entdecken“. Koen hat die idealen Charakteristika von Heuristiken wie folgt definiert (vgl. Ansari et al. 2017, S.5):

- Heuristiken garantieren keine optimale Lösung.
- Zwei Heuristiken können einander widersprechen oder eine unterschiedliche Antwort auf dieselbe Problemstellung geben.
- Heuristiken erlauben die näherungsweise Lösung von unlösbaren Problemen bzw. reduzieren sie den Suchraum, um eine zufriedenstellende Lösung zu finden.
- Heuristiken sind stark kontextbezogen: Die Herausforderung besteht darin, in einer gegenwärtig verstandenen Situation mit den verfügbaren Ressourcen eine gute Lösung zu finden.

3.2.2.1 Vorteile von Heuristiken

Die Antwort auf die Frage, wann und warum sich ein*e Analyst*in mit einer Heuristik, also den Vorteilen einer hinreichend guten Lösung, begnügen sollte, gibt Silver (vgl. Silver 2004, S. 937ff.):

1. *Realitätsnahe Abbildung*: Heuristiken benötigen zumeist nicht diverse Einschränkungen von Optimierungsverfahren und sind entsprechend frei in der Modellierung. Dies erlaubt die Verwendung von Modellen, die relevante Probleme aus der realen Welt besser repräsentieren.
2. *Leichte Umsetzbarkeit*: Die Akzeptanz eines*einer Stakeholder*in bezüglich der Verwendung von Entscheidungsregeln, die das Problem auf verständliche Art und Weise lösen, ist mehr gegeben als in Bezug auf eine komplexe Optimierungsstrategie, deren Arbeitsweise nicht verständlich (umgesetzt) ist. Diese Annahme beruht auf dem folgenden Zitat von Woolsey and Swanson: „*People would rather live with a problem they cannot solve than accept a solution they cannot understand*“ (Woolsey und Swanson 1975, S. 68).
3. *Überlegenheit in Bezug auf aktuell verwendete Methoden*: In Bezug auf Punkt 2., Manager*innen begnügen sich bereits mit einer heuristischen Lösung, welche bessere Resultate produziert als die bisherige, verwendete Methode.
4. *Schnelle Resultate*: Oftmals ist es wichtiger, in kurzer Zeit (z.B. wenigen Minuten) gute Resultate zu erhalten als nach stundenlangen Optimierungsroutinen.
5. *Robustheit*: Während optimale Lösungen häufig fragil in Bezug auf Änderungen von Randbedingungen oder den zugrunde liegenden Daten sind, teilen Heuristiken das Problem in mehrere Teile auf und ignorieren auf diese Art und Weise Zusammenhänge zwischen diesen einzelnen Teilen.
6. *Verwendung in Optimierungsroutinen*: Einerseits werden Heuristiken verwendet, um gute Ausgangslösungen zu generieren. Zweitens setzen sie Grenzen, um die Elimination von Teilen des Lösungsraums in Optimierungsverfahren zu erleichtern. Drittens werden sie häufig verwendet, um den Suchprozess gezielt zu steuern.

In Bezug auf Punkt 6 ist anzumerken, dass Heuristiken häufig problemspezifisches Wissen in Metaheuristiken verkörpern und dadurch den Suchprozess gezielt (in eine Richtung) steuern können. Aufschluss darüber wird der Abschnitt Stochastische Verfahren geben.

3.2.2.2 Klassifikation von Heuristiken

Heuristiken werden nach bestimmten Eigenschaften eingeordnet (vgl. Merz 2003, F. 38):

1. *Problembezogen*: Viele Heuristiken sind stark problembezogen und nur für bestimmte Probleme bzw. eine bestimmte Problemklassen anwendbar (z.B. ist der Algorithmus von Johnson in seiner Grundform nur für das 2-Maschinen-Belegungsproblem für n Aufträge in Bezug auf das Kriterium Durchlaufzeitminimierung anwendbar) (vgl. Kamburowski 1997, S. 1).
2. *Komplexität*: Heuristiken werden in Bezug auf ihre Laufzeit und Komplexitätsklasse unterschieden. Grundsätzlich gibt es einfache als auch hybride Heuristiken.
3. *Methodik*: Dabei wird zwischen Konstruktions- und Verbesserungsheuristiken unterschieden. Während Konstruktionsheuristiken bzw. *Eröffnungsverfahren* eine (erste) Lösung generieren, stellen Verbesserungsheuristiken bessere Lösungen zur Verfügung.
4. *Deterministisch versus Stochastisch*: Während deterministische Heuristiken nach vordefinierten Regeln oder Ausführungsschritten ablaufen, arbeiten stochastisch basierte, heuristische Verfahren unter Berücksichtigung von (zumindest einer) Zufallsvariablen. So kann die methodisch und problemspezifisch gleiche Heuristik z.B. sowohl mit einer deterministischen als auch mit einer stochastischen Bearbeitungszeit implementiert sein. Dabei wird durch die mehrmalige wiederholte Ausführung, im Gegensatz zu einer deterministischen Heuristik, eine andere Lösung erreicht.

Ansari klassifiziert Heuristiken einerseits über (a) zunehmendem bzw. (b) abnehmendem Freiheitsgrad, wodurch die Entscheidungen vorheriger Optimierungsschritte entweder revidiert oder schrittweise eingegrenzt werden, sowie über die Ausprägung (c) des Greedy-Ansatzes und (d) prädiktive bzw. vorausschauende Ansätze (z.B. die *Vogel'sche Approximationsmethode*⁴) sowie uninformierte Ansätze, welche die Zielfunktion im Lösungsfindungsprozess nicht berücksichtigen (z.B. die *Nordwest-Ecken-Regel*⁵)(vgl. Ansari et al. 2017, S.5).

Silver stellt fest, dass bestimmte Kriterien den Einsatz einer (Meta-) Heuristik bestimmen: (i) Die Frequenz, mit der eine (ggf. kostenintensive) Entscheidung getroffen wird, (ii) die Abhängigkeit eines strategischen, taktischen oder operativen Entscheidungsbereichs, (iii) die analytischen Qualifikationen des*der Stakeholder*in⁶, (iv) die Problemgröße bzw. Anzahl an Entscheidungsvariablen, (v) die der Entwicklung des Verfahrens erübrigbare Zeitspanne sowie (vi) die Abwesenheit oder der sinnvolle (ergänzende) Einsatz stochastischer Elemente (vgl. Silver 2004, S. 939f.).

Problemspezifische Heuristiken: Vertreter aus dieser Klasse von Heuristiken sind nur für die jeweilige, spezifische Problemklasse anwendbar. Sie zeigen dabei gleichzeitig die individuellen Anwendungsgebiete im Bereich der Produktionsplanung auf:

1. *Maschinenbelegungsprobleme* (Flowshop-Probleme): Im Rahmen des Flowshop-Problems werden n Aufträge auf m unabhängigen Maschinen gefertigt, wobei für jede Maschine die beste Auftragsreihenfolge in Bezug auf das Kriterium Maschinenbelegungszeit oder auch Gesamtdurchlaufzeit ermittelt wird. Während der bereits erwähnte *Johnson-Algorithmus* eine einfache Heuristik für das 2-Maschinen-Belegungsproblem in Bezug auf Durchlaufzeitminimierung darstellt, listet Fernandez-Viagas eine Übersicht an heuristischen Konstruktionsverfahren in Bezug auf den Stand der Technik für die Durchlaufzeitminimierung in Flowshops (vgl. Fernandez-Viagas et al. 2018, S. 345-356; Kamburowski 1997, S. 1).

⁴Idee: Benutzung der Kante, die im Falle des Verzichts die größten Kosten verursachen würde.

⁵Idee: Der Algorithmus, beginnend in der oberen linken Ecke der Kostenmatrix, schreitet sukzessive waagrecht oder senkrecht voran.

⁶siehe auch den nächsten Unterabschnitt 3.2.3 bezüglich Präferenz ausdrückender Methoden

2. *Rundreiseprobleme* (Travelling Salesman Problem, TSP): Das TSP stellt in der Literatur das bekannteste Beispiel für ein NP-hartes Problem dar. Alle Städte (V Knotenpunkte) sind untereinander durch Distanzen (E Kanten) miteinander verbunden und werden während einer Rundreise exakt einmal besucht. Die Gewichtung der einzelnen Kanten berechnet sich über die Distanzen der einzelnen Städte voneinander. Das Ziel besteht in der Ermittlung der Rundreise mit der minimalen Gesamtdistanz. Dieses Problem wird auch durch einen vollständigen Graphen $G = G(V, E)$ beschrieben, wobei ein sogenannter Hamilton'scher Kreis mit dem geringsten Kantensummengewicht gesucht wird. Der Umstand, dass jeder Knotenpunkt exakt einmal besucht wird, liefert die Begründung dafür, dass *Lösungen des TSP auch als Permutationen* bezeichnet werden. Die Ähnlichkeit eines TSP zu einem viel beachteten Problem im Bereich der Produktionsplanung, dem Auftragsreihenfolgeproblem, ist groß: Der einzige strukturelle Unterschied eines Auftragsreihenfolgeproblems gegenüber dem TSP besteht darin, dass der letzte Auftrag nicht dem ersten entspricht, während das TSP eine geschlossene Rundreise beschreibt, wo *Endknoten = Ausgangsknoten* gilt. Bernardino stellt eine Reihe von Heuristiken und ein Iteratives Lokales Suchverfahren (ILS) vor, um eine Generalisierung des TSP, zu lösen (vgl. Bernardino und Paia 2018, S. 453ff.; Michalewicz und Fogel 2004, S. 12-15, 35f.; Kamhuber 2010, S. 19f.).
3. *Auftragsreihenfolgeprobleme* (Permutation Flow Shop Scheduling Problem, PFSSP): Für Auftragsreihenfolgeprobleme gilt, dass jeder Auftrag über dieselbe Maschinenabfolge verfügt. Das gesamte Produktionsprogramm, bestehend aus n Aufträgen, (*jobs*) wird auf m unabhängigen Maschinen produziert, wobei jeder Auftrag an jeder Maschine exakt einmal bearbeitet wird. Dementsprechend ergibt sich, dass für eine idente Maschinenabfolge mit m Maschinen für n Aufträge gemäß dem Prinzip einer Permutation ohne Wiederholung $n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 1$ mögliche Lösungen existieren (vgl. Rossit et al. 2021, S.841f.). Laha und Sarin haben eine auf der von Framinan und Leisten basierende konstruktive Heuristik zur Gesamtdurchlaufzeitminimierung eines PFSP vorgestellt (vgl. Laha und Sarin 2009, S. 734-739; Framinan und Leisten 2003, S. 311-317). Da beide Heuristiken eine, gemäß Landau-Notation, relativ lange Laufzeit von $\mathcal{O}(m \cdot n^4)$ aufweisen, werden sie für diese Arbeit nicht in Betracht gezogen. Die Idee dieser Heuristiken liegt einerseits in der Optimierung von Teilreihenfolgen innerhalb der gesamten Auftragsreihenfolge. Andererseits basiert die Heuristik von Laha und Sarin auf der Anwendung eines iterativen auf Insertion basierendem Verfahren, wobei eine effizientere Nachbarschaft erzeugt wird, als durch den paarweisen Tausch im entsprechenden Schritt bei der Heuristik von Framinan und Leisten (vgl. Laha und Sarin 2009, S. 734-739). Lawrynowicz gibt eine Übersicht über aktuelle Entwicklungen in der Implementierung von GA für Reihenfolgeprobleme (vgl. Ławrynowicz 2011, S. 7-26). Im Zuge des Vergleichs diverser lokaler Suchalgorithmen wurde die Erkenntnis gewonnen, dass auf Insertion basierende Verfahren die Struktur der Reihenfolge weniger stark (zer) stören als der paarweise Tausch von Aufträgen. Dementsprechend wurde auf Basis eines algorithmischen Vergleichs, bestehend aus fünf Metaheuristiken (GA, EA, SA, ACO, ILS), empfohlen, das im Zuge der Arbeit zu behandelnde Auftragsreihenfolgeproblem mittels einer Hybridisierung aus einem GA für die globale Suche und dem IS-Verfahren für die lokale Suche zu lösen (vgl. Kamhuber 2010, S. 117-120).

Greedy-Heuristiken: Greedy (basierte) Algorithmen beruhen auf dem *gierigen* Prinzip der Aneinanderreihung von möglichst optimalen Teillösungen. Das Vorgehen ist dabei *top-bottom* und bedeutet, dass eine Serie von Entscheidungen derart zu treffen ist, sodass jeder Schritt eine optimale Teillösung darstellt und damit im idealen Fall eine möglichst global optimale Lösung in Bezug auf das gesamte Problem erreicht wird (vgl. Pohlmann 2007, S. 242f.). Das Prinzip der Greedy Algorithmen kann anhand von bekannten Vertretern aus der Graphentheorie, beispielsweise dem Dijkstra (vgl. Jungnickel 2010, S. 76f.) bzw. Kruskal-Algorithmus (vgl. Jungnickel 2010, S. 108-111), erklärt werden: In Bezug auf einen Graphen $G = (V, E)$ mit einer dazugehörigen Kantenbewertungsfunktion $w : E \rightarrow \mathbb{R}$ werden die Kanten absteigend (gewichtet) sortiert und konsekutiv betrachtet. Eine Kante (mit minimaler Kantenbewertung) wird nur in den Spannbaum T eingefügt, wenn dadurch kein Kreis mit bereits in T befindlichen Kanten entsteht. Am Ende des Algorithmus bildet T einen Spannbaum mit minimalem

Gewicht bzw. einen minimalen Spannbaum, siehe Algorithmus 3.1. Gemäß Korte und Jungnickel kann der Kruskal-Algorithmus als Spezialfall eines allgemeiner definierten Greedy-Algorithmus definiert werden. Jungnickel definiert den Kruskal-Algorithmus in verallgemeinerter Form auch als Greedy-Prozedur (vgl. Jungnickel 2010, S. 127f.).

Algorithmus 3.1 Der Kruskal Algorithmus (vgl. Korte und Vygen 2012, S. 146)

Input: Ein zusammenhängender ungerichteter Graph G mit $n := |V|$ und $m := |E|$, Gewichte $w : E(G) \rightarrow \mathbb{R}$

Output: Ein Spannbaum T mit minimalem Gewicht

1. Sortiere die Kanten in der Form, sodass $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$
 2. Setze $T := V(G), 0$
 3. **for** $i := 1$ **to** m **do**
 if $T + e_i$ kreisfrei **then** setze $T := T + e_i$
-

Das Greedy-Prinzip besteht in Analogie z.B. bei einer Bedienreihenfolge darin, bei jedem Schritt den Job, der die anderen am wenigsten warten lässt, auszuwählen und in die zu erstellende Liste einzufügen. Dabei entsteht eine Liste mit minimaler Gesamtanwesenheitsdauer aller Kund*innen im System. Auf Grund einer fixen Bedienzeit wird dadurch auch die Gesamtwartzeit minimiert (vgl. Pohlmann 2007, S. 243f.).

Greedy Heuristiken existieren für die verschiedensten Problemstellungen. Korte stellt als einfachste Heuristik für das Bin-Packing-Problem den Next-Fit-Algorithmus bzw. als Verbesserung den sogenannten First-Fit-Algorithmus vor (vgl. Korte und Vygen 2012, S. 506f.).

Lokale Konstruktions- und Verbesserungsheuristiken: Lokale Suche stellt ein algorithmisches Prinzip insofern dar, als diese darauf beruht, eine Lösung mit gewissen *lokalen* Änderungen zu verbessern. So kann eine Rundreise beispielsweise durch das Entfernen zweier Kanten in zwei Stücke geteilt werden und diese anschließend einer anderen Tour hinzugefügt werden. Dabei müssen prinzipiell zwei Entscheidungen vorab getroffen werden (vgl. Korte und Vygen 2012, S. 606ff.):

- Welche Änderungen sind erlaubt bzw. wie wird die Nachbarschaft einer Lösung definiert?
- Wann werden die jeweiligen Änderungen vorgenommen?

Für die weitere Diskussion ist es notwendig, das lokale Minimum zu definieren und anschließend mit dem globalen Minimum zu vergleichen (vgl. Stöcker 2007, S. 7):

Definition 14. Ein Punkt $\mathbf{x}^* \in S$ ist ein *lokales Minimum* der Funktion $f : \mathbb{R}^N \rightarrow \mathbb{R}$, wenn es eine Umgebung U von \mathbf{x}^* gibt, sodass gilt: $f(\mathbf{x}^*) \leq f(\mathbf{x}) \forall \mathbf{x} \in U \subseteq S$.

Das globale Minimum lässt sich wie folgt definieren (vgl. Stöcker 2007, S. 7):

Definition 15. Ein Punkt $\mathbf{x}^* \in S$ ist ein *globales Minimum* der Funktion $f : \mathbb{R}^N \rightarrow \mathbb{R}$, wenn es eine Umgebung U von \mathbf{x}^* gibt, sodass gilt: $f(\mathbf{x}^*) \leq f(\mathbf{x}) \forall \mathbf{x} \in S$.

Dementsprechend stellt Definition 14. den Kern eines jeden lokalen Suchverfahrens dar, siehe auch Zeile 2 aus Algorithmus 3.2. Der Unterschied zwischen einem lokalen Minimum und einem globalen Minimum besteht also darin, dass eine lokale Lösung in Bezug auf die Nachbarschaft U ein lokales Optimum aufweist, welches im Falle eines globalen Optimums gleichzeitig auch ein globales Optimum darstellt.

Algorithmus 3.2 Lokale Suche (vgl. Glover und Kochenberger 2003, S. 221)

Input: Ausgangslösung s

Output: Lokal optimale Lösung s'

```

1: while  $s$  is not locally optimal do
2:   Find  $s' \in U(s)$  with  $f'(s) < f(s)$ 
3:    $s \leftarrow s'$ 
4: end while

```

Abschnitt A.4 im Anhang beschreibt und diskutiert die wichtigsten Vertreter aus dieser Kategorie: Das auf Austausch basierende *K-Opt* Verfahren, das Remove and Insert basierte *Insertion Search* (IS) Verfahren sowie die *Lin-Kernighan Heuristik* (LKH). Außerdem wird an dieser Stelle auf die Konzepte der Tabu-Suche, siehe Anhang A.8.3, und zielgerichteten (und strafbasierten) lokalen Suche (vgl. Glover und Kochenberger 2003, S. 37-54 u. 185-218) verwiesen.

3.2.3 Präferenz ausdrückende Ersatzoptimierungsverfahren

Mathematisch betrachtet werden multikriterielle Optimierungsprobleme durch das Auffinden von Pareto-optimalen Lösungen als gelöst betrachtet (vgl. Miettinen Kaisa 2001, S. 2ff.). Diese entsprechen der sogenannten Pareto-Menge im Designraum bzw. der Pareto-Front im Bildraum. In vielen Fällen reicht das allerdings nicht, weil der*die Entscheidungsträger*in oftmals nur eine finale Lösung erhalten möchte. Dementsprechend müssen die Lösungen entsprechend seiner *Präferenzen* gereiht bzw. sortiert werden. Auf Grund des Vorliegens von einander teils (je nach Problem) konkurrierenden Zielsetzungen muss man sich im Rahmen der multikriteriellen Optimierung mit nichtlinearen Lösungsmethoden beschäftigen, weil ein*e Analyst*in grundsätzlich nicht davon ausgeht, dass ein lineares Problem vorliegt (siehe Definition 2 in Abschnitt 2.3), welches mittels Methoden aus dem Bereich der linearen Optimierung gelöst werden kann (Simplex-Algorithmus, Gauß-Elimination, Ellipsoidmethode, etc.) (vgl. Korte und Vygen 2012, S. 57-105).

Ersatzoptimierungsverfahren werden auch als *Kompromissmodelle* bezeichnet und ersetzen das Zielsystem durch eine skalare Zielfunktion sowie ggf. zusätzliche Nebenbedingungen. Die Präferenzen des*der Entscheidungsträger*in werden dabei einerseits durch die Transformation der Ziele in einen Skalar und andererseits in Form von Nebenbedingungen berücksichtigt (vgl. Miettinen Kaisa 2001, S. 9f.; Petri 2006, S. 30). Die in diesem Teil der Arbeit diskutierten Methoden werden, basierend auf dem Umfang der Mitwirkung des*der Entscheidungsträger*in, in die vier folgenden verschiedenen Kategorien unterteilt (vgl. Miettinen Kaisa 2001, S. 1-13; Branke et al. 2008, S. 14f.): Methoden ohne Präferenz, A-priori-Methoden, A-posteriori-Methoden und interaktive Methoden.

Die Klasse der Methoden ohne Präferenz suggeriert, dass die Meinung des*der Entscheidungsträger*in ignoriert bzw. im Rahmen der Lösungsfindung nicht berücksichtigt wird. Das Problem wird über eine meist einfache Methodik oder Heuristik gelöst und anschließend präsentiert. Der*die Entscheidungsträger*in nimmt die Lösung an oder verwirft sie (vgl. Branke et al. 2008, S. 13). Die entsprechenden Verfahren werden im Anhang (Abschnitt A.6) diskutiert.

A-priori-Methoden stellen Verfahren dar, bei denen die Präferenzen vorab der eigentlichen Optimierung bekannt sind. Dazu muss der*die Entscheidungsträger*in neben einem gewissen Zeitaufwand auch ein bestimmtes Problemverständnis sowie ein (*a priori*) Verständnis über die relative Ordnung bzw. Wichtigkeit (*Ranking*) der Präferenzen, Abhängigkeiten unter den Teilzielen sowie die Gültigkeit von Lösungen mitbringen. Diese Präferenz ausdrückenden Informationen werden anschließend entsprechend modelliert. Es werden Entscheidungen auf Basis von Informationen getroffen, welche der Suche nach Lösungen vorangesetzt sind. Daher sind diese Methoden einfach nachvollziehbar, sehr gut verständlich und werden genau modelliert. Der Vollständigkeit wegen ist an dieser Stelle auch die Zielfunktionsmethode genannt, welche antizipiert, dass (*a priori*) eine explizite mathematische Formulierung gegeben ist, welche die Präferenzen allumfassend spiegelt. Die Optimierung widmet sich

dann der Optimierung des durch den*die Entscheidungsträger*in formulierten Problems inklusive der definierten Nebenbedingungen. Bei der Anwendung von A-posteriori-Verfahren findet die Entscheidung über die präferierte (Pareto-optimale) Lösung erst nach der eigentlichen simultanen Optimierung statt. Dementsprechend werden keine bzw. wenige problemspezifische (Vorab-) Informationen benötigt (vgl. Branke et al. 2008, S. 18f.; Chiandussi et al. 2012, S. 917).

Eine Auflistung und Diskussion weiterer A-priori- bzw. A-posteriori-Verfahren ist dem Beitrag von Marler zu entnehmen (vgl. Marler und Arora 2004, S. 369ff.).

Interaktive Methoden erfordern den kontinuierlichen wechselseitigen Dialog zwischen Entwicklungs- und Anwendungspartner*in. Für interaktive Methoden gilt daher, dass der*die Entscheidungsträger*in, im Gegensatz zu Methoden aus den anderen vorgestellten Kategorien, über keine globale Präferenzstruktur verfügt. Stattdessen kann er seine Präferenzen während der Feedback-Phase ausdrücken bzw. generieren, aktualisieren und überdenken, sobald er das Problem und die dazugehörigen Lösungen besser versteht. Der*die Industriepartner*in benötigt mehr Zeit bzw. Fähigkeiten für die Kooperation, im Gegenzug generieren Methoden aus dieser Kategorie die vielversprechendsten Ergebnisse. Prinzipiell gibt es drei Sub-Kategorien von interaktiven Methoden (vgl. Branke et al. 2008, S. 69-87):

1. Referenzpunkt-basierte Methoden.
2. Trade-Off-basierte Methoden.
3. Auf Klassifikation basierende Methoden.

Zunächst werden die relevantesten A-priori- und A-posteriori-Methoden diskutiert. Abschließend werden die zwei bekanntesten interaktiven Methoden vorgestellt sowie in Bezug auf ihre Praktikabilität für die vorliegende Problemstellung diskutiert. Eine vollumfängliche Übersicht über interaktive Methoden wird in der Übersicht bzw. der Literatur gegeben (vgl. Branke et al. 2008, S. 69-87; Ehrgott und Gandibleux 2003, S. 227-276).

Abschnitt A.6 des Anhangs diskutiert 3 weitere interaktive Methoden an, welche auf Grund ihrer vergleichsweise raren Anwendung sowie Spezifika für die nähere Betrachtung im Rahmen dieser Arbeit nicht relevant sind.

Grundsätzlich besteht der Lösungsfindungsprozess multikriterieller Probleme aus zwei Phasen, welche unterschieden werden (vgl. Chiandussi et al. 2012, S. 918):

1. Die Optimierung der aufgestellten Zielfunktion.
2. Der (multikriterielle Entscheidungsfindungs-) Prozess, welche Kompromisse aus Sicht des*der Entscheidungsträger*in angebracht bzw. notwendig sind.

Phase 2 beschreibt, welche Inhalte problemrelevant sind bzw. durch eine (mathematische) Optimierung gelöst werden. Dabei werden ggf. unterschiedliche Techniken (z.B. Simulation und Optimierung) miteinander kombiniert, wenn bestimmte für die Optimierung relevante Informationen im Zuge einer Simulation bewertet werden (z.B. der spezifische Energieverbrauch von Anlagen, Pufferfüllstände, Zwischenankunftszeiten). Bestimmte Kriterien fallen ggf. weg, weil diese außerhalb der Systemgrenze liegen (z.B. die Verlässlichkeit von Lieferant*innen in Bezug auf Mengen und Zeitfenster) oder erst ab einer bestimmten Granularität sinnvoll betrachtet werden. Zunächst ist die Definition einiger Fachbegriffe erforderlich.

Definition einiger notwendiger Fachbegriffe: Interaktive Lösungsverfahren verwenden vielfach sogenannte *Nutzenfunktionen* (englisch: *value functions*). Dabei nimmt der*die Analyst*in an, dass Unternehmensentscheidungen auf Basis einer zugrunde liegenden Funktion zu treffen sind. Diese Funktion stellt die Präferenzen in Form einer Nutzenfunktion $U : \mathbb{R}^k \rightarrow \mathbb{R}$ dar. Oft ist der Wert bzw. Nutzen dieser Funktion implizit bekannt. Nutzenfunktionen sind wichtig im Lösungsfindungsprozess und als theoretischer Background. Generell sind Nutzenfunktionen stark abnehmend umgesetzt. Das bedeutet, dass die Präferenz der Stakeholder*innen zunimmt, während ein Zielfunktionswert abnimmt und alle anderen Zielfunktionswerte unverändert bleiben (vgl. Mietinen Kaisa 2001, S. 4f.).

Anstelle der Suche nach dem Optimum begnügt sich ein*e Analyst*in mit der Suche nach einer befriedigenden Lösung, welche die gesteckten Aspirationsniveaus erfüllt bzw. übertrifft. Aspirationsniveaus \bar{z}_i stellen gewünschte oder befriedigende Bereiche von Zielfunktionswerten dar, welche deshalb von spezieller Bedeutung für die Stakeholder*innen sind. Der Vektor $\bar{\mathbf{z}} \in \mathbb{R}^k$ wird auch *Referenzpunkt* genannt (vgl. Mietinen Kaisa 2001, S. 8f.).

Die Bestandteile des idealen Zielfunktionsvektors \mathbf{z}^* sind die einzelnen Optima der Teilzielfunktionen. Manchmal wird ein Vektor benötigt, der grundsätzlich besser ist als jede (ideale) Pareto-optimale Lösung. Dieser Vektor wird *utopischer Zielfunktionswert* genannt und mit $\mathbf{z}^{**} \in \mathbb{R}^k$ bezeichnet. Seine Bestandteile werden durch Hinwegnahmen von Komponenten aus \mathbf{z}^* geformt. Die oberen Grenzwerte einer Pareto-optimalen Lösung stellen die Komponenten des Nadir Zielfunktionsvektors, \mathbf{z}^{nad} , dar. Diese sind deutlich schwieriger zu erhalten, nachdem bis heute keine Methode existiert, um diese Vektoren zu berechnen. Dementsprechend wird dieser Vektor bei Bedarf auf Basis der vorhandenen (jeweils besten) Teilzielfunktionsvektoren geschätzt (vgl. Ehrgott und Gandibleux 2003, S. 230; Hellwig 2009, S. 31-33).

3.2.3.1 Die Gewichtete-Summen-Methode (GSM)

Die konventionelle *Gewichtete-Summen-Methode*⁷ transformiert ein multikriterielles Problem mittels einer linearen Kombination von gewichteten Teilzielfunktionen in ein einkriterielles Problem mit einer Zielfunktion. Dies wird üblicherweise durch die (multiplikative) Zuweisung von Gewichtungen (*Skalierung*) für jedes einzelne Teilziel f_i erreicht. Anschließend werden die gewichteten Teilziele additiv oder multiplikativ zu einem einzelnen Zielfunktionswert kombiniert, dem sogenannten gewichteten Fitnesswert. Eine optimale Lösung des einkriteriellen Problems stellt eine Pareto-optimale Lösung für das multikriterielle Problem dar (vgl. Freitas 2004, S. 78f.; Chiandussi et al. 2012, S. 918).

Die Zielfunktion des skalaren Optimierungsproblems wird durch eine gewichtete Summe der einzelnen Teilziele dargestellt und wie folgt definiert (vgl. Chiandussi et al. 2012, S. 918; Stöcker 2007, S.11f.):

$$\min \sum_{i=1}^k w_i \cdot f_i(\mathbf{x}) \quad (3.1)$$

für alle $\mathbf{x} \in S$

mit Gewichtungen $w_i \geq 0$. Auf Grund des häufigen Einsatzes der Methode werden die damit einhergehenden Vor- und Nachteile im nächsten Paragraph diskutiert.

Vor- und Nachteile dieser Methode: Aus den positiven Gewichtungen entsteht der Nachteil, dass im Falle des Vorliegens einer nicht konvexen Funktion, welche z.B. konkav ist, bestimmte Punkte bzw. ganze Abschnitte der Pareto-Front (PF), siehe Punkt C in Abb. 3.3, unabhängig von der Wahl der einzelnen Gewichtungen nicht erreicht werden können. Deshalb wird dieses Verfahren auch als *Konvexkombination* bezeichnet (vgl. Petri 2006, S. 32; Marler und Arora 2010, S. 860).

⁷engl.: weighted sum method, weighting method, linear combination of weights.

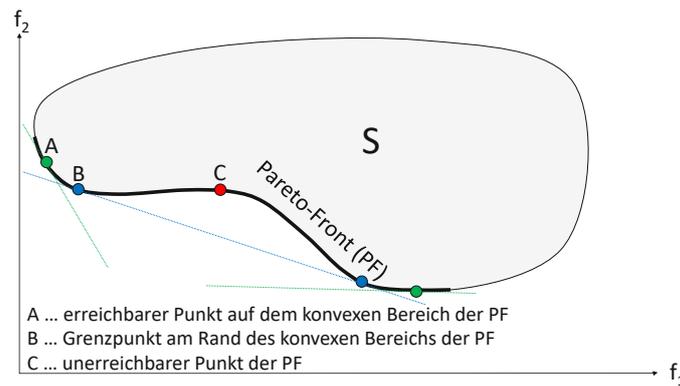


Abb. 3.3: Erreichbare Punkte (A, B) und unerreichbarer Punkt (C) der Pareto-Front (Quelle: eigene Darstellung in Anlehnung an Branke et al. 2008, S. 5)

Wenn alle Gewichtungen gleichwertig vorliegen, reduziert sich die Formel auf die Minimierung von $f_i(\mathbf{x})$. Diese *Nutzenfunktion* aus (3.1) ist einfach zu implementieren und intuitiv. Die Güte der Lösung hängt in entscheidendem Ausmaß von den einzelnen Gewichtungen ab. Diese werden vorab der Ausführung der Optimierung initial definiert. Dieses Rahmenwerk ermöglicht es, alle einzelnen Teilziele, ähnlich zu einer *Nutzwertanalyse*, transparent und vergleichbar für die Entscheidungsträger*innen zu gestalten. Üblicherweise werden mehrere Schleifen benötigt, bis die einzelnen Gewichtungen so definiert sind, dass sie die Präferenzen möglichst realitätsnahe umsetzen bzw. berücksichtigen (vgl. Freitas 2004, S. 79f.; Chiandussi et al. 2012, S. 918f.).

Darüber hinaus ist eine geeignete Skalierung (z.B. in Bezug auf eine Ausgangslösung) erforderlich, welche ein entsprechendes Verständnis für das konkret vorliegende Problem voraussetzt. Die Methodik ist v.a. wegen der raschen bzw. effizienten Potentialabschätzung in Bezug auf die einzelnen Teilziele und der vergleichsweise einfachen Implementierbarkeit vielversprechend. Weil die Methode eine lineare Näherung der Präferenzen der (originalen) Zielfunktion darstellt, bleiben die ursprünglich gesetzten Präferenzen während der Optimierung möglicherweise nicht aufrechterhalten. Die Gesamtlösung hängt außerdem von der relativen Magnitude von den Zielfunktionswerten ab (vgl. Marler und Arora 2010, S. 860).

Dies suggeriert in Kombination, dass beispielsweise ein weniger wichtiges Teilziel mit mehr Potenzial stärker verbessert wird als ein ursprünglich wichtigeres Teilziel mit weniger vorhandenem Optimierungspotenzial. Es wird festgehalten, dass die Optimierung in einem Bereich bleibt, aus dem sie anschließend ohne Hinzunahme weiterer Methoden selbständig nicht wieder herauskommt. Dementsprechend empfiehlt Marler bei der Definition der einzelnen Gewichtungen, dass lediglich die relative Wichtigkeit berücksichtigen werden sollte. Wenn dies nicht möglich ist oder die Gewichtungen als *Trade-Offs* fungieren, empfehlen die Autoren, dass die originalen Einheiten der ursprünglich gesetzten Ziele ohne Veränderungen bzw. Transformationen belassen werden. Die einzige grundsätzliche Restriktion beläuft sich darauf, dass für alle Gewichtungen $w_i \geq 0$ gilt, wodurch die Pareto-Optimalität grundsätzlich ermöglicht wird (vgl. Marler und Arora 2010, S. 860).

Abschnitt A.13 aus dem Anhang diskutiert die Methode anhand eines konkreten Beispiels.

Einordnung dieser Methode: Die GSM wird im Rahmen der Literatur sowohl als A-priori-Methode (vgl. Marler und Arora 2010, S. 854ff.) als auch A-posteriori-Methode gemäß Chiandussi (vgl. Chiandussi et al. 2012, S. 917ff.) oder Miettinen (vgl. Miettinen Kaisa 2001, S. 5ff.) angeführt. Im Zuge der systematischen Alternierung der Gewichtungen wird die Methode zumeist a posteriori verwendet, um die Gewichtungen auf Basis der Präferenzen des*der Entscheidungsträger*in (im Nachgang, also erst im Zuge der Lösungsaufbereitung durch die Optimierung) möglichst gut einzustellen (vgl. Marler und Arora 2010, S. 860f.). Grundsätzlich müssen die Vorlieben bzw. Gewichtungen jedoch zuerst einmal a priori definiert werden, damit die Methode ausgeführt werden kann. Bei Unterlassung der systematischen Alternierung der Gewichtungen spricht man daher auch von einer a priori Artikulation

der Präferenzen. Dies garantiert grundsätzlich noch nicht, dass die dadurch erhaltene finale Lösung auch akzeptiert wird. Daher erfolgt die Einordnung der Methodik in dieser Arbeit als a posteriori.

3.2.3.2 Die Epsilon-Constraint Methode (ECM)

Neben der gewichteten-Summen-Methode stellt die Epsilon-Constraint (auch ε -Constraint) Methode die am besten bekannte (a posteriori) Vorgehensweise dar, um multikriterielle Probleme einer Lösung zuzuführen. Anstelle des Aggregierens von unterschiedlichen Kriterien wird nur eine der definierten Zielsetzungen, p , optimiert, während die anderen $p - 1$ Zielsetzungen zu Nebenbedingungen mit einem fest definierten Grenzwert ε_k (wobei $k = 1, \dots, p, k \neq j$) umgeformt werden. Dies passiert durch Festlegen einer oberen Schranke ε_k , welche nicht überschritten werden darf. Das ursprüngliche Problem der allgemeinen Form (vgl. Chiandussi et al. 2012, S. 919f.; Ehrgott und Gandibleux 2003, S. 77f.; Martin Stöcker 2007, S.12f.):

$$\min (f_1(\mathbf{x}), \dots, f_p(\mathbf{x})) \quad \text{für alle } \mathbf{x} \in S \quad (3.2)$$

wird dadurch wie folgt substituiert (vgl. Chiandussi et al. 2012, S. 919f.):

$$\min f_j(\mathbf{x}) \quad \text{für alle } \mathbf{x} \in S \quad (3.3),$$

$$\text{wobei gilt: } f_k(\mathbf{x}) \leq \varepsilon_k, \quad k = 1, \dots, p, \quad k \neq j.$$

Abb. 3.4 zeigt ein bikriterielles Optimierungsproblem mit einer oberen Schranke für $f_1(\mathbf{x})$. Die optimalen Lösungswerte für $j = 2$ für zwei Werte von ε_1 sind auf der Abbildung ersichtlich. Diese zeigen, dass die Nebenbedingungen für $f_i(x) \leq \varepsilon_i$ gleichzeitig ggf. an einer optimalen Lösung liegen. Die ε_i Koeffizienten stellen zu garantierende Zielfunktionswerte dar. Bei einer alternativen Anordnung der einzelnen Teilziele bleibt deren Souveränität bzw. Identität erhalten, sodass sie nicht miteinander vermischt werden. Im Gegensatz zur GSM ist diese Methode unabhängig von der Konvexität des Problems *zumeist* in der Lage, die komplette Menge an Pareto-optimalen Lösungen zu erzielen. Die erhaltenen Lösungen stellen mit Bestimmtheit nicht-dominierte Lösungen dar, wenn sie einzigartig sind. Die Eigenschaft der einzigartigen Optimalität ist je nach der Modellierung des Problems nicht einfach zu überprüfen, weshalb es möglich ist, dass ein anderes Optimum $\bar{\mathbf{x}}$ existiert, welches im Rahmen der Methode grundsätzlich nicht zugänglich ist (vgl. Chiandussi et al. 2012, S. 919f.; Petri 2006, S. 33).

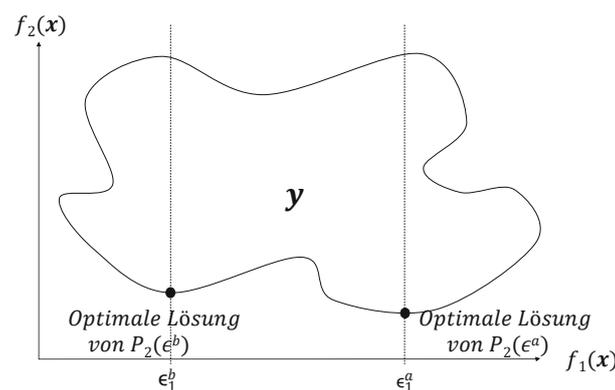


Abb. 3.4: Optimale Lösungen der *Epsilon – Constraint* Methode für ein bikriterielles Problem ($j = 2, \varepsilon_1$) (Quelle: eigene Darstellung, in Anlehnung an Chiandussi et al. 2012, S. 920)

Esteso verwendet diese Methode für die Erzeugung einer implementierbaren Lösung im Rahmen der Profitmaximierung, während die anderen Zielsetzungen (Minimierung von Teillieferungen, Minimierung der Anzahl an Aufträgen mit Verspätung) als Nebenbedingungen formuliert werden. Gleichzeitig wird auch die GSM angewendet, um Szenarien mit definierten Gewichtungen zu lösen (vgl. Esteso et al. 2018, S.180-194).

Vor- und Nachteile dieser Methode: Nachteile dieser Methode stellen der berechnungsintensive Aufwand sowie die Bestimmung individueller Werte für ε_i dar. Im Falle zu kleiner Schranken für ε_i kann beispielsweise keine Lösung gefunden werden. Das Hinzufügen von Nebenbedingungen sowie die veränderte Struktur der Fitnesslandschaft verwandelt ein einfaches in ein schwieriger zu lösendes Problem. Darüber hinaus ist das Konzept für den*die Entscheidungsträger*in nicht mehr einfach nachvollziehbar (vgl. Ehrgott und Gandibleux 2003, S. 288).

Miettinen verweist auf Literatur, die sich damit beschäftigt, diese Grenzwerte systematisch zu zerrütten (*perturbieren*), um unterschiedliche Pareto-optimale Lösungen zu erhalten. Ein weiterer Vorteil der Methode liegt neben der festgestellten Simplität darin, dass Pareto-optimale Lösungen auch für nichtkonvexe Probleme gefunden werden (vgl. Miettinen Kaisa 2001, S. 6).

3.2.3.3 Die Methode der gewichteten Metriken (GM)

Die Methode der gewichteten Metriken *generalisiert* die Idee aus der Global Criterion⁸ Methode: Es wird der Abstand zwischen einem Referenzpunkt und den möglichen Lösungen in der Fitnesslandschaft minimiert (vgl. Branke et al. 2008, S. 15f.). Üblicherweise wird dabei die Nähe einer Lösung (in Form des Zielfunktionsvektors) zu einem idealen Zielfunktionsvektor z^* anhand einer Distanzfunktion, der sogenannten L_p -Metrik (bzw. L_p -Norm oder L_p -Distanz) bewertet. Unterschiedliche Lösungen können durch eine Variation der Gewichtungen der Kenngrößen erreicht werden. Die Methode ist auch unter dem Synonym **Compromise Programming** bekannt (vgl. Branke et al. 2008, S. 16; Jianbang Gan et al. 1995, S. 379f.). Die Lösungen aus dieser Methodik sind schwach Pareto-optimal, werden jedoch Pareto-optimal, wenn z^{**} als Referenzpunkt verwendet wird. Da die L_p -Metrik in den verschiedensten Ausführungen existiert, erläutert der folgende Paragraph die verfügbaren Varianten.

Abstandsfunktionen: Ersatzoptimierungsaufgaben, die auf Abstandsfunktionen beruhen, minimieren den Abstand zu einem Referenzpunkt. Die bekanntesten Maße repräsentieren die *gewichtete L_p -Distanz*, die *ungewichtete L_p -Distanz*, die *gewichtete Tchebycheff-Distanz* (wobei L_p mit $p = \infty \rightarrow L_\infty$) und die *ungewichtete Tchebycheff-Distanz* (vgl. Petri 2006, S. 35-37). Für jeweils 2 Punkte $(x_1^1, x_2^1, \dots, x_n^1)$ und $(x_1^2, x_2^2, \dots, x_n^2)$ wird die Distanz im n -dimensionalen Raum anhand der L_p -Metrik grundsätzlich wie folgt bewertet (vgl. Jianbang Gan et al. 1995, S. 379):

$$L_p = \left(\sum_{j=1}^n | \mathbf{X}_j^1 - \mathbf{X}_j^2 |^p \right)^{1/p} \quad (3.4)$$

Für jeden Parameter $1 \leq p < \infty$ wird eine bestimmte Distanz bestimmt. Für $p = 1$ reduziert sich die beschriebene Gleichung auf den Abstand gemäß dem Satz von Pythagoras und es resultiert eine rechtwinklige Abstandsmessung (*Manhattan-Distanz*) (vgl. Petri 2006, S. 35). Wenn man für $p = 2$ einsetzt, misst man den Abstand auf Basis der euklidischen Distanz und optimiert eine quadratische Funktion. Das Konzept der Distanzen dient grundsätzlich als *Proxy-Maß* für menschliche Präferenzen. Im Rahmen der L_p -Metrik stellt L_1 den größten Abstand zwischen 2 Punkten dar, während L_∞ den geringsten Abstand darstellt (vgl. Jianbang Gan et al. 1995, S. 380). Im allgemeinen Fall hat das Problem, basierend auf der gewichteten L_p -Metrik, die folgende Form (vgl. Branke et al. 2008, S. 16):

$$\min \sum_{i=1}^k w_i | (f_i(\mathbf{x}) - z_i^*) |^p \quad (3.5) \text{ für alle } \mathbf{x} \in S$$

Wenn man für $p = \infty$ wählt, kann der Exponent $1/p$ weggelassen werden, und man spricht von der L_∞ (Tchebycheff-) Metrik oder der gewichteten Tchebycheff-Distanz der Form (vgl. Branke et al. 2008, S. 16):

⁸siehe Anhang A.6.1.1

$$\min \max_{i=1,\dots,k} [w_i | f_i(\mathbf{x}) - z_i^* |] \quad (3.6) \text{ für alle } \mathbf{x} \in S.$$

Wenn alle Teilzielfunktionsbestandteile als gleichwertig betrachtet werden, entfällt der Gewichtungsterm w_i und man erhält die ungewichtete L_p -Distanz, bzw. im Falle von $p = \infty$ die ungewichtete Tchebycheff-Distanz. Die konkrete Lösung hängt dabei von der Wahl des Parameters p ab. Meistens wird für $p = 1, 2$ oder ∞ gewählt (vgl. Miettinen 1998, S. 68). Petri verwendet im Zuge der Entwicklung seiner ACO-Algorithmen z.B. die gewichtete Tchebycheff-Distanz (vgl. Petri 2006, S. 37, S. 174, S. 209).

3.2.3.4 Lexikographische Optimierung (LO)

Die lexikographische Optimierung beruht darauf, dass die einzelnen K Teilziele vorab entsprechend ihrer relativen Wichtigkeit (absteigend sortiert) angeordnet werden. Nach dieser Anordnung wird zuerst das wichtigste Teilziel unter Berücksichtigung der vorliegenden Randbedingungen minimiert. Aus der Menge der Lösungen der Optimierung für das wichtigste Teilziel wird anschließend die Menge von Lösungen bestimmt, die in Bezug auf das zweitwichtigste Teilziel den besten Zielfunktionswert aufweist. Das zweitwichtigste Teilziel wird in Analogie zum wichtigsten minimiert, wobei nun jedoch eine Nebenbedingung in Bezug auf die vorangegangene Optimierung des wichtigsten Teilziels hinzukommt, welche sicherstellt, dass das im Zuge der vorherigen Optimierung gefundene Ergebnis zumindest beibehalten bzw. nicht verschlechtert wird. Das Verfahren wird solange fortgesetzt, bis alle K Teilziele entsprechend optimiert worden sind oder die Menge an Lösungen nur noch aus einem Element besteht (vgl. Branke et al. 2008, S. 19f.; Petri 2006, S. 31).

Die lexikographische Optimierung in Reinform entspricht dem sukzessiven Bestimmen der Mengen aller effizienten Lösungen in Bezug auf alle K Teilziele. In der konkreten Anwendung kann die Methode auch relaxiert werden. Freitas verweist im Zuge seiner Darstellung auf Literatur, wo das Maß an Toleranz über einen sogenannten *Schwellenwert* definiert wird (vgl. Freitas 2004, S. 78f.).

Vor- und Nachteile dieser Methode: Die (finale) Lösung der lexikographischen Optimierung ist Pareto-optimal. Die Methode ist relativ einfach und darüber hinaus werden Entscheidungen oft sukzessiv, also aufeinander in Abhängigkeit folgend, getroffen. Es kann sich schwierig gestalten, für den*die Entscheidungsträger*in vorab der Optimierung die relative Wichtigkeit festzulegen. Darüber hinaus ist es wichtig, dass der Prozess endet, bevor wenig wichtige Teilzielsetzungen im Rahmen der Optimierung betrachtet werden. Obwohl ein Kompromiss bzw. Abtausch von Teilzielsetzungen im Rahmen der Anwendung der Methode grundsätzlich möglich ist, erlaubt die Methode keine Verbesserung eines weniger wichtigen Teilziels zu Lasten einer Verschlechterung eines wichtigeren Teilziels (vgl. Branke et al. 2008, S. 20).

Verglichen mit der bereits vorgestellten Gewichteten-Summen-Methode hat die lexikographische Optimierung mehrere Vorteile (vgl. Freitas 2004, S. 81):

1. Jedes Teilziel wird, nach absteigender relativer Gewichtung, für sich betrachtet und gemäß den Randbedingungen optimiert.
2. Schwierig im Zuge einer Gewichtung vergleichbar darzustellende Kriterien werden nicht miteinander im Zuge eines Fitnesswerts vermischt, sondern unabhängig voneinander betrachtet und verbessert.
3. Nicht intuitive Lösungen, wie beispielsweise in Tabelle A.1 (siehe Anhang Abschnitt A.13) ermittelt, sind bei Anwendung dieser Methode nicht möglich, weil die Optimierung sukzessive auf Basis des jeweils nächstwichtigsten Teilziels erfolgt.

Die Methode ist allerdings nicht zur Bestimmung der Menge aller effizienten Lösungen geeignet, weil nicht alle Elemente aus dieser Menge erzeugt werden können. Durch die unterschiedliche Rangordnung der Ziele können maximal $K!$ unterschiedliche Lösungen generiert werden (vgl. Petri 2006, S. 32).

3.2.3.5 Goal Programming (GP)

Goal Programming ist eine der ältesten Methoden im Bereich multikriterieller Optimierung und stellt eine Verallgemeinerung der Ansätze der Abstandsfunktionen dar (vgl. Petri 2006, S. 39). Dabei wird der*die Entscheidungsträger*in bezüglich des (Mindest-) Anspruchs $\bar{z}_i (i = 1, \dots, k)$ für jedes einzelne Teilziel befragt. Eine Zielfunktion zusammen mit dem dazugehörigen Anspruchsniveau bildet ein Ziel. Minimierungsprobleme weisen Ziele der Form $f_i(\mathbf{x}) \leq \bar{z}_i$ auf. Die Methodik wird in verschiedenen Varianten angewendet. Anschließend werden die Abweichungen von diesen Ansprüchen minimiert. Im Zuge der *gewichteten Variante* wird die gewichtete Summe der Abweichungsvariablen minimiert. Dementsprechend werden auch positive Gewichtungen definiert (vgl. Branke et al. 2008, S. 20). Also liegt ein Problem der folgenden Form vor (vgl. Branke et al. 2008, S. 20f.):

$$\min \sum_{i=1}^k w_i \cdot \delta_i \quad (3.7)$$

wobei gilt: $f_i(\mathbf{x}) - \delta_i \leq \bar{z}_i$ für alle $i = 1, \dots, k$, und alle $\mathbf{x} \in S$,

$$\delta_i \geq 0 \text{ für alle } i = 1, \dots, k.$$

Hierbei stellen $\mathbf{x} \in \mathbb{R}^n$ und $\delta_i (i = 1, \dots, k)$ die Variablen dar, wobei $\delta_i = \max[0, f_i(x) - \bar{z}_i]$.

Im Zuge der *lexikographischen Variante* wird analog zur originalen Methode, siehe 3.2.3.4, mit einer relativen, absteigend sortierten, gewichteten Ordnung in Bezug auf die Ziele und ihre Aspirationsniveaus vorgegangen. Ggf. wird auch die lexikographische und gewichtete Variante kombiniert. Dabei fallen unterschiedliche Teilziele in dieselbe Prioritätsklasse, wobei in jeder dieser Klassen die Summe der Abweichungen minimiert wird. Die Lösung ist Pareto-optimal, wenn entweder die Aspirationsniveaus einen Pareto-optimalen Referenzpunkt bilden oder alle Abweichungsvariablen $\delta_i \geq 0$ über positive Werte beim gefundenen Optimum verfügen. Goal Programming stellt auf Grund seiner Simplität eine weit verbreitete und häufig angewendete Methodik dar, um Entscheidungen zu treffen. Die Herausforderungen liegen in der Spezifikation der Aspirationsniveaus bzw. Gewichtungen oder lexikographisch festgelegten Reihenfolge. Eine Übersicht in Bezug auf verschiedenste Goal-Programming-Ansätze liefert Ehrgott (vgl. Ehrgott und Gandibleux 2003, S. 129-170).

3.2.3.6 Referenzpunkt-Methode (RM)

Gemäß ihrem Namen basiert die Referenzpunkt-Methode auf einem Referenzpunkt \bar{z}^h , welcher auf Vektoren von Aspirationsniveaus basiert oder einem angemessenen Niveau gebildet wird. Dieser Referenzpunkt wird verwendet, um eine sogenannte *Leistungsfunktion* zu erhalten. Diese hat ihre Minima bei (schwach) Pareto-optimalen Punkten. Die Suche wird durch eine kontinuierliche Modifikation des Referenzpunkts stetig verändert, um eine zufriedenstellendere Lösung anzubieten als eine Optimierung einer Nutzenfunktion. Referenzpunkte sind für den*die Entscheidungsträger*in intuitiv und einfach zu spezifizieren, deren Konsistenz ist eine wichtige Anforderung (vgl. Ehrgott und Gandibleux 2003, S. 239f.).

Ein spezifizierter Referenzpunkt entspricht dabei einer Klassifikation der Zielfunktionen. Wenn das Aspirationsniveau unter dem aktuellen Zielfunktionswert liegt, wird diese Lösung nicht akzeptiert. Wenn das Niveau gleich oder über dem Zielfunktionswert liegt, wird die Lösung angenommen. Der Unterschied ist dabei, dass der Referenzpunkt nicht immer realisierbar ist, sodass *Trade-Offs*⁹ unvermeidbar sind, wenn entlang der Pareto-Front optimiert wird. Die grundlegenden Schritte lauten wie folgt (vgl. Ehrgott und Gandibleux 2003, S. 240):

⁹Das sind Kompromisslösungen, bei denen die Verbesserung aus einem Teilziel einer Lösung mit der gleichzeitigen Verschlechterung aus einem anderen Teilziel dieser Lösung einhergeht bzw. mit der Verbesserung aus einem anderen Teilziel bei einer anderen Lösung abgewogen und ggf. abgetauscht wird, wenn die neue Lösung in Summe für besser bzw. zumindest nicht (oder kaum) schlechter bewertet wird.

1. Auswahl einer geeigneten Leistungsfunktion. Problemspezifische Informationen werden dem*der Entscheidungsträger*in präsentiert. Setze $h = 1$.
2. Erfragen eines geeigneten Referenzpunktes $\bar{\mathbf{z}}^h$ bei dem*der Stakeholder*in.
3. Minimierung der Leistungsfunktion. Abgleichen der Lösung \mathbf{z}^h mit dem*der Entscheidungsträger*in.
4. Kalkulation einer Menge an anderen (schwachen) Pareto-optimalen Lösungen durch Minimierung der Leistungsfunktion mit verteilten bzw. *perturbierten* Referenzpunkten $\bar{\mathbf{z}}(i) = \bar{\mathbf{z}}^h + d^h \mathbf{e}^i$, wobei $d^h = \|\bar{\mathbf{z}}^h - \mathbf{z}^h\|$ und \mathbf{e}^i den i -ten Einheitsvektor für $i = 1, \dots, k$ darstellt.
5. Die alternativen Lösungen werden dem*der Entscheidungsträger*in. präsentiert. Bei Zufriedenstellung wird die Optimierung abgebrochen. Sonst wiederholt sich das Prozeder ab Schritt 2, wo ein neuer Referenzpunkt $\bar{\mathbf{z}}^{h+1}$ erfragt wird. Setze $h = h + 1$.

Die Methode beruht auf der Perturbation des Referenzpunkts in Schritt 4, dabei soll dem*der Entscheidungsträger*in eine bessere Vorstellung von möglichen Lösungen um die aktuelle Lösung herum bekommen. Im Zuge dieser Methode muss der*die Entscheidungsträger*in Aspirationsniveaus definieren und Zielfunktionsvektoren untereinander vergleichen. Dabei kann sich die Meinung während dem Lösungsfindungsprozess ggf. ändern und den Lösungsfindungsprozess steuern. Dementsprechend wird keine Konsistenz im Vergleich mit der nächsten vorgestellten Methode gefordert (vgl. Ehrgott und Gandibleux 2003, S. 240f.).

3.2.3.7 Tchebycheff Methode (TM)

Die Tchebycheff Methode stellt ein interaktives, gewichtetes Vektorraumreduktionsverfahren dar, das ohne Nutzenfunktionen auskommt. Die Methodik ist gleichwertig zur Verwendung von gewichteten Koeffizienten, welche durch Referenz-Niveaus angedeutet werden und zählt dementsprechend zur Kategorie der Referenzpunkt-Methoden (vgl. Ehrgott und Gandibleux 2003, S. 236f.; Branke et al. 2008, S. 44).

Zu Beginn wird ein utopischer Zielfunktionsvektor \mathbf{z}^{**} benötigt. Anschließend wird die Distanz von diesem utopischen Vektor zur zulässigen Lösungsmenge, gemessen durch die sogenannte *gewichtete Tchebycheff Metrik* (L_∞) gemäß Glg. (3.4), minimiert. Grundsätzlich kann jede Pareto-optimale Lösung gefunden werden, sofern das gewichtete Tchebycheff Problem mit \mathbf{z}^{**} gelöst wird, wobei einige Lösungen ggf. schwach Pareto-optimal sind. Diese Schwäche wird überwunden, indem das Distanzminimierungsproblem in *lexikographischer Notation* aufgeschrieben und gelöst wird (vgl. Ehrgott und Gandibleux 2003, S. 236):

$$\text{lex minimize } \max_{i=1, \dots, k} [w_i^h (f_i(\mathbf{x}) - z_i^{**})], \sum_{i=1}^k (f_i(\mathbf{x}) - z_i^{**}) \quad \text{für alle } x \in S. \quad (3.8)$$

Wenn das min-max Problem keine eindeutige Lösung liefert, wird der Summenterm in Bezug auf die erhaltenen Punkte minimiert. Pareto-optimale Lösungen werden mittels der TM durch die Veränderung der Gewichtsvektoren \mathbf{w}^h erhalten. Im Rahmen der ersten Iteration wird ein Sample an Pareto-optimalen Lösungen mittels (3.8) generiert. Der vektorielle Lösungsraum W^h wird bei jeder Iteration h durch die Einengung der oberen und unteren Schranken für die Gewichtungen reduziert. Dann wird eine konzentrierte Gruppe von Gewichtsvektoren um den ausgewählten Lösungsvektor gebildet. Der Kern der Idee beruht darauf, eine Sequenz von sukzessive kleineren Teilmengen der Pareto-optimalen Lösungsmenge auszuwählen, bis die finale Lösung gefunden worden ist. Die Anzahl an alternativen Zielfunktionsvektoren wird dem*der Entscheidungsträger*in präsentiert und als fest oder variabel definierte Menge (*Set*) P bezeichnet. Die Hauptschritte des *Tchebycheff-Algorithmus* umfassen dementsprechend (vgl. Ehrgott und Gandibleux 2003, S. 236f.):

1. Definition der Menge an Werten für P sowie Bestimmung der Anzahl an Interaktionen H . Konstruktion des utopischen Zielfunktionsvektors \mathbf{z}^{**} . Setze $h = 1$.

2. Bildung des gewichteten Vektorraums W^h und Erzeugung von $2P$ verteilten Gewichtsvektoren $\mathbf{w}^h \in W^h$.
3. Lösung des Distanzminimierungsproblems in lexikographischer Form (3.8) für jeden von den $2P$ verteilten Gewichtsvektoren.
4. Präsentation der P Vektoren, die am meisten voneinander unterschiedlich sind, und Auswahl der am meisten zu präferierenden Lösung durch den*die Entscheidungsträger*in.
5. Falls $h = H$, Abbruch. Andernfalls werden Informationen gesammelt, um den gewichteten Vektorraum W^h zu reduzieren, damit W^{h+1} erhalten wird. Setze $h = h + 1$ und gehe zu Schritt 2.

Vor- und Nachteile dieser Methode: Im Zuge der Tchebycheff Methode muss der*die Entscheidungsträger*in nur einige Pareto-optimale Lösungen miteinander vergleichend bewerten und die am meisten zu präferierende davon auszuwählen. Deshalb gilt sie, in Relation zu anderen interaktiven Methoden, als relativ *anwenderfreundlich*. Die Leichtigkeit dieses Vergleichs hängt von P sowie der Anzahl an Teilzielfunktionen ab. Die Flexibilität der Methode wird jedoch dadurch reduziert, dass verworfene Teile des gewichteten Vektorsuchraums während dem Suchprozess nicht wiederhergestellt werden können, wenn der*die Entscheidungsträger*in seine*ihre Meinung ändert. Daher ist Konsistenz gefordert. Ein weiterer Nachteil resultiert darin, dass der Berechnungsaufwand bei dieser Methode in jeder Iteration sehr aufwändig ist und die meisten Resultate verworfen werden. Dementsprechend ist diese Methode für große und komplexe Probleme nur eingeschränkt empfehlenswert. Die Tchebycheff Methode benötigt keine Nutzenfunktion und vergleichsweise wenige Informationen von dem*der Entscheidungsträger*in. Die Pareto-Optimalität (PO) wird garantiert und prinzipiell kann auch jede Pareto-optimale Lösung gefunden werden (vgl. Ehrgott und Gandibleux 2003, S. 237). Einen illustrativen Anwendungsfall dieser Methodik liefert der Beitrag von Karaivanova (vgl. Karaivanova et al. 1993, S. 347).

3.2.3.8 Bewertung und Diskussion

Keine der im Rahmen dieses Abschnitts vorgestellten Methoden ist grundsätzlich den anderen Methoden gemäß Ehrgott überlegen (vgl. Ehrgott und Gandibleux 2003, S. 253). Mietinen ist der Meinung, dass man eine geeignete Methode basierend auf den folgenden beiden Kriterien auswählen sollte (vgl. Mietinen Kaisa 2001, S. 18f.):

1. Die spezifischen Eigenschaften des Problems müssen berücksichtigt werden.
2. Die Meinung des*der Entscheidungsträger*in ist von besonderer Bedeutung.

Während der erste Punkt aussagt, dass die Problem-Spezifika abgebildet bzw. modelliert werden, reflektiert der zweite Punkt das Ausmaß des Kooperationsverhaltens des*der Anwendungspartners*in bezüglich der Auswahl einer (z.B. interaktiven) Methode. Prinzipiell erscheint es im Zuge des Lösungsfindungsprozesses sinnvoll, zwei Methoden miteinander derart zu kombinieren, sodass die jeweiligen Schwächen der einen Methode durch die andere ausgeglichen werden. Für die weitere Diskussion werden Methoden ohne Präferenz nicht betrachtet, weil eine grundlegende Präferenz von Seiten des*der Anwendungspartner*in vorliegen soll, wenn ein Problem einer systematischen Lösung zugeführt wird. Die Tchebycheff Methode ist deutlich anwendungsfreundlicher, weil der*die Anwendungspartner*in lediglich Pareto-optimale Lösungen miteinander vergleicht und dabei ohne die Erfragung relaxierter Schranken auskommt (vgl. Branke et al. 2008, S. 22f.).

Die nachfolgende Tabelle 3.1 zeigt die Eigenschaften der verbleibenden grundsätzlich zu favorisierenden Präferenz-Methoden auf. Diese verdeutlicht, dass verschiedene Möglichkeiten existieren, Präferenzen zu formulieren. Die Notation X bedeutet, dass eine Eigenschaft auf eine Methode voll zutrifft, während (X) bedeutet, dass eine Eigenschaft manchmal oder teilweise zutrifft (vgl. Branke et al. 2008, S. 21f.).

Goal-Programming kann auf lexikographische Art und Weise als auch in gewichteter Form angewendet werden. Auf Basis der in dieser Tabelle zusammengefassten Methoden erscheint es sinnvoll, sich für eine Applikation nicht auf eine Methode zu beschränken, sondern die Vorteile von zwei Präferenz ausdrückenden Methoden zu kombinieren. Das Kriterium Pareto-Optimalität bedeutet, dass

die vorgestellte Methode praktisch jede Pareto-optimale Lösung finden kann. Dies gilt z.B. für die Gewichtete-Summen-Methode speziell für konkave Probleme (siehe Abb. 3.3) nicht, weil bestimmte Bereiche der Pareto-Front nicht erreichbar sind oder Einschränkungen auf Basis einer vorab definierten Gewichtung bestehen. Dementsprechend besteht eine vielversprechende Option daraus, zwei Methoden miteinander zu kombinieren, z.B. die Gewichtete-Summen-Methode mit der ε – *Constraint* Methode oder der lexikographischen Optimierung (vgl. Branke et al. 2008, S. 22).

Tab. 3.1: Gegenüberstellung wichtiger Eigenschaften der diskutierten Präferenz-Methoden (vgl. Branke et al. 2008, S. 22)

	GSM	ECM	GM	GP	LO	TM	RM
Interaktive Methode						X	X
A-priori-Methode	X	X		X	X		
A-posteriori-Methode	X	X	X				
Pareto-Optimalität (PO)		X	(X)	X		X	(X)
Garantie für PO Lösung	(X)	(X)	(X)		X	X	X
Präferenz: Gewichtungen	X			(X)			
Präferenz: Schranken		X					
Präferenz: Referenzpunkt				X		X	X
Präferenz: Lexikographisch				(X)	X		

Miettinen und Hader halten für die Zukunft außerdem fest, dass die Benutzerfreundlichkeit deutlich mehr im Vordergrund steht als früher (vgl. Hader 2000, S. 10). Der Algorithmus selbst muss adaptiv in Bezug auf die Berücksichtigung der einzelnen Charakteristika ausgestaltet werden. Der*die Anwendungspartner*in möchte während des Lösungsfindungsprozesses über Kontrolle verfügen und verstehen, was im Ablauf genau passiert (vgl. Miettinen Kaisa 2001, S. 18-19).

Dies liefert, in Ermangelung verfügbarer Referenzpunkte im konkreten Anwendungsfall, den Hauptgrund für die abgewandelte Verwendung der Lexikographischen Optimierung in Kombination mit der Gewichteten Summenmethode. Für die lexikographische Umsetzung der heuristischen Optimierung spricht die hierarchische Vorgabe, welche konkret definierte Zielsetzungen priorisiert.

3.2.4 Fazit

Der Abschnitt über deterministische Verfahren erlaubt die Schlussfolgerung, dass eine laufzeiteffizient implementierte, heuristische Ausgangslösung die Erwartungen bzw. Präferenzen des*der Entscheidungsträger*in entsprechend gut abbildet. Dabei wird die als wichtigste definiert und entsprechend gereichte Zielstellung bestmöglich bedient. Daher wird eine problemspezifisch angepasste, deterministische Heuristik auf Basis der lexikographischen Optimierung zumindest für das wichtigste Teilziel vorgeschlagen und auf weitere Teilziele ausgedehnt, wenn eine heuristisch lexikographische Optimierung dafür umsetzbar ist¹⁰. Die lexikographische Optimierung wird gewählt, weil in dem Fall der vorliegenden Arbeit eine a priori Beeinflussung durch die Entscheidungsträger*innen bezüglich der gewünschten Zielsetzungen ausdrücklich vorliegt. Die Anwendung der lexikographischen Ordnung in relaxierter Form erlaubt einen weniger stark beschränkten Suchraum sowie die Entwicklung von Strategien, die im Rahmen einer strengen Ordnung ohne Toleranzen nicht realisierbar sind. Die Relaxation besteht in der Zulassung und vollständigen Anerkennung von spezifischen Strategien, die gemäß einer bestimmten Ablaufreihenfolge innerhalb der Heuristik v.a. einem zur Laufzeit in Bearbeitung befindlichen Teilziel dienen und eine Verschlechterung von einem (gemäß Ordnung) wichtigeren Teilziel zugelassen wird.

¹⁰Es gibt Teilziele, für die sich auf Grund ihrer Widersprüchlichkeit mit anderen Teilzielen oder ihres Charakters nur schwierig regelbasierte, lexikographische Heuristiken definieren und umsetzen lassen.

3.3 Stochastische Verfahren: Populationsbasierte Metaheuristiken und trajektorienbasierte Algorithmen

Populationsbasierte Metaheuristiken und trajektorienbasierten Algorithmen repräsentieren die Kategorie der stochastischen Optimierungsverfahren in Abb. 3.1. Diese naturinspirierten Verfahren orientieren sich grundsätzlich an den Phänomenen der Natur, z.B. am Abkühlungsprozess von geschmolzenen Festkörpern oder um Modelle, welche die Evolution des Lebens simulieren. Der sogenannte *Lösungsmodus* wird oftmals anhand einer der bereits vorgestellten Ersatzoptimierungsverfahren bedarfsgerecht für die Entscheidungsfindung ausgewählt. Im Zuge der Anwendung ist eine Kombination mit einem *stochastischen* Optimierungsverfahren grundsätzlich empfehlenswert. Die Art und Auswahl der zufälligen Änderungen wird durch das gewählte Verfahren bestimmt. Während die Zufallszahlen im Rahmen der Monte-Carlo-Methode zufällig (auf Basis eines Zufallszahlengenerators) bestimmt werden, bilden Metaheuristiken grundsätzlich (teils physikalische) Vorgänge bzw. Verhaltensweisen aus der Natur ab. Die Art des Verfahrens wird in Bezug auf den Einsatz (als z.B. alleiniges globales Verfahren oder lokales Verfahren zur Hybridisierung, etc.) entsprechend ausgewählt. Das erreichte globale Optimum ist zumeist weder gesichert noch beweisbar. Auf Grund der breiten Anwendungsgebiete (stochastische Optimierungsverfahren, Simulation, Künstliche Intelligenz, etc.) von Zufallszahlen beschäftigt sich der Anhang (Abschnitt A.11) mit deren Verhalten und Struktur sowie deren Erzeugung. Das Monte-Carlo-Verfahren repräsentiert den einfachsten Vertreter von stochastischen Optimierungsverfahren und wird im Anhang, siehe A.7, diskutiert (vgl. Kux 2011, S. 28; Schröder 2010, S. 387).

3.3.1 Grundlegende Definitionen und Klassifikation von Metaheuristiken

Metaheuristiken werden v.a. für kombinatorische Optimierungsprobleme angewendet, für welche auf Grundlage der NP-Vollständigkeit kein effizienter (polynomieller) Algorithmus existiert (vgl. Ansari et al. 2017, S. 6). Typische Probleme stellen Rundreiseprobleme, Reihenfolgeprobleme und quadratische Zuordnungsprobleme dar. Der Fachbegriff wurde u.a. von Glover, der im Rahmen seines Handbuchs über Metaheuristiken, neben dem von ihm entwickelten Tabu Search Verfahren, einen Überblick über die verschiedensten Vertreter der Kategorie aus den stochastischen Optimierungsverfahren gibt, wie folgt definiert (vgl. Glover und Kochenberger 2003, Vorwort):

Definition 16.1. „*Metaheuristics, in their original definition, are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space.*“ (Glover und Kochenberger 2003, Vorwort: S. XI)

Definition 16.2. „*A metaheuristic 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. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method.*“ (Imam et al. 1999, S. 11)

Diese beiden Definitionen verinnerlichen das Prinzip, dass diese iterativ ausgeführten Lösungsmethoden Interaktionen aus einerseits lokalen Verbesserungen sowie einer dazugehörigen globalen Suchstrategie auf einer höheren Ebene kombinieren. Die Definition von Imam betrachtet die Lösungsmethode als Masterprozess, welcher die untergeordneten heuristischen Routinen koordiniert, um qualitativ hochwertige Lösungen zu generieren. Dadurch ist es möglich, lokalen Minima zu entkommen (Imam et al. 1999, S. 11). Ansari ergänzt, dass sich Metaheuristiken durch ihre Unabhängigkeit in Bezug auf ein bestimmtes (kombinatorisches) Problem bzw. Rahmensystem (engl.: *framework*), im Gegensatz zu problemspezifischen Heuristiken, für prinzipiell jedes Problem angewendet werden können und als fundamentales Rahmengerüst universal einsetzbar sind (vgl. Ansari et al. 2017, S. 6). Für metaheuristische Optimierungsverfahren existiert ein fundamentales Theorem:

Definition 17. No-Free-Lunch-Theorem: *Das No-Free-Lunch-Theorem besagt, dass alle (Blackbox-) Optimierungsalgorithmen dieselbe durchschnittliche Leistung in Bezug auf die vollständige Menge*

an Optimierungsproblemen haben. Das verhindert nicht, dass bestimmte Algorithmen für die Lösung bestimmter Probleme bzw. Problemklassen besser geeignet sind (vgl. Boussaïd et al. 2013, S. 111).

Daraus kann geschlossen werden, dass keine Metaheuristik einer anderen *prinzipiell* überlegen ist. Das Theorem basiert auf vielen theoretischen Studien in Bezug auf die Topologie von Fitnesslandschaften und hat gezeigt, dass nicht nur unterschiedliche Probleme eine unterschiedliche Struktur, sondern auch unterschiedliche Instanzen desselben Problems unterschiedliche Strukturen, aufweisen. Also stellt das Verständnis für die einem Problem zugrunde liegende Struktur den ersten Schritt in Bezug auf eine besser abgestimmte Metaheuristik dar. Dies mündet in einem verbesserten algorithmischen Design der entsprechenden (modifizierenden) Operatoren bzw. (heuristischen) Suchstrategien. Die wachsende Forschung in diesem Bereich zeigt die Trends im Bereich der Hybriden Optimierung auf, welche im Zuge dieser Arbeit in Abschnitt 3.4 diskutiert werden (vgl. Boussaïd et al. 2013, S. 111).

Klassifikation von Metaheuristiken Metaheuristiken werden gemäß Gass und Fu in die folgenden vier Subkategorien eingeteilt (vgl. Ansari et al. 2017, S. 6):

1. *Konstruktive Metaheuristiken*: Vertreter aus dieser Klasse von Algorithmen konstruieren sich ihre Lösung (häufig nach einem Greedy Prinzip) schrittweise. Im Gegensatz zu Konstruktionsheuristiken verwenden sie eine Gedächtnisfunktion bzw. einen Speicher und arbeiten *stochastisch*. Beispiele dafür sind Greedy Algorithmen und die von Dorigo bekannten ameisenkoloniebasierten Algorithmen (vgl. Dorigo und Stützle 2004, S. 25-63).
2. *Verbesserungsmetaheuristiken*: Vertreter aus dieser Kategorie verbessern eine bestehende Ausgangslösung.
 - a) *Einzellösungsbasierte Metaheuristiken*: Algorithmen aus dieser Klasse transformieren eine bekannte Lösung sukzessive in eine bessere Lösung und verwenden dafür ein internes Gedächtnis oder eine bestimmte Funktion. Bekannte Beispiele daraus sind Simulated Annealing, Tabu Search bzw. Variable Neighborhood Search.
 - b) *Populationsbasierte Metaheuristiken*: Vertreter dieser Klasse treffen eine Auswahl und Rekombination von bestehenden Lösungen aus einer Lösungsmenge, welche als Population bezeichnet wird, um neue Lösungen hervorzubringen. Die neuen Lösungen werden zufällig mittels sogenannter Mutationen verändert und zur neuen Population hinzugefügt. Typische Vertreter dafür sind Genetische und Evolutionäre Algorithmen.
3. *Hybride Metaheuristiken*: Diese bilden Kombinationen der Kategorien 1 und 2.
4. *Matheuristiken*: stellen Kombinationen aus Metaheuristiken und mathematischen (exakten) Optimierungsverfahren dar, wobei mathematisch exakte Verfahren v.a. für Teilprobleme oder kleine Instanzen eines Problem es herangezogen werden (vgl. Caumont et al. 2009, S. 706-722).

Im Zuge dieser Arbeit werden in Kapitel 3 v.a. Genetische Algorithmen auf Grund ihrer Popularität und Stabilität in Anwendungen diskutiert. Populationsbasierte Verfahren eignen sich eher für die Suche in großen Lösungsräumen und weisen mehr Kontrolle in Bezug auf den Suchprozess auf (vgl. Wari und Zhu 2016, S. 336). Wari hält im Rahmen einer tabellarischen Aufstellung fest, dass in 68 von 116 wissenschaftlichen Beiträgen ein Genetischer Algorithmus in Bezug auf die Lebensmittelindustrie zur Anwendung kommt, während die anderen in seiner Tabelle gelisteten Metaheuristiken (SA, TS, ACO, PSO, etc.) entsprechend seltener angewendet werden (vgl. Wari und Zhu 2016, S. 329-331).

Die bekannteste einzellösungsbasierte Metaheuristik, Simulated Annealing, wird v.a. für größere Lösungsräume eingesetzt. Wari erwähnt, dass trajektorienbasierte Verfahren (SA, TS) mit zunehmender Problemgröße ggf. ineffizient werden und populationsbasierte Verfahren dann geeigneter sind (vgl. Wari und Zhu 2016, S. 338). Das bekannteste konstruktive Verfahren, ACO, sowie weitere Metaheuristiken (Partikelschwarmoptimierung, Tabu Search, Variable Neighborhood Search, Iterated Local Search) werden zur Vervollständigung des Stands der Technik im Anhang in Abschnitt A.8 diskutiert. Eine umfangreiche Verfahrensübersicht liefert Boussaïd. Er teilt Metaheuristiken in einzellösungsbasierte Verfahren, populationsbasierte genetische und evolutionäre Verfahren sowie auf Schwarmintelligenz basierende Verfahren ein, zu denen er ACO zählt (vgl. Boussaïd et al. 2013, S. 82-117).

3.3.2 Genetische und Evolutionäre Algorithmen (GA, EA)

Die Idee der genetischen Algorithmen (GA) wurde 1975 von Holland vorgestellt und seither stetig weiterentwickelt, allen voran von Goldberg und Michalewicz (vgl. Goldberg 2012; Holland 2010; Michalewicz 1999). Das fundamentale einem GA zugrundeliegende heuristische Prinzip stammt aus der Biologie bzw. Evolutionstheorie und wird als „*survival of the fittest*“ bezeichnet (vgl. Rozenberg et al. 2012, S. 626-630).

Eine *Population* besteht aus einer (i.d.R. fest) definierten Anzahl von Lösungen, sogenannten *Individuen* bzw. *Chromosomen* (kodiert in Form von *Genen*¹¹), für das vorliegende Problem. Gemäß den Prinzipien der Natur wird jedes Individuum anhand seiner sogenannten *Fitness* bewertet. Diese Bewertung, mit welcher die Qualität einer Lösung assoziiert wird, erfolgt im Rahmen einer Bewertungsfunktion, welche die Charakteristika des jeweils konkret behandelten Optimierungsproblems widerspiegelt. Genetische Algorithmen unterscheiden sich sowohl auf Grund der Ausprägung ihrer drei verschiedenen, zur Anwendung kommenden *genetischen Operatoren* (Selektion, Rekombination und Mutation), wobei die Individuen nach bestimmten Regeln ausgewählt und dynamisch verändert werden, als auch durch ihre populationsbasierte, multidirektionale Suche in der Fitnesslandschaft. Im Zuge der Selektion ergibt sich ein bestimmter *Selektionsdruck*, der das Konvergenzverhalten steuert (vgl. Rozenberg et al. 2012, S. 626-635, S. 1042f.; Feltl 2003, S. 28f.; Boussaïd et al. 2013, S. 91; Whitley und Sutton 2012, S. 638f.).

Genetische Algorithmen zeichnen sich durch die folgenden Vorteile aus (vgl. Sih et al. 2018, S. 447-450; Wari und Zhu 2016, S. 329-331; Kamhuber 2010, S. 26; Feltl 2003, S. 28; Imam et al. 1999, S. 11):

- Robuste Lösungen (bedingt durch eine globale Sichtweise) in vernünftiger Zeit.
- Keine grundsätzlichen Einschränkungen bezüglich der Bewertungsfunktion für die Optimierung (z.B. Stetigkeit, Ableitbarkeit, Dimensionalität) ermöglichen den universellen Einsatz.
- Eignung für Probleme mit großem und komplexem Suchraum, wo Enumeration nicht sinnvoll möglich ist und zudem wenig weitere, problemspezifische Informationen vorhanden sind.
- Bessere Resistenz der Metaheuristik ggü. lokalen Minima durch den populationsbasierten Ansatz.
- Leichte Verständlichkeit sowie breite Anwendbarkeit für verschiedenste Problemstellungen.

Als Nachteile eines GA wird angeführt, dass auf Grund des populationsbasierten Ansatzes, eine relativ hohe Anzahl an Funktionsbewertungen notwendig ist, sowie je nach Parametrisierung bzw. zu starkem Selektionsdruck das Verfahren frühzeitig zu einem lokalen Minimum konvergiert (vgl. Wari und Zhu 2016, S. 338; Rozenberg et al. 2012, S. 629).

Aufbau und prinzipielle Funktionsweise eines genetischen Algorithmus: Genetische Algorithmen stellen gerichtete Zufallsverfahren mit probabilistischen Entscheidungen dar, die v.a. auf den Operatoren in Bezug auf ihre Selektion, Rekombination und Mutation beruhen. Algorithmus 3.3 zeigt den Aufbau und die prinzipielle Funktionsweise eines GA: Zuerst wird eine *Ausgangspopulation* $P(t)$ initialisiert, welche anschließend bewertet wird. Eine Population besteht aus n Individuen, wobei $n \geq 1$ gilt und jedes Individuum seine genetischen Erbinformationen im Rahmen der chromosomalen Struktur bei sich trägt bzw. weitergibt. Danach werden die drei genetischen Operatoren Selektion, Rekombination (auch Kreuzung bzw. *Crossover* genannt) und Mutation jeweils mit einer bestimmten Wahrscheinlichkeit festgelegt und angewendet, wobei im Zuge jeder Iteration (Alg. 3.3, Zeile 5) Lösungen bzw. Individuen aus der jeweils vorherigen Generation $P(t-1)$ zur Selektion ausgewählt (ergebend die Zwischenpopulation $P_s(t)$ in Zeile 6) und anschließend mit einer bestimmten Wahrscheinlichkeit rekombiniert werden (ergebend $P_{sr}(t)$ in Zeile 7). Die rekombinierten Individuen werden abschließend mit einer definierten Wahrscheinlichkeit mutiert (Zeile 8), bevor die neu geformte Menge an Lösungen der aktuellen Generation, die sogenannte *aktuelle Population* $P(t)$, vollständig bewertet wird (Zeile 9). Der GA verfügt über keine explizite Abbruchbedingung (Zeile 4) und wird nach einer bestimmten Anzahl an Iterationen oder bei Vorliegen einer ausreichend guten Lösung bzw. Stagnation abgebrochen (vgl. Michalewicz 1999, S. 1-19; Feltl 2003, S. 28-37).

¹¹Die Menge aller Gene wird dabei durch einen String bzw. eine Zeichenkette (geformt aus binären Werten) dargestellt.

Algorithmus 3.3 Aufbau und Prinzip eines GA (vgl. Michalewicz 1999, S. 2; Feltl 2003, S. 29)

```

1:  $t \leftarrow 0$ 
2: Initialize  $P(t)$  // create initial population
3: Evaluate  $P(t)$  // evaluate initial population
4: while (not termination-condition) do
5:  $t \leftarrow t + 1$ 
6: Select  $P_s(t)$  from  $P(t - 1)$  // apply selection operator to  $P(t - 1)$ 
7: Recombine  $P_s(t) : P_{sr}(t) \leftarrow P_s(t)$  // apply crossover (recombination) operator to  $P_s(t)$ 
8: Mutate  $P_{sr}(t) : P(t) \leftarrow P_{sr}(t)$  // apply mutation operator to  $P_{sr}(t)$ 
9: Evaluate ( $P(t)$ ) // evaluate fitness of altered (current) population  $P(t)$ 
10: end while

```

Michalewicz hält fest, dass die Balance zwischen der Diversität einer Population und dem dazugehörigen Selektionsdruck fundamental essentiell für den Erfolg eines GA ist. Ein zu geringer bzw. schwacher Selektionsdruck führt zu einer ineffektiven Suche, während ein (zu) starker Selektionsdruck (ggf.) zu früher Konvergenz in Folge mangelnder Diversität in der Population führt. Mangelnde Diversität kann z.B. durch Fitness Sharing-Skalierungsmethoden entsprechend vorgebeugt bzw. reduziert werden (vgl. Michalewicz 1999, S. 58).

Diversität und Fitness Sharing: *Fitness Sharing* ist eine Methode, bei der die Fitness jedes einzelnen Individuums auf der Grundlage seiner Nähe zu anderen Individuen skaliert bzw. beurteilt wird. Die Fitness wird also als gemeinsame Ressource unter den Individuen betrachtet, um die Redundanz in der Population zu reduzieren. Das bedeutet, dass gute Lösungen in bereits dicht besiedelten Regionen der Population einen geringeren Fitnesswert erhalten als vergleichsweise gute Lösungen in dünn besiedelten Regionen. Tatsächlich legt die Auswahltechnik des Algorithmus weniger Gewicht auf diese qualitativ hochwertigen, hochdichten Lösungen. Der Abstand kann auf der Grundlage der Werte entweder im Entscheidungsraum (Genotyp), im Lösungsraum (Phänotyp) oder in beiden berechnet werden (vgl. Grefenstette 1987, S. 41ff.; Rozenberg et al. 2012, S. 1045f.).

Abbildung 3.5 verdeutlicht die Bedeutung des Fitness Sharing als Diversität erhaltende Funktion: Im linken Teil a) wurde die Vielfalt während der gesamten Optimierung beibehalten. Dadurch umfassen die Lösungen einen beträchtlichen Teil der echten Pareto-Front, welche hier als Drahtgitter dargestellt ist. Im rechten Teil b) konvergiert die Population lediglich zu einem kleinen Bereich der Pareto-Front. Entscheidungsträger*innen ziehen i.d.R. die Vielfalt der in Teil a) dargestellten Optionen der (nominalen) Qualitätsverbesserung in Teil b) vor, um das Risiko zu minimieren, Teile der Pareto-Front am Ende der Optimierung nicht erreicht zu haben. Während die Distanz im Genotyp normalerweise mit der Hamming-Distanz¹² definiert wird, erschließt sich die Distanz im Phänotyp über die Euklidische Distanz¹³ (vgl. Merz 2003, F. 101). Fitness Sharing wird auch als eine Art Skalierungsmethode bezeichnet, um den Selektionsdruck und die Diversität der Population entsprechend einzustellen: Die Sharing-Funktion $sh(d_{i,j})$ berechnet sich dabei grundsätzlich über die Ähnlichkeit bzw. Distanz zweier Individuen voneinander, während der sogenannte *niche count* für jedes Individuum die Sharing-Funktion des jeweiligen Individuums über alle Individuen einer Population aufsummiert (vgl. Yu und Gen 2010, S. 171ff.; Rozenberg et al. 2012, S. 1045f.). Über diese Parameter lassen sich formal die Individuen einer Population in Abhängigkeit ihrer Distanz voneinander charakterisieren. Konak liefert eine Übersicht über Diversität erhaltende Mechanismen in MOGA-Algorithmen (vgl. Konak et al. 2006, S. 995).

Während im klassischen GA relativ geringe Mutationsraten und hohe Rekombinationsraten vorherrschen, kann im Falle des Vorliegens einer guten Ausgangslösung eine hohe Wahrscheinlichkeit für Mutationen gepaart mit einer geringen Wahrscheinlichkeit für Rekombinationen positiv zur Konvergenz beitragen (vgl. Kamhuber 2010, S. 26; Sih et al. 2018, S. 447-450). Die fundamentale Ursache dafür liegt in *Holland's Schema-Theorem* (vgl. Holland 2010; Whitley und Sutton 2012, S. 650).

¹²Diese ist in Bezug auf zwei gleichlange Zeichenketten definiert als die Anzahl der Stellen, deren Zeichen (*Bits* bzw. *Allele*) in beiden Ketten (formal) verschieden sind (vgl. Gerdes et al. 2004, S. 47).

¹³Diese stellt die Summe der Quadratwurzel-Differenzen zwischen den Komponenten reeller Vektoren dar.

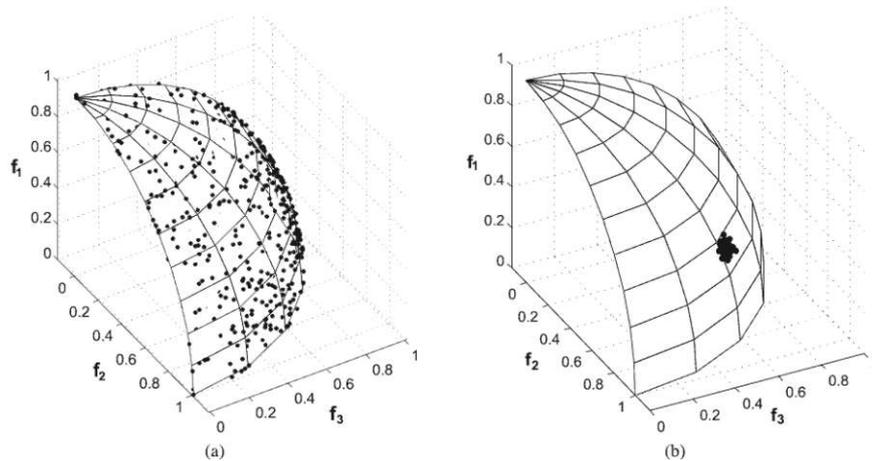


Abb. 3.5: Gegenüberstellung eines (a) gut durchsuchten Lösungsraums mit großer Diversität und (b) weniger gut durchsuchten Lösungsraums mit mangelnder Diversität (vgl. Austin 2020)

Holland's Schema-Theorem: Ein Schema H ist ein Muster bzw. eine Zeichenkette der Länge l , das eine Teilmenge aller Individuen mit gleichen Zeichen (0, 1, *) an bestimmten (Gen) Positionen definiert und mind. einen *Asteriskus* (*) enthält. Dessen Ordnung $o(H)$ wird durch die Anzahl der festen Positionen (0, 1) im (binären) Chromosom determiniert. Die definierende Länge $\delta(H)$ beschreibt den Abstand zwischen dem ersten und letzten festgelegten Zeichen eines Schemas, wobei der Asteriskus als Platzhalter (bzw. Joker oder Wildcard) für 0 oder 1 steht¹⁴. Das Schema $H = 10 * 1*$ verfügt z.B. über die Ordnung $o(H) = 3$. Schema $H = *1 * 01$ hat als definierende Länge $\delta(H) = 5 - 2 = 3$. Für das Schema $H = 1 * 0 * * 1*$ ist $o(H) = 3$ und $\delta(H) = 6 - 1 = 5$ (vgl. Michalewicz 1999, S. 45-55; Gerdes et al. 2004, S. 47-49; Bodenhofer 2003, S. 37). Ein Beispiel für das Wirkprinzip der GA-Suche beschreibt Abschnitt A.14 im Anhang.

Mathematische Überlegungen bezüglich der ds. Fitness eines Schemas und dessen Vermehrungsraten bei Rekombination bzw. Mutation führen zu Holland's Schema-Theorem in Bezug auf genetische Algorithmen. Es besagt, dass sich Schemata mit überdurchschnittlich hoher Fitness, kleiner definierender Länge und geringer Ordnung mit exponentieller Geschwindigkeit vermehren. Diese Schemata werden als *building blocks* bezeichnet. Der GA führt durch das parallele Zusammenfügen derartiger Schemata eine effiziente Suche nach dem Optimum durch. Lösungen mit einer besseren Fitness vermehren sich schneller, weil sie öfters ausgewählt werden. Das Schema-Theorem zeichnet sich dadurch aus, dass das Zusammenspiel der genetischen Operatoren in Kombination mit Mustern bzw. zusammenhängenden Teilen einer Lösung ein Problem effizient löst (vgl. Michalewicz 1999, S. 45-56; Goldberg 1989, S. 41-45; Holland 2010; Altenberg 1995; Whitley und Sutton 2012, S.650-653; Bodenhofer 2003, S. 37-43).

Bei Vorliegen von Schemata mit geringer Ordnung und relativ großer, definierender Länge wird der Einfluss von Letzterem reduziert, wenn die Raten von Rekombination entsprechend reduziert bzw. jene von Mutationen erhöht werden. Damit erklärt das Schema-Theorem bei der Lösung des untersuchten Auftragsreihenfolgeproblems eines Schienenherstellers mittels GA das rasche Auffinden guter Lösungen (vgl. Kamhuber 2010, S. 117f.). Gerdes und Whitley dokumentieren diverse Kritikpunkte am Schema-Theorem, teils bedingt durch die Stochastik der Operatoren sowie begrenzte Populationsgrößen, siehe Abschnitt A.15 im Anhang (vgl. Gerdes et al. 2004, S. 55f.; Whitley und Sutton 2012, S. 653ff.).

Genetische Operatoren: Ein klassischer genetischer Algorithmus besteht im Kern aus den drei in Alg. 3.3 (Zeilen 6 – 8) beschriebenen Operatoren der Selektion, Rekombination und Mutation (vgl. Goldberg 1989, S. 10-14). Dementsprechend wird das Prinzip hinter diesen zentralen Operatoren besprochen, weil es teils auch auf andere (Meta-) Heuristiken übertragbar ist. Die Initialisierung erfolgt über den Erhalt und die Vervielfältigung einer Ausgangslösung, auf Basis von Zufallszahlen generierte Lösungen,

¹⁴Dadurch ergeben sich exakt 2^r mögliche *Spezifikationen* (binäre Zeichenketten) pro Schema, wobei r die Anzahl der Asterisken definiert.

oder über die Konstruktion von Individuen mittels einer geeigneten Heuristik, z.B. der Heuristik von Martello und Toth bzw. der Constraint-Ratio-Heuristik (vgl. Feltl 2003, S. 47ff.).

Selektion: Auf Basis der Selektion wird ermittelt, welche Individuen in jeder Generation für die Reproduktion stochastisch ausgewählt werden. Dieser Operator spiegelt also den natürlichen Ausleseprozess wider. Je nach Optimierungsziel und dem Wertebereich der einzelnen Teilziele wird die Fitness normalisiert. Der im Zuge der Selektion prinzipiell entstehende *Selektionsdruck* bestimmt, wie sehr gute Individuen gegenüber schlechter bewerteten Lösungen begünstigt werden. Während ein zu hoher Selektionsdruck die rasche Vermehrung sogenannter Superindividuen fördert, verringert sich dadurch gleichzeitig die Diversität der Population, und der GA neigt dazu vorzeitig gegen ein lokales Optimum zu konvergieren. Ein zu niedriger Selektionsdruck führt hingegen dazu, dass sich bessere Individuen kaum vermehren und schlechter bewertete Lösungen in der Population bleiben, wodurch das Verfahren kaum oder sehr langsam konvergiert und zur Zufallssuche verfällt. Die Steuerung des richtigen Selektionsdrucks ist ein Schlüsselement im GA (vgl. Feltl 2003, S. 32ff.; Whitley und Sutton 2012, S. 629; Rozenberg et al. 2012, S. 1042f.).

Es existiert eine Vielzahl an verfügbaren, naturinspirierten Selektionsmethoden (vgl. Glover und Kochenberger 2003, S. 65-68), wobei der Selektionsdruck im Zuge der folgenden Auflistung kontinuierlich abnimmt: Trunkierung¹⁵, Tournament-Selektion bzw. Turnierauswahl, Rangbasierte Selektion, Fitnessproportionale Selektion. Während im Zuge der Turnierauswahl das beste Individuum aus einer Zufallsauswahl aus K (gleichverteilten) Individuen selektiert wird, wählt die rangbasierte Selektion die Individuen anhand ihrer Fitness gemäß ihrem Rang in der Population aus (vgl. Whitley und Sutton 2012, S. 629).

Die fitnessproportionale Selektion wählt Individuen mit einer proportionalen, d.h. im Verhältnis zu ihrer Fitness $f(S_i)$, entsprechenden Selektionswahrscheinlichkeit

$$p_S(S_i) = \frac{f(S_i)}{\sum_{j=1}^n f(S_j)}, \text{ wobei } \sum_{j=1}^n f(S_j) > 0 \quad (3.9)$$

aus, siehe auch Abb. 3.6. In Analogie zu einem Roulette-Rad entspricht die Kugel einer zufälligen Zahl im Intervall $[0, 1)$ und selektiert das Individuum, dessen Bereich sie überdeckt (vgl. Feltl 2003, S. 32).

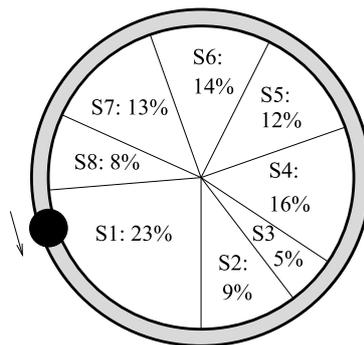


Abb. 3.6: Fitnessproportionale Selektion bzw. Roulette-Rad-Auswahl (Feltl 2003, S. 32)

Die Methode ist auch als *stochastic universal sampling* bekannt, wenn man die Individuen einer Population in zufälliger Reihenfolge als Zahlenreihe auslegt und jedem Individuum eine bestimmte Länge im Verhältnis der jeweiligen Fitness zur Gesamtfitness zugewiesen wird (vgl. Whitley und Sutton 2012, S. 641).

Amous schlägt eine elitäre Selektionsmethode vor, welche, basierend auf der Wurzel der Populationsgröße, die besten n Individuen für den Paarungspool auswählt (vgl. Amous et al. 2008, S. 4f.). Die Steuerung

¹⁵Auswahl der N fittesten Individuen

des Selektionsdrucks ist dabei von zentraler Bedeutung und steht im Zusammenhang mit der Auswahl und Parameterbesetzung des jeweiligen Selektionsverfahrens, beispielsweise der Turniergröße k bei der Tournament-Selektion (vgl. Feltl 2003, S. 33).

Rekombination: Der Prozess der Rekombination bzw. Kreuzung (engl.: Crossover) stellt den Prozess dar, welcher neue Individuen aus den Elternteilen formt und dabei Geninformationen zwischen zwei potentiellen Lösungen austauscht. Die Paarbildung als auch die Wahl der Crossover Punkte geschieht zufällig, wobei eine Vielfalt an klassischen und modernen Varianten der Operatoren existiert. Der klassische One-Point-Crossover basiert auf einer stochastisch gewählten Kreuzungsstelle, wobei der Teil bis zu dieser Stelle von einem Elternteil und ab dieser Stelle von dem anderen Elternteil genommen wird. Eine gute Übersicht über gängige Ausprägungen der Operatoren liefert Sivanandam (vgl. Sivanandam und Deepa 2008, S. 39-81).

Lawrynowicz beschreibt die beiden in der Anwendung beliebtesten Operatoren für die Rekombination, den order crossover und den partial-mapped crossover, ausführlich (vgl. Ławrynowicz 2011, S. 10-11). Der partial-mapped crossover ist eine implizite Mutation, welche Teile einer aktuellen Lösung vererbt (vgl. Merz 2003, F. 155f.) und verfügt über eine, insbesondere für TSP-Anwendungen, vielversprechende Erweiterung (vgl. Koohestani 2020, S. 1ff.).

Grundsätzlich ist anzumerken, dass die Rekombination viel Erbmaterial zerstören kann und die klassischen Operatoren für die Rekombination auf Grund von Nebenbedingungen oft nur schwer anwendbar sind. Daher muss im Zuge der Auswahl des jeweiligen Kreuzungsoperators auf die durchgängige Erzeugung gültiger Lösungen geachtet werden (vgl. Karner 2019, S. 85f.).

Mutation: Im Zuge der Mutation werden bestimmte Gene bzw. Stellen einer Lösung zufällig verändert. So wird z.B. eine einzelne Genposition entweder modifiziert (*Flipmutation*) oder zwei bzw. mehrere zufällig bestimmte Stellen tauschen ihren jeweiligen Geninhalt (*Swap-Mutation* bzw. *Exchange-Based Mutation*), siehe auch Abbildung 3.7 (vgl. Feltl 2003, S. 34f.).

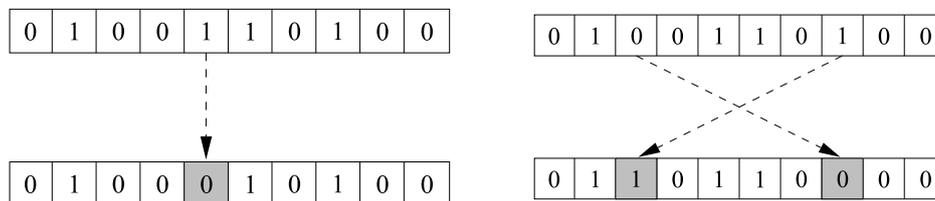


Abb. 3.7: *Flipmutation* (links) und *Swapmutation* (rechts) (Feltl 2003, S. 35)

Auf Basis des Insertion Search Verfahrens (siehe Anhang, A.4.2) wird ein (Gen-) strukturerhaltender *Remove and Insert* Operator vorgestellt. Prinzipiell ist das der Operator, der anderen Optimierungsverfahren durch die stochastisch basierte Veränderung einer Lösung am ähnlichsten ist (vgl. Tseng und Lin 2009, S. 87).

Jiao-Hong Yi stellt einen GA mit einer adaptiven Mutationsrate, basierend auf Pareto-Dominanz, vor, welcher einer fest definierten Mutationsstrategie überlegen ist (vgl. Yi et al. 2018, S. 574f.). Mutation wird als der wichtigste Bestandteil von Evolutionsstrategien angesehen. Die 1/5-Erfolgsregel besagt, dass jede fünfte Mutation eine Verbesserung gegenüber den Eltern darstellt (vgl. Diekmann 2010, S. 3; Whitley und Sutton 2012, S. 629).

Unterschiede zwischen Genetische Algorithmen (GA), Evolutionsstrategien (ES) und Evolutionärer Programmierung (EP): Genetische Algorithmen sind Evolutionsstrategien in einigen grundlegenden Charakteristika sehr ähnlich: Beide verfügen über einen populationsbasierten Ansatz und bedienen sich der Vorteile des Selektions-Mechanismus, welcher dazu führt, dass die fitteren Individuen überleben. Trotz dieser Ähnlichkeiten besteht der grundlegendste Unterschied von GA gegenüber ES im versteckten

Selektions-Mechanismus, sodass bei Evolutionsstrategien eine Zwischenpopulation, bestehend aus der Summe von λ Nachkommen und μ Eltern, gebildet wird, welche anschließend wieder auf die originale Populationsgröße reduziert wird. Bei Evolutionsstrategien kommt ein deterministischer Selektions-Mechanismus zum Einsatz, wobei die besten μ Individuen deterministisch ausgewählt werden, während bei einem GA ein stochastisch basierter dynamisch-präservierender Selektions-Mechanismus zum Einsatz kommt. Während beim GA die Selektion der Rekombination und Mutation vorausgeht, siehe Algorithmus 3.3, folgt die Selektion beim EA nach diesen verändernden Operatoren. Während die Reproduktions-Parameter (Wahrscheinlichkeitsrate von Rekombination und Mutation) bei einem GA zur Laufzeit konstant bleiben, sind diese bei einer ES selbst-adaptiv und veränderlich. Evolutionäre Programmierung (EP) ist, trotz der großen Ähnlichkeit, eine Abwandlung von Evolutionsstrategien. EP kommt ohne Rekombination aus und konzentriert sich stattdessen auf die Mutation als einzig verwendeten Suchoperator. EP verwendet probabilistische Selektionsoperatoren (z.B. Turniererlektion, etc.) und reduziert dabei (wie auch ES) die Population von $\lambda + \mu$ auf μ Individuen (vgl. Michalewicz 1999, S. 164-168; Glover und Kochenberger 2003, S. 71f.).

Diskussion und Fazit: Banzi vergleicht mehrere multikriterielle Metaheuristiken (GA, ACO, TS) in Bezug auf eine Analyse der jeweils eingesetzten Operatoren und hält für seine Problem-Instanzen fest, dass spezifisch zugeschnittene Operatoren den klassischen Operatoren überlegen sind (vgl. Banzi et al. 2012, S. 12140f.). Dies liefert auch den Hauptgrund, warum es nicht nur auf die Auswahl des richtigen Algorithmus ankommt, sondern auch auf die für dieses Problem am besten geeigneten Operatoren. Felzl beschreibt diesbezüglich den GA von Chu und Beasley (vgl. Chu und Beasley 1997, S. 17-23), welcher über einen eigenen, heuristischen Reparatur-Operator verfügt, um die Qualität einzelner Individuen zu verbessern bzw. ungültige Lösungen in gültige Lösungen zu überführen (vgl. Felzl 2003, S. 40ff.). Genetische Algorithmen werden häufig zur Hybridisierung, d.h. in Kombination mit anderen Verfahren, auf Grund ihrer globalen Sichtweise eingesetzt. Sazvar und Saracoglu stellen eine umfassende Übersicht über veröffentlichte Literatur in Bezug auf das MPMP Bestandsproblem in Bezug auf verderbliche Produkte zur Verfügung und, wie bei Wari bereits erwähnt, setzen sich dafür (Hybride) Genetische Algorithmen auf Grund ihrer Beliebtheit in der Anwendung durch (vgl. Saracoglu et al. 2014, S. 8189-8202; Sazvar et al. 2016, S. 232-261). Obwohl die Partikelschwarmoptimierung, siehe Anhang A.8.2, bei mittelgroßen und großen Test-Instanzen im Mittel (5-10%) schneller ausführbar ist als ein GA, liegt der GA in der Lösungsfindung knapp vorne (vgl. Sazvar et al. 2016, S. 253f.). Jans and Degraeve haben diverse Metaheuristiken untersucht und sind ebenfalls zu dem Schluss gekommen, dass sich Genetische Algorithmen, speziell für mittelgroße und schwere MPMP-Probleme besonders gut eignen (vgl. Jans und Degraeve 2007, S. 1855-1875). Die Begründung liegt darin, dass sich der*die Analyst*in dreier verschiedener Operatoren problemspezifisch bedienen und anpassen und derart relativ viel Einfluss auf die spezifische Ausgestaltung bzw. Optimierung des Problems nehmen kann. Zhou, Mashwani und Ponsich liefern eine ausführliche Übersicht bezüglich multikriterieller Evolutionsstrategien (vgl. Zhou et al. 2011, S. 32-49; Mashwani 2011, S. 374-392; Ponsich et al. 2013, S. 321-344).

An dieser Stelle sei noch erwähnt, dass sich bei einem GA mit *variierender Populationsgröße* die Anzahl der Individuen in einer Population über Generationen hinweg verändern kann. Das Konzept des Alters eines Chromosoms ist äquivalent zu der Anzahl an Generationen, die es am Leben bleibt bzw. dem GA zur Verfügung steht. Das fitnessabhängige Alter, welches einmalig zu Beginn der Optimierung (also bei der Geburt des Individuums) fitnessabhängig definiert wird, ersetzt den Selektions-Operator und beeinflusst dadurch die Größe der Population eines GA in jeder Iteration. In der Population eines GA wird nun anstelle der Selektion bei jeder Iteration das Alter gleichmäßig, z.B. um eins, erhöht und am Ende der Generation werden alle Individuen aus der Population entfernt, deren Lebensdauer abgelaufen ist. Michalewicz stellt auch einige Lebensdauer-Berechnungsstrategien zur Verfügung und ermittelt in den *meisten* Tests etwas bessere Ergebnisse als der Standard GA von Goldberg. Da die Populationsgröße großen Einfluss auf die Anzahl an Bewertungen hat, ist es auch möglich, dass der Algorithmus selbst steuert, wann er seine Population vergrößert bzw. reduziert. Die von Michalewicz vorgestellte Methode verfügt über eine gewisse Ähnlichkeit zu Evolutionsstrategien, welche aus μ Eltern λ Nachkommen generieren und im Gegensatz zum GA primär mutieren und sekundär rekombinieren (vgl. J. Arabas et al. 1994, S. 73-78; Michalewicz 1999, S. 72-80; Michalewicz und Fogel 2004, S. 151ff.).

3.3.3 Simulated Annealing (SA)

Simulated Annealing ist auch unter den Begriffen *probabilistic hill climbing*, *Monte Carlo Annealing* bzw. *stochastische Relaxation* bekannt und basiert auf einer der Thermodynamik entnommenen Analogie in Bezug auf Optimierungsprobleme (vgl. Michalewicz und Fogel 2004, S. 120f.). Das Prinzip dieses Verfahrens, basierend auf dem 1953 vorgestellten Algorithmus von Metropolis, basiert auf dem Abkühlungsprozess von Kristallen in der Metallurgie (vgl. Metropolis et al. 1953, S. 1087ff.). Ein Metall bzw. Kristall soll in einen stabilen und damit belastbaren Zustand gebracht werden. Daher wird es zunächst erhitzt und anschließend einer kontrollierten Abkühlung (*annealing*) unterzogen. Wenn dieser Auskühlungsprozess *zu rasch* abläuft, werden dabei bestimmte Unregelmäßigkeiten in der Kristallstruktur festgehalten und das eingeschlossene Energieniveau ist deutlich höher als in einem perfekt strukturierten Kristall (vgl. Schoner 2008, S. 58f.).

Funktionsprinzip und Analogien aus der Thermodynamik: Im Zuge der Anwendung des Verfahrens wird ein Gleichgewichtszustand von Atomen bei einer gegebenen Temperatur T simuliert. Die Grundidee basiert darauf, dass gefundene schlechtere Zustände (bzw. Lösungen in der Optimierung) mit einer bestimmten Wahrscheinlichkeit akzeptiert werden. Ausgangspunkt ist ein Zustand, welchem bei konstanter Temperatur T eine bestimmte Energie E entspricht. Gesucht wird ein solider *Grundzustand* mit einem minimalen Energieniveau, welcher metastabile Strukturen vermeidet. Eine zufällig gewählte Veränderung des Systems entspricht der Energieänderung dE des Systems. Während negative Energieänderungen unmittelbar akzeptiert werden, wird im anderen Falle die Wahrscheinlichkeit des Auftretens eines höheren Energiezustands berechnet, wobei diese Berechnung auf dem Boltzmann Gesetz beruht (vgl. Schröder 2010, S.389; Boussaïd et al. 2013, S. 83f.):

$$P(dE) = e^{-dE/(k \cdot T)} \quad (3.10)$$

Die Boltzmann Konstante k wird in der Optimierung durch eine problemspezifisch angepasste Konstante ersetzt. Die aus der Gleichung (3.10) resultierende Wahrscheinlichkeit $P(dE)$ wird mit einer Zufallszahl Z verglichen und akzeptiert, sofern $Z < P(dE)$ ist. Nach vielen Iterationen entwickelt sich das System in den nach dem Boltzmanngesetz wahrscheinlichsten Zustand. Diese Analogien sind in Tabelle 3.2 ersichtlich. Das damit verbundene Hauptmerkmal von Simulated Annealing liegt dementsprechend darin, dass es eine Möglichkeit anbietet, lokalen Optima durch die Akzeptanz von ggf. mehreren, aufeinander folgenden, schlechteren Zügen zu entkommen (vgl. Glover und Kochenberger 2003, S. 288).

Tab. 3.2: Analogien zwischen der Thermodynamik und einem Optimierungsproblem (Quelle: eigene Darstellung, in Anlehnung an Michalewicz und Fogel 2004, S. 121)

Physikalisches System	Optimierungsproblem
Zustand	Zulässige Lösung
Energie	Zielfunktion
Grundzustand	Optimale Lösung
Rasches Abschrecken	Lokale Suche
Temperatur	Regelparameter T
Simuliertes Abkühlen bzw. Ausglühen	Simulated Annealing

Ablauf der Optimierung mittels Simulated Annealing: Algorithmus 3.4 beschreibt die grundlegenden Schritte des Optimierungsverfahrens: Zu Beginn wird ein Abkühlungsplan, bestehend aus einer initial festgelegten Temperatur T_{init} und einer dazugehörigen Abkühlungsrate α bestimmt. Diese beiden Parameter definieren, wie stark bzw. *schnell* die initial definierte Temperatur schrittweise bis zum Abbruchkriterium (Alg. 3.4, Zeile 16), z.B. dem Erreichen von T_{min} , fällt. Im Idealfall erhält eine Temperatur so viele Bewertungen bzw. Iterationsschritte, bis sich ein thermodynamisches Gleichgewicht einstellt (Zeile 15). Diese Anzahl kann initial entweder über eine Konstante definiert bzw. dynamisch

dem System überlassen werden. Anschließend werden auf der jeweils aktuellen Temperatur lokale Lösungen aus der Nachbarschaft der jeweils aktuellen Lösung s gewählt (Zeile 6). Die Umsetzung berücksichtigt an dieser Stelle ggf. auch zusätzliche heuristische Informationen in Bezug auf das Aufsuchen und die Bestimmung neuer Lösungen. Während bessere Lösungen vom Algorithmus sofort angenommen werden (Zeile 7), akzeptiert das Verfahren schlechtere Lösungen nur mit einer bestimmten Wahrscheinlichkeit p (Zeilen 10 - 12), welche einerseits von einer stochastisch ermittelten Zufallszahl Z und andererseits von der aktuellen Temperatur T und der absoluten Zielfunktionswertdifferenz der beiden verglichenen Lösungen abhängt (vgl. Chiandussi et al. 2012, S. 912-942; Boussaïd et al. 2013, S. 83f.; Schröder 2010, S.387-392; Schoner 2008, S. 58-60; Raidl und Chwatal 2008, F. 140-150).

Algorithmus 3.4 Simulated Annealing (vgl. Boussaïd et al. 2013, S. 84; Schoner 2008, S. 59; Raidl und Chwatal 2008, F. 142)

```

1:   $T \leftarrow T_{\text{init}}$  //  $T > T_{\text{min}}$ , damit eine Abkühlung erfolgen kann
2:   $s \leftarrow \text{Startlösung}$ 
3:  repeat
4:    repeat
5:      wähle ein  $s' \in N(s)$  zufällig //  $s'$ ... lokale Lösung aus der Nachbarschaft von  $s$ 
6:      if  $f(s') \leq f(s)$  then
7:         $s \leftarrow s'$ 
8:      else
9:         $Z \leftarrow \text{Random}(0, 1)$  // bestimme Zufallszahl  $Z$ 
10:       if  $Z < e^{-|f(s')-f(s)|/T}$  then // Boltzmann-Wahrscheinlichkeit
11:          $s \leftarrow s'$ 
12:       end
13:     end
14:   until thermodynamisches Gleichgewicht für aktuelle Temperatur erreicht
15:    $T \leftarrow \alpha \cdot T$  //  $\alpha$ ...Abkühlungsplan – Konstante :  $0 < \alpha < 1$ 
16:   until Abbruchkriterium erfüllt // z.B.  $T \leq T_{\text{min}}$ 
17: return beste gefundene Lösung  $s$ 

```

Je mehr die aktuelle Temperatur abkühlt, desto geringer wird prinzipiell die Wahrscheinlichkeit deutlich schlechtere Lösungen zu akzeptieren, während diese anfangs bei einer höheren Temperatur noch viel mehr gegeben ist. Diese Akzeptanzregel mit der sogenannten Boltzmann-Wahrscheinlichkeitsverteilung

$$p(\text{akzeptiere } s') := \begin{cases} 1 & \text{wenn } f(s') \leq f(s) \\ \exp\left(-\frac{f(s')-f(s)}{T}\right) & \text{sonst} \end{cases} \quad (3.11)$$

basiert auf der Idee der Abkühlung von geschmolzenem Metall zu einem Festkörper. Simulationsexperimente haben bestätigt, dass sich bisher keine andere Wahrscheinlichkeitsverteilung als besser erwiesen hat (vgl. Schoner 2008, S. 59f.).

Schoner liefert einige Vorschlagswerte in Bezug auf das Abkühlungsschema. So beträgt die anfängliche Akzeptanzwahrscheinlichkeit in Bezug auf schlechtere Lösungen idealerweise 75% und 95%, für α hat sich der folgende Wertebereich bewährt: $\alpha \in [0.9, 0.99]$. Das Abbruchkriterium (Zeile 16) kann über eine Endtemperatur fest vorgegeben werden, ab welcher keine Veränderung mehr zugelassen wird, oder über einen Maximalwert für die Gesamtanzahl der Iterationen seit der letzten Zielfunktionswertverbesserung (Kriterium der *Stagnation*) (vgl. Schoner 2008, S. 60).

Ming stellt eine umfangreiche Sammlung von modernen und erfolgreichen SA Anwendungen in den Domänen Produktion und Verkehr (Multikriterielle Zielsetzungen, Hybridisierung von GA, Reihenfolgeprobleme, Routing Probleme, etc.) bzw. praktischen Überlegungen für die SA Implementierung (z.B. zur alternativen Perturbation einer Lösung (vgl. Ming 2008, S. 401-420) zur Verfügung (vgl. Ming 2008, S. 1-420). Außerdem wird SA heute im Bereich der Künstlichen Intelligenz, z.B. im Zuge der Anwendung von künstlichen neuronalen Netzen, verwendet (vgl. Ghasemalizadeh et al. 2016, S. 1668-1686).

Vor- und Nachteile des Verfahrens:

Chiandussi verweist auf eine umfangreiche Literatursammlung und hält dabei fest, dass der SA teilweise als *too greedy* angesehen wird. Dementsprechend ist die Parametrisierung des Abkühlungsplans bei diesem Verfahren besonders wichtig, weil es sich einerseits um ein *einzellösungsbasiertes* Verfahren handelt, d.h. alle Iterationsschritte passieren ausgehend von der jeweils aktuellen Lösung. Andererseits ist der Mechanismus, der es erlaubt, dass sich ein SA lokalen Minima entziehen kann, im Rahmen der Anwendung essentiell wichtig (vgl. Chiandussi et al. 2012, S. 84).

Abbildung 3.8 verdeutlicht die zentrale Bedeutung dieses Mechanismus: Nach Erreichen eines lokalen Minimums schafft es das Verfahren, sich aus dieser besten Lösung wieder herauszulösen und dabei schlechtere Funktionswerte anzunehmen, bevor es wieder weitere lokale Minima erreicht, bis es schlussendlich in einem globalen Minimum landet, aus welchem es, bedingt durch die inzwischen erreichte Temperatur, nicht mehr entkommt. Wichtig ist dabei zu bedenken, dass sich das Verfahren iterativ verschlechtern kann, d.h. möglicherweise mehrere schlechte Züge hintereinander akzeptiert. Dies wird auch als Unruhe oder *Perturbation* bezeichnet. Diese Unruhe nimmt im Laufe des Verfahrens mit abnehmender Temperatur ab. Schröder demonstriert diese Trajektorie im Zuge einer beispielhaften Optimierung mittels SA (vgl. Schröder 2010, S. 390-392).

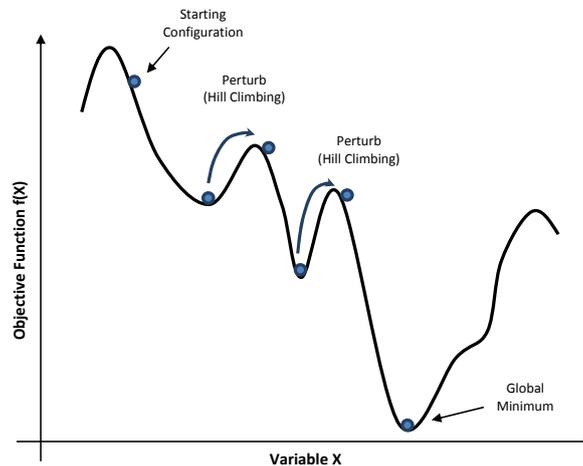


Abb. 3.8: Globale Optimierung mittels Simulated Annealing (Ghasemalizadeh et al. 2016, S. 1671)

SA stellt zusammenfassend eine relativ einfach und gut verständlich zu implementierende Metaheuristik dar und liefert, in Kombination mit einem als geeignet bestimmten Abkühlungsplan, relativ gute Ergebnisse. Eine entsprechende Implementierung eines SA ist auch in der Lage, einen GA zu übertreffen (vgl. Karner 2019, S. 185).

Im direkten Vergleich von verschiedenen Metaheuristiken ist Simulated Annealing insbesondere bei Auftragsreihenfolgeproblemen relativ erfolgreich. Der SA wurde v.a. auf Grund mangelnder Robustheit nicht als bestes Verfahren empfohlen. Bei Vorliegen einer bereits heuristisch optimierten Lösung kann durch SA deutlich effizienter eine globale Optimierung vorgenommen werden als beispielsweise durch ein populationsbedingt exploratives, genetisches Verfahren. Als alleiniges Verfahren ist es für eine globale Optimierung bedingt v.a. für kleinere Probleme empfehlenswert, wenn man mehrere Läufe durchführt und heuristische Informationen integriert (vgl. Karner 2019, S. 108f.; Kamhuber 2010, S. 119).

Delahaye bestätigt, dass Simulated Annealing auf Grund seiner Simplizität häufig in verschiedensten Domänen in Kombination mit Simulation erfolgreich in realen Anwendungen ausgeführt wird, weil es speicherintensiven, populationsbasierten und dadurch in der Ausführung langsamen (z.B. genetischen) Verfahren teilweise überlegen ist. Die Problematik liegt darin begründet, dass populationsbasierte Verfahren je nach Größe ihrer Populationsgröße n viele Bewertungen pro Generation g durchführen,

und dabei pro Individuum lediglich bis zu 2 Änderungen (Mutation, Rekombination) pro Individuum vornehmen, während eine einzellösungsbasierte Metaheuristik in derselben Laufzeit bereits n konsekutiv ausgeführte Bewertungen an der aktuellen Lösung vornimmt. Diesen Effekt kann man auch durch die Verwendung von relativ kleinen Populationen reduzieren (vgl. Delahaye et al. 2019, S. 1ff; Sobottka et al. 2018, S. 417ff.).

Weitere Metaheuristiken (Ant Colony Optimization, Partikelschwarmoptimierung, Tabu Search, Variable Neighborhood Search, Iterierte Lokale Suche) werden im Anhang im Abschnitt A.8 näher diskutiert.

3.3.4 Fazit

Das No-Free-Lunch-Theorem lässt grundsätzlich die Annahme zu, dass keine der vorgestellten Metaheuristiken (in Reinform) prinzipiell einer anderen vorzuziehen ist. Es hängt demnach von der Art des Einsatzes bzw. der dazugehörigen Problemcharakterisierung ab, welches Verfahren gewählt wird. Grundsätzlich sind daher Genetische Algorithmen, Ameisenalgorithmen, als auch Simulated Annealing auf Grund ihrer häufigen Anwendung in der Praxis und vielseitigen Implementierbarkeit die Favoriten, wobei Ameisenalgorithmen wegen ihres konstruktiven Charakters in dieser Arbeit in der Anwendung unpraktisch sind. Ein ACO verlangt, dass sich eine Ameise jedes Mal die komplett aufzuspannende Matrix, bestehend aus allen Artikeln und Planungsperioden, neu konstruiert. Auf Grund der Vielzahl der Planungsperioden in Kombination mit dem konstruktiven Charakter sind viele Nebenbedingungen während der Konstruktion einer Lösung durch jede einzelne Ameise einzuhalten, wodurch sich die Implementierung dieses Algorithmus erschwerend und unvorteilhaft gestaltet, da sich die Prüfung der Nebenbedingungen erst nach bzw. aus der vervollständigten Matrix ergibt.

Dieser Abschnitt erlaubt die zentrale Schlussfolgerung, dass auf Basis der Heuristik als gute Startlösung eine Metaheuristik zur Verfeinerung und stochastischen Suche aufgesetzt und implementiert wird. Diese bewertet alle Teilziele gemeinsam in einer global definierten Zielfunktion und setzt dabei direkt auf die beste heuristisch ermittelte Lösung auf. Dieser Umstand favorisiert sowohl den die chromosomale Struktur erhaltenden GA als auch SA gegenüber dem konstruktiven ACO Verfahren. Der Autor entscheidet sich für die Gewichtete-Summen-Methode (siehe 3.2.3.1) in Kombination mit einer problemspezifisch angepassten Implementierung eines Genetischen Algorithmus, der mit einem möglichst gleichwertig umgesetzten Simulated Annealing Verfahren verglichen wird. Simulated Annealing eignet sich erfolgreich in simulationsbasierten Anwendungen, wo sich mit jeder Simulationsbewertung der Speicher füllt und die Laufzeit des GA entsprechend der Anzahl an Individuen einer Generation vervielfacht. Außerdem behält der SA, im Gegensatz zur Diversität eines GA, eher die grundsätzliche Lösungsstruktur der besten gefundenen heuristischen Lösung (vgl. Gendreau und Potvin 2019, S. 19).

Ein Schlüsselement einer Metaheuristik stellt eine Ausgewogenheit zwischen Lösungsvielfalt und Lösungsgeschwindigkeit dar. Dementsprechend verfügt ein ausgewogenes Verfahren über einen global explorativen Charakter, als auch die lokale Ausnutzung von heuristischem Wissen (vgl. Yang 2012, S. 1-3).

Trotz der genannten Vorteile des SA kommt in der überwiegenden Mehrzahl, knapp 90%, der Publikationen im Bereich der lebensmittelherstellenden Produktion ein populationsbasiertes Verfahren zum Einsatz, wobei rund 70% der populationsbasierten Ansätze einen Genetischen Algorithmus verwenden. Dieser bietet die Vorteile, dass er sich während der Suche nicht auf die Trajektorie einer einzelnen (besten) Lösung beschränkt, sondern populationsbasiert mehrere verschiedene Lösungen pro Generation gleichzeitig bewertet und weiterentwickelt. Dementsprechend ist es beim GA auch möglich, dass eine relativ schwach bewertete Lösung sich nach einigen Änderungen zunehmend (d.h. iterativ) verbessert und im Lösungspool stets mitgenommen wird, während sich trajektorienbasierte Verfahren lediglich auf die Optimierung von einer (als beste bewertete) Lösung konzentrieren. Außerdem bietet der GA grundsätzlich die Verwendung von (bis zu) drei Operatoren, und damit mehr Individualisierungs-Potential, als z.B. ein SA Verfahren an (vgl. Wari und Zhu 2016, S. 328-343).

Die Praxis zeigt außerdem, dass bei trajektorienbasierten Verfahren die Ergebnisse zwischen einzelnen Optimierungsläufen deutlich stärker schwanken als bei robusteren populationsbasierten Metaheuristiken (vgl. Sobottka et al. 2017, S. 443).

Die Implementierung des GA bzw. SA in dieser Arbeit fokussiert sich, neben der synchronen Betrachtung bzw. Bewertung aller Teilziele, auf von der Heuristik weniger stark explizit optimierte Teilziele. Für Simulated Annealing wird eine Studie bezüglich der Parameteroptimierung im Rahmen eines *Grid Search-Verfahrens* benötigt. Beim GA entfällt diese grundsätzliche Parameteroptimierung, stattdessen wird der Einfluss der Populationsgröße sowie des Selektionsoperators untersucht. Damit die beiden Algorithmen vergleichbar sind, wird der Schritt der Suche nach einer neuen Lösung auf Basis einer Veränderung der aktuellen Lösung gleichwertig umgesetzt.

Der Methodik fehlt grundsätzlich noch eine spezifische Strategie, um mit der gewählten gewichteten Präferenz ausdrückenden Methode möglichst viele Punkte der Pareto-Front zu erreichen bzw. untereinander zu bewerten. Jedenfalls wird eine Normierung aller Teilzielfunktionswerte angestrebt. Die Gewichtete-Summen-Methode bietet, auf Basis der Abwägung aller Vor- und Nachteile (siehe auch 3.2.3.1), den entscheidenden Vorteil des einfachen Lösungsverständnisses sowie der leichten Parametrisierbarkeit durch den*die Anwender*in. Der nächste Abschnitt gibt deshalb Aufschluss, welche Möglichkeiten der Hybridisierung gemäß dem Stand der Technik im relevanten Kontext existieren und welche davon für einen rollierenden Planungsansatz sinnvoll sind. Es liegt nahe, dass eine rollierende Planung bereits eine Vielzahl an Planungsdaten mit sich bringt, welche im Zuge der Entwicklung und kontinuierlichen Anwendung der Methodik laufend analysiert wird. Die Ergebnisse der heuristisch-lexikographischen Optimierung werden im Nachgang einmalig mit der definierten, gewichteten Zielfunktion evaluiert, damit die Lösungsgüte der heuristischen Optimierung quantitativ bewertet vorliegt.

3.4 Hybride Optimierungsverfahren

Dieser Abschnitt erweitert den Stand der Technik um sogenannte hybride Optimierungsverfahren. Das sind verschiedenste Kombinationen einzelner Optimierungsverfahren untereinander. Außerdem werden spezifische Optimierungsstrategien diskutiert, die im Rahmen dieser Arbeit relevant sind (vgl. Blum et al. 2011, S. 4135-4151; Talbi 2016, S. 171-215).

Technische Klassifikation: Raidl klassifiziert die Ausprägung hybrider Metaheuristiken in Form von vier Stufen, basierend auf der Taxonomie von Talbi (vgl. Talbi 2016, S. 171-215; Gendreau und Potvin 2019, S. 385-417):

1. Die Art der Ausprägung hybridisierter Algorithmen.
2. Den Grad der Hybridisierung.
3. Die Reihenfolge der Ausführung.
4. Die Steuerungsstrategie.

Einerseits werden Komponenten verschiedener Metaheuristiken miteinander kombiniert. Zweitens werden Metaheuristiken mit stark problemspezifischen Heuristiken bzw. einer Simulation gekoppelt. Drittens werden Metaheuristiken mit anderen OR- bzw. AI- Verfahren kombiniert. Die OR-Methoden stellen v.a. exakte Methoden (Dynamic Programming¹⁶, Constraint Programming, Branch-and-Bound, etc.) dar (vgl. Kumar und Aouam 2019, S.429-448; Edis 2021, S. 1ff.), während unter AI Methoden z.B. Neuronale Netzwerke, Fuzzy Logic oder Statistische Methoden verstanden werden. Der Grad der Hybridisierung regelt die Kopplung der Algorithmen aneinander. So können die Algorithmen jeweils ihre Identität bewahren und aufeinander folgen (*schwache Kopplung*), oder auch in Abhängigkeit voneinander stehen, also integriert implementiert sein oder bestimmte Teile oder Funktionen voneinander vermischen (*starke Kopplung*). Die Reihenfolge der Ausführung findet im einfachen Fall *in serieller Abfolge* statt, während anspruchsvollere Ansätze die Algorithmen *ineinandergreifend* oder in *paralleler* Art und Weise miteinander koppeln. Die Kontrollstrategie wird integrativ oder kollaborativ ausgeführt. Im beliebten

¹⁶DP ist eine rekursive, algorithmische Technik, die Optimierungsprobleme sequenziell in einfachere Teilprobleme zerlegt und bedingt, dass die optimale Lösung des Gesamtproblems von der optimalen Lösung der Teilprobleme abhängt (vgl. Michalewicz und Fogel 2004, S. 93-101). Das Vorgehen wird z.B. in der Graphentheorie angewendet.

integrativen Anwendungsfall stellt ein Verfahren die untergeordnete Komponente eines anderen dar. Im Gegensatz dazu tauschen die individuellen Algorithmen im kollaborativen Ansatz Informationen aus, ohne Teil des anderen zu sein (vgl. Talbi 2016, S. 171-215; Gendreau und Potvin 2019, S. 385-417).

Bezüglich der Taxonomie hält Yang fest, dass es grundsätzlich zwei Kategorien gibt, nämlich kollaborativ arbeitende, hybride Algorithmen und integrativ ausgeführte Hybridverfahren. Während die erste Kategorie üblicherweise als mehrstufiges Optimierungsverfahren betrachtet wird, wo mehrere (zumeist zwei) Verfahren sequenziell ausgeführt werden (z.B. globale Suche durch Algorithmus eins und anschließende lokale Suche durch das angeschlossene zweite Verfahren), können die Algorithmen auch alternativ oder parallel ausgeführt werden. Bei Parallelität laufen beide Verfahren gleichzeitig, wobei ein Verfahren auf einen bestimmten Prozentsatz des anderen Algorithmus oder eines Teiles davon, z.B. der Population eines GA, angewendet wird (vgl. Yang 2015, S. 76f.). Bei integrativ ausgeführten Hybridlösungen wird ein Verfahren in das andere, den sogenannten Master, eingebettet. Der Beitrag des zweiten Verfahrens reduziert sich dabei auf ca. 10-20%. So kann beispielsweise der Mutations-Operator des GA in einer Partikelschwarmoptimierung verwendet werden. Dabei existiert die Möglichkeit der vollen Manipulation, wo die komplette Population in jeder Iteration entsprechend durch das Subverfahren manipuliert wird, oder die teilweise Manipulation, wo nur ein Teil der gesamten Population entsprechend anders behandelt wird. Die Herausforderung besteht in der richtigen Parametrisierung der Anteile, sowohl prozentuell als auch in der zeitlichen Abfolge, d.h. wann die Prozedur im Verfahren angewendet wird (vgl. Yang 2015, S. 71-83).

Hader unterscheidet in seiner Taxonomie zwischen drei Grundtypen, nämlich *assimiliert-hybrid*, *eingebettet-hybrid* und *separat-hybrid*, wobei diese in Reinform als auch gemischt auftreten. Assimiliert-hybride Verfahren erweitern ein Basisverfahren um Ideen, Ansätze oder Konzepte aus anderen Methoden entweder als echte Erweiterung oder Ersatz existierender Bestandteile. Eingebettet-hybride Verfahren integrieren ein anderes Optimierungsverfahren, welches zur Laufzeit auch wiederkehrend aufgerufen wird, zur Unterstützung in das Hauptverfahren, während separat-hybride Verfahren aus einer Reihe gleichberechtigter Verfahren, welche separat, d.h. sequenziell (*seriell-hybrid*) oder auch unabhängig voneinander (*parallel-hybrid*) ausgeführt (vgl. Hader 2000, S. 8ff.).

3.4.1 Varianten der Hybridisierung

Der Fokus liegt heute, sinngemäß in Übereinstimmung mit dem No-Free-Lunch-Theorem, auf der problemspezifisch gerechten Lösung eines vorliegenden Problems, ohne dabei eine bestimmte Metaheuristik zu bewerben. Die Forschung hat dementsprechend den Fokus vom algorithmusorientierten Ansatz auf den problemorientierten Ansatz verschoben. Die vorliegende Arbeit wird dies bestätigen. Blum liefert Hybridisierungen von Metaheuristiken mit anderen (Meta-) Heuristiken, Constraint Programming, Baumbasierten Suchverfahren, Relaxationsbasierten Ansätzen und Dynamic Programming. Juan stellt eine Übersicht über die Kombination bzw. Kopplung von Metaheuristiken mit Simulation in Bezug auf die verschiedensten Anwendungsbereiche (Produktion und Logistik, Supply-Chain Management, Medizinbereich etc.) zur Verfügung (vgl. Juan et al. 2015, S. 62-72). Talbi wiederum liefert einen sehr umfangreichen Beitrag, wo er die Möglichkeiten der Hybridisierung und Anwendungen in Kombination mit Mathematischen Verfahren, Constraint Programming und Machine Learning Methoden aufzeigt. Auf Basis der bisher aufgezeigten Hybridisierungs-Möglichkeiten konzentrieren sich die folgenden Unterabschnitte auf die Kombination mit (Meta-) Heuristiken, Simulation und Machine Learning Methoden (vgl. Talbi 2016, S. 171-215). Diese Arten der Hybridisierung weisen für die Bewältigung der Aufgabenstellung in dieser Arbeit die größte Bedeutung bzw. Relevanz auf. Die Kombination mit exakten Verfahren wird in dieser Arbeit nicht betrachtet, weil das der Suchraum des komplexen, real vorliegenden industriellen Problems nicht genügend zulässt.

3.4.2 Kombination mit Heuristiken bzw. Metaheuristiken

Die Idee, Metaheuristiken untereinander bzw. Heuristiken mit Metaheuristiken zu kombinieren, ist sehr populär und wird häufig angewendet, um v.a. die Vorteile der individuellen Eigenschaften von

Methoden in Bezug auf die Suche und Erschließung neuer, vielversprechender Lösungen zu kombinieren. Yannibelli und Amandi integrieren einen multikriteriellen SA-Algorithmus in den dazugehörigen evolutionären Algorithmus. Dieses Konzept kombiniert die evolutionärbasierte, explorative Suche und das zielgerichtete Verhalten des SA. Der SA ist dabei insofern an den EA gekoppelt, als er sowohl explorativ global als auch zielgerichtetes, lokales Suchverhalten je nach aktuellem Fortschritt des EA aufweist (vgl. Yannibelli und Amandi 2013, S. 2421-2434). Die *serielle* Kopplung eines GA mit einem konsekutiv ausgeführten lokalen Suchverfahren, konkret dem *Remove and Insert* Verfahren, hat vergleichsweise die besten Resultate für ein reales Auftragsreihenfolgeproblem geliefert (vgl. Kamhuber 2010, S. 73ff.).

Blum stellt z.B. ein populationsbasiertes Iterated Local Search Verfahren (siehe Anhang A.8.5 bezüglich des ILS Verfahrens) vor, welches als sehr kompetitiv in Bezug auf ein quadratisches Zuordnungsproblem bezeichnet wird. Anstelle eines einzellösungsbasierten ILS Verfahrens wird eine Population P , bestehend aus n Lösungen, gleichzeitig betrachtet. Während dabei kurzfristig $2n$ Lösungen entstehen, werden davon die schlechtesten n Lösungen in der nächsten Generation gelöscht, siehe auch Alg. 3.5. Hybridisierungen eignen sich v.a. auch für die mehrstufige Optimierung von Produktionssystemen, um einerseits die Vorteile von verschiedenen Verfahren zu erhalten und andererseits eine tragfähige Lösung zu einem komplexen realen Problem mit einem großen Suchraum zu erhalten (vgl. Gendreau und Potvin 2019, S. 476). So werden beispielsweise häufig problemspezifisch heuristische Konstruktionsheuristiken zur Findung einer initialen Lösung verwendet, welche anschließend durch eine Metaheuristik verbessert wird. Oft wird dabei auch das originale Problem relaxiert (vgl. Gendreau und Potvin 2019, S. 474).

Kelner stellt in seinem Beitrag eine Kombination aus einem Evolutionären Algorithmus und einer lokalen Suchstrategie vor. Dabei wird die lokale Suchstrategie auf bestimmte Individuen der Population angewendet, sodass diese einen definierten Anteil der originären Population ersetzen. Diese integrative Kopplungsprozedur verhilft dem hybriden Algorithmus nicht nur dabei, aus lokalen Minima zu entkommen, sondern auch, rasch gegen das globale Optimum zu konvergieren (vgl. Kelner et al. 2008, S. 448-456).

Algorithmus 3.5 Populationsbasierte Iterierte Lokale Suche (vgl. Blum et al. 2011, S. 4136f.)

```

1:  $P \leftarrow$  Generiere Initialpopulation ( $n$ )
2: Führe lokale Suche für alle  $s \in P$  aus
3: while (not termination-condition) do
4:    $P' \leftarrow P // P'$ ...modifizierte Population
5:   for all  $s \in P$  do
6:      $s' \leftarrow$  Perturbation ( $s$ , history)
7:      $\hat{s}' \leftarrow$  Lokale Suche ( $s'$ )
8:      $P' \leftarrow P' \cup \{\hat{s}'\}$ 
9:   end for
10:  $P \leftarrow$  Selektiere  $n$  beste Lösungen aus  $P'$ 
end while

```

Das Verfahren ist immer anwendbar, wenn eine Nachbarschaftsstruktur für ein Problem entwickelt wird. Dieses Verfahren ist dem ILS Verfahren v.a. bei Vorliegen von verstreuten lokalen Optima über den gesamten Suchraum hinweg überlegen. Andererseits ist es auf Grund des populationsbasierten Ansatzes lauffeuchtintensiver (vgl. Blum et al. 2011, S. 4137).

Talbi unterscheidet in seiner Taxonomie zwischen einer Relay-Hybridisierung, welche zwei Verfahren nacheinander ausführt, und einer Teamwork-Hybridisierung, welche kooperativ ausgeführte Optimierungsverfahren beschreibt. Weiters differenziert er zwischen einer Low-Level und einer High-Level-Hybridisierung. Während bei der Low-Level-Hybridisierung Teile eines Optimierungsverfahrens durch ein anderes Verfahren ausgetauscht bzw. ersetzt werden, sind die verschiedenen Optimierungsverfahren bei einer High-Level Hybridisierung in sich geschlossen. Das bedeutet, dass kein direkter, interner ausführungsbezogener Zusammenhang zwischen den Verfahren besteht (vgl. Talbi 2016, S. 171-185).

Talbi hält in seinem Beitrag weiters fest, dass sich Genetische bzw. Evolutionäre Algorithmen auf Grund

ihrer drei genetischen Operatoren relativ gut zur Hybridisierung eignen. Ein Low-Level-Teamwork Hybrid-Algorithmus integriert eine einzellösungsbasierte Metaheuristik (z.B. SA, TS) in den populationsbasierten GA. Dafür eignet sich insbesondere der Mutations-Operator zur Hybridisierung mit lokalen Suchverfahren, während der Rekombinations-Operator typischerweise *blind* ist und somit keine heuristischen Informationen verwendet. Gleichmaßen führt er auch Beispiele an, wo der Crossover-Operator mit problemspezifischen Informationen bestückt wird. Die Hybridisierung erfolgt nicht nur statisch, sondern auch konditional bzw. adaptiv. Diese Adaption stellt z.B. ein bestimmtes Ereignis oder auch eine Stagnation des Optimierungsfortschritts dar. High-Level Relay Hybridisierung bedeutet, dass die Initiallösung für eine Metaheuristik von einer (*greedy*) Heuristik bzw. einem Eröffnungsverfahren ermittelt wird. Das funktioniert gleichermaßen für einzellösungsbasierte als auch populationsbasierte Metaheuristiken, siehe Abb. 3.9 (vgl. Talbi 2016, S. 174ff.).

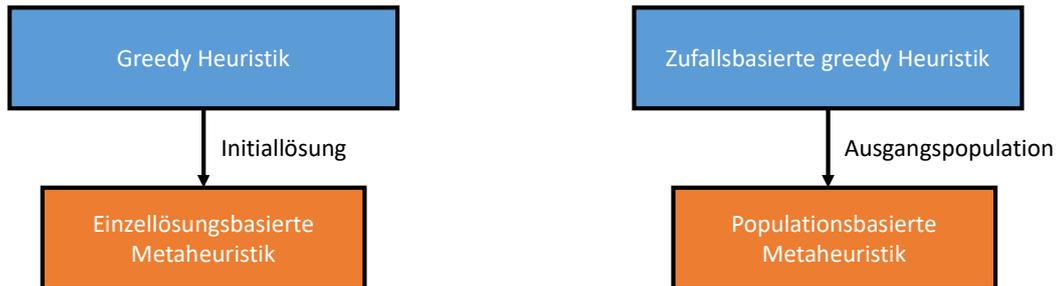


Abb. 3.9: High-Level Relay Hybridisierung (eigene Darstellung, angelehnt an Talbi 2016, S. 176)

Eine weitere, häufige Kombination besteht in der Kopplung einer populationsbasierten Metaheuristik für die globale Suche und einer einzellösungsbasierten Metaheuristik für die anschließende lokale Suche. Die einzellösungsbasierte Metaheuristik kann dabei für die ganze Population, vielversprechende Teile daraus oder auch nur auf das beste Individuum angewendet werden. Eine Metaheuristik kann außerdem dazu verwendet werden, eine andere möglichst optimal zu parametrisieren (vgl. Talbi 2016, S. 176f.).

Mashwani beschreibt den Stand der Technik in Bezug auf hybrid ausgeführte, genetische und evolutionäre Algorithmen mit multikriteriellen Zielsetzungen und erläutert dazu die folgenden Strategien (vgl. Mashwani 2011, S. 374ff.):

1. Evolutionäre Algorithmen in Kombination mit lokalen Suchverfahren, welche auf die besten Individuen angewendet werden.
2. Pareto-Dominanz basierte, multikriterielle, evolutionäre Algorithmen: Non-dominated Sorting Genetic Algorithms (*NSGA*, *NSGA-II* bzw. *NSGA-III*): Diese Methoden sortieren die Population hierarchisch in Sub-Populationen, basierend auf der Reihung der Pareto-Dominanz, ein.
3. Verbesserte Versionen multikriterieller, evolutionärer Algorithmen: Mashwani nennt einen GA, welcher im Rahmen der 2-Opt lokalen Suche (siehe Anhang A.4.1) zwei Nachbarschaften verwendet, wobei eine neu erhaltene Lösung mehrere alte Lösungen ersetzt. Ein anderer Beitrag kombiniert die gewichtete Summenmethode und die Tchebycheff-Methode als Skalierungsfunktionen. Die Anwendung hat die Effektivität insofern bestätigt, als die simultane Verwendung von beiden *Maßen* der Nutzung von nur einer der beiden Skalierungsfunktionen überlegen ist. Weitere genannte Beiträge kombinieren PSO bzw. GA mit einem *Dekompositionsverfahren*, welches ein Gesamtproblem in Teile zerlegt, jeweils mit einer Zielfunktion versieht und für sich optimiert.
4. Multimethodische Suchansätze: Diese Ansätze kombinieren die Stärken einer adaptiven Evolutionsstrategie mit genetischen Algorithmen und Partikelschwarmoptimierung. Durch die Selbstadaption wird die Populationsgröße determiniert.

Raidl erwähnt außerdem sogenannte *variable fixing strategies*, welche einige Variablen an bestimmte Werte binden und die Optimierung über diesen beschränkten Suchbereich ausführen. Dies stellt eines der Kernkonzepte zur Lösung von Knapsack-Problemen dar. Weiters hält Raidl fest, dass der Trend

weg von einer generalisierten Metaheuristik hin zu einer leistungsstarken, effizient implementierten Metaheuristik geht. Dieser Trend spiegelt sich an sogenannten *Hyper-Heuristiken*, welche Algorithmen darstellen, die mit wenig Aufwand für potentiell ähnliche Probleme (automatisiert und) angewendet werden können (vgl. Blum et al. 2011, S. 4137). Diese Arbeit leistet einen entsprechenden Beitrag durch die Entwicklung einer über die Anwendung hinausgehende Heuristik, die mit wenig Aufwand an ein neues Problem anpassbar ist.

Talbi beschreibt mehrere Möglichkeiten, um Metaheuristiken mit problemspezifischen Wissen auszustatten (vgl. Talbi 2016, S. 202f.):

1. Hybridisierungen, die das Wissen aus den Daten vorab der Lösung des OR-Problems extrahieren (*offline*), oder *online* Strategien, welche den Wissensaustausch während der Optimierung dynamisch durchführen.
2. Ziel der Hybridisierung: Entweder eine Verbesserung der Effizienz oder der Effektivität des Algorithmus. Ersteres wird z.B. durch eine Suchraumreduktion oder einer Approximation der Zielfunktion erreicht, während zweiteres durch die Integration von Wissen oder eine adaptive Parameter-Aktualisierung passiert.
3. Auf Ebene der Suchkomponente, z.B. im Zuge der Kodierung bzw. Initialisierung von Lösungen oder den Suchvariationsoperatoren (Mutation, Rekombination, Nachbarschaft).

Ein anderer Zweig der Hybridisierung beschäftigt sich mit Techniken, welche die Laufzeit reduzieren, die Qualität der Resultate verbessert, oder beides. Blum beschreibt dazu eine Tabu-Suche, die durch heuristische Manipulation zusätzliche Nebenbedingungen, welche derart den Suchraum reduzieren, erhält. Weitere Arbeiten hybridisieren genetische Algorithmen mit Partikelschwärmen, oder Tabu Search mit Scatter Search Prinzipien (vgl. Blum et al. 2011, S. 4135-4151; Mashwani 2011, S. 374-392).

3.4.3 Simheuristics: Kombination mit Simulation

Juan, Faulin, Grasman, Rabe und Figueira stellen in ihrem Beitrag einen aktuellen Rückblick über die Kombination von Metaheuristiken zur Lösung kombinatorischer Optimierungsprobleme mit simulationsbasierten Techniken vor (vgl. Juan et al. 2015, S. 62-72). Simulation erlaubt die Modellierung und künstliche Reproduktion komplexer Systeme auf natürliche Art und Weise. Die Simulationsdetails können in mathematische Modelle und Verteilungen einfließen, stochastische Prozesse (Bearbeitungs- und Rüstzeiten, Transporte, Lagerbestandsverläufe, Ausfallszeiten) können entsprechend akkurat und detailgetreu berücksichtigt werden (z.B. Bearbeitungszeiten in Form einer Dreiecksverteilung, Ankunfts-raten in Warteschlangensystemen bzw. Zwischenpuffer mit einer Poisson Verteilung, Bedienraten mit einer Exponentialverteilung etc.). Ziel ist dabei, eine höhere Qualität der Lösung zu erreichen, sodass das dargestellte Ergebnis über mehr Glaubwürdigkeit in der realen Welt verfügt, als ein entsprechendes Optimierungsverfahren ohne Berücksichtigung von Simulation. Die Erweiterung einer Metaheuristik um eine integrierte Simulation zur besseren Lösung und Optimierung bezeichnet Juan als Simheuristik und beschreibt diesbezüglich zwei ausgeprägte Charakteristika (vgl. Juan et al. 2015, S. 63f.):

1. Engere Verknüpfung zwischen Optimierung und Simulation. Die Bewertung der Zielfunktion ist dabei ggf. nicht zur Gänze auf Simulation zurückzuführen, wenn ein problemspezifischer und relevanter Teil der Zielfunktion mittels Simulation gelöst wird.
2. Das Simulationsfeedback wird nicht nur zur Bewertung der Lösung verwendet, sondern verfeinert stattdessen den analytischen Teil.

Prinzipiell wird zwischen simulationsgestützten (*Simulation Optimization*) und optimierungsbasierenden (*Hybrid Simulation-Analytic*) Ansätzen unterschieden. Das Konzept der simulationsgestützten Optimierung verwendet die Simulation zur Bewertung der Zielfunktion. Stattdessen verwenden optimierungsbasierenden Ansätze Simulation dazu, das analytische Modell zu verbessern bzw. zu verfeinern, oder um einen Teil der Lösung zu generieren (vgl. Juan et al. 2015, S. 63f.). Beim Einsatz von Simulation wird je nach Anwendung zwischen diskreter, kontinuierlicher und hybrider Simulation unterschieden (vgl. Sobottka et al. 2018; Sihn et al. 2018). Diskrete, ereignisgestützte Simulation¹⁷ wird v.a. für

¹⁷engl.: Discrete-event-simulation (DES)

die Szenarienentwicklung im Zuge der Fabrik- und Lagerplanung oder für die Dimensionierung von Anlagen, Zwischenpuffern oder der Beurteilung der Leistungsfähigkeit von industriellen Prozessen verwendet. In bestimmte DES-Software-Umgebungen sind teilweise auch integrierte und vorgefertigte Optimierungssolver verfügbar, z.B. der GAWizard in PlantSimulation bzw. OptQuest in Arena (vgl. Bergeron et al. 2009, S. 1091-1095; Haraszko und Németh 2015, S. 143-152). Diese bieten jedoch nur begrenzt problemspezifisch einstellbare Möglichkeiten (Parameter, Nebenbedingungen, etc.) im Zuge der Optimierung an und sind daher nicht sinnvoll für die Methodenentwicklung in dieser Arbeit geeignet. Dias gibt eine vergleichende Übersicht bezüglich der weltweit am häufigsten eingesetzten DES-Software-Tools, darunter Arena, ProModel, FlexSim, Simio, PlantSimulation und AnyLogic (vgl. Dias et al. 2016, S. 1060-1071).

Praktische Einsatzmöglichkeiten von Simulation: Lothar März liefert in seinem Werk einen praxisorientierten Leitfaden zu den Themen Simulationsgestützte Optimierung, den dazugehörigen Stell- und Zielgrößen und widmet sich auch der Kopplung von Simulation und Optimierung. Bezüglich der dadurch entstehenden Komplexität hält er den enormen Bedarf an Rechenaufwand anhand eines Beispiels, der Minimierung der Zykluszeit für n Aufträge in einem Fertigungssystem, fest. Bereits die Variation der Auftragsreihenfolge vor der Engpassmaschine liefert $n!$ mögliche Permutationen (vgl. März et al. 2011, S. 37f.).

Die ereignisdiskrete Simulation bietet als Werkzeug zur Analyse und Bewertung von realen Produktionssystemen mit stochastischem Verhalten eine mächtige Methode, um komplexe Systeme möglichst realitätsnah zu bewerten. Derartige Systeme weisen den Bedarf hinsichtlich einer Optimierung der Kriterien Ressourcenauslastung, Durchlaufzeit, Termintreue oder Kosten auf. Daher eignet sich Simulation zur Kombination mit Optimierungsmethoden, welche ein hinreichend einfach zu modellierendes System erfordern, an das vereinfachte Annahmen unterstellt werden. Die Integration von Simulation erlaubt die Berücksichtigung praxisrelevanter Komplexität sowie stochastischer Ereignisse (Ausfallsverhalten, stochastische Bearbeitungs- und Rüstzeiten, Wartezeiten, etc.). Für die Art der Kopplung sind die gegenseitige Abhängigkeit als auch die zeitliche Abfolge der Berechnungsschritte relevant, wodurch sich sequenzielle und hierarchische Verknüpfungen unterscheiden lassen (vgl. März et al. 2011, S. 41f.).

Folgende Arten der Kopplung lassen sich unterscheiden (vgl. März et al. 2011, S. 42ff.):

1. **Integration der Optimierung in die Simulation:** Die führende Methode ist die Simulation, auf dessen Basis bzw. Ergebnisse die Optimierung die vorhandene Fragestellung löst.
2. **Simulation als Bewertungsfunktion der Optimierungsaufgabe:** Dieser Fall entspricht der am häufigsten anzutreffenden Kopplungsart. Die Simulation wird durch die Optimierung gestartet und stellt die Ergebnisdaten als Grundlage für eine darauf basierende Bewertung des dynamischen bzw. stochastischen Verhaltens des abzubildenden Produktionssystems dar.
3. **Simulation als Startwert der Optimierung:** Die Simulationsergebnisse liegen geschlossen vor und werden als Eingangsgröße für die anschließende Optimierung verwendet. Ein Beispiel ist z.B. die Ermittlung der Kapazitätsanforderungen für den Bau einer Fabrik bzw. der Dimensionierung der dazugehörigen Produktionsanlagen. Im Nachgang der Optimierung werden die so ermittelten Kapazitätsbedarfe mit den Betriebsmitteln verknüpft.
4. **Ergebnisse der Optimierung zur Konfiguration der Simulation:** Dabei wird mittels Simulation die Machbarkeit einer durch die Optimierung vorgeschlagenen Lösung validiert, weil die Optimierung als vereinfachtes Modell teilweise die kausalen bzw. stochastisch entstehenden Bedingungen nicht vollständig analytisch löst bzw. antizipiert.

März unterscheidet drei verschiedene Problemklassen für die Auswahl des richtigen Optimierungsverfahrens: Reihenfolgeprobleme, Zuordnungsprobleme und Probleme, welche sich mit der Variation von Parametern (Losgrößen, Planzeiten, Leistungsgrößen, Pufferflächen, etc.) beschäftigen. Im Bereich der klassischen industriellen Produktion wird diskrete Simulation häufig dazu verwendet, optimale Puffergrößen in Bezug auf eine Stückgutproduktion zu erhalten. In den weiteren Kapiteln liefert März diverse Fallbeispiele aus den Themenfeldern einer optimierten Personaleinsatzplanung bzw. Reihenfolge in der

Automobilindustrie bzw. Endmontage auf Basis einer personalorientierten Simulation. Die Fallbeispiele (Kapitel 7, 12, und 14) belegen, dass der optimierte Personaleinsatz ein Themenschwerpunkt ist, der von Simulation betrachtet bzw. bewältigt werden soll (vgl. März et al. 2011, S. 44f.).

Im Zuge der Fabrikplanung bei dem in dieser Arbeit betrachteten Lebensmittelhersteller werden die Ergebnisse einer statischen Optimierung als Initiallösung und Grundlage für eine darauf basierende Kapazitätssimulation verwendet (Art 4). Diese Kopplung wurde initial gewählt, um die kapazitive Grundlage und Voraussetzung für diese Arbeit zu schaffen. Im Zuge der rollierenden Betrachtung des vorliegenden MOMPMP-Problems wird die Simulation v.a. als (Teil-) Bewertungsfunktion der vorliegenden Optimierungsaufgabe herangezogen (Art 2). Diese Teilbewertung umfasst auch den optimierten Personaleinsatz an der Engpassanlage des neuen Werks.

Matthias Gruber¹⁸ stellt in seinem Buchbeitrag das Konzept der simulationsgestützten Optimierung mittels SiRO (Simulation Rating Optimization) vor, siehe auch Abbildung 3.10. Im Zuge der Bewertung des Verfahrens wird bemängelt, dass die Laufzeit der Optimierung durch eine relativ komplexe Prozessablaufsimulation limitiert ist (vgl. März et al. 2011, S. 65-77).

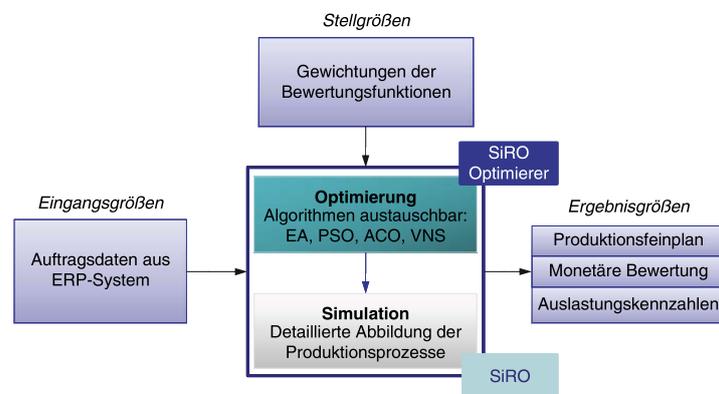


Abb. 3.10: Konzept der Simulationsgestützten Optimierung mittels SiRO (März et al. 2011, S. 68)

Eine ähnliche Schlussfolgerung ist aus der Kombination von hybrider Simulation und metaheuristischer Optimierung hervorgegangen. Eine hybride Simulation ist eine Kombination aus diskreter und kontinuierlicher Simulation, welche das thermo-physikalische Verhalten berücksichtigt. Den Hauptgrund gegen den Einsatz von hybrider Simulation liefert die benötigte Rechenzeit zum Lösen des energetischen (*kontinuierlichen*) Teils des Problems (vgl. Heinzl 2020, S. 49-87; Sobottka et al. 2018, S. 413-420).

Der Beitrag von Freitag verwendet das Konzept der simulationsgestützten Optimierung (mittels MOGA) als Hyper-Heuristik, um automatisiert optimierte Planungsregeln zu erstellen. Anhand eines Anwendungsfalles aus der Halbleiterindustrie wird demonstriert, dass der Stand der Technik damit signifikant übertroffen wird. Dabei werden die Individuen des GA mit einer Menge von Standard-Planungsregeln (first in first out, earliest release time first, shortest processing time, etc.) befüllt. Die Individuen liegen dabei in Form von Baumstrukturen vor, siehe auch Mallawaarachchi (vgl. Vijini Mallawaarachchi 2020). Das Ziel der Methodik besteht darin, das am besten geeignete Set an Regeln für das vorliegende Problem zu ermitteln (vgl. Freitag und Hildebrandt 2016, S. 433-436). Das Konzept ist einerseits interessant, aber für das vorliegende Problem andererseits nicht unmittelbar relevant, weil die Zielsetzung in der Suche nach einer konkreten Lösung besteht. Das Konzept ist indirekt auf diese Arbeit übertragbar, sodass im Zuge dieser Arbeit eine Methodik entsteht, welcher in ihrer Form als Hyper-Heuristik auf ähnliche Probleme übertragbar ist.

¹⁸Florian Felix Kamhuber hat unter der Betreuung von Matthias Gruber (Profactor GmbH) seine Masterarbeit mit dem Titel „Simulationsbasierte metaheuristische Optimierung eines realen Auftragsreihenfolgeproblems“ an der Montanuniversität Leoben entwickelt und verfasst.

3.4.4 Kombinationen mit Machine Learning (ML) und Data-Mining (DM) Methoden

Die Hybridisierung von Metaheuristiken mit Machine Learning bzw. Data-Mining Methoden stellt eine weitere Möglichkeit dar, die Effizienz und Effektivität der jeweiligen Optimierungsverfahren positiv als auch automatisiert zu beeinflussen. Data Mining beschreibt den Prozess der automatisierten Analyse und Auswertung von großen Datenmengen, um daraus für die Untersuchung des Problems interessante Erkenntnisse (z.B. Muster) abzuleiten. Data Mining bedient sich von Methoden aus der Statistik, dem Maschinellen Lernen und der Mustererkennung. Die Unterscheidung in Bezug auf den Einsatz des jeweils geeignetsten Verfahrens hängt von der Aufgabe ab, wobei Talbi zwischen dem überwachten Lernen (*supervised learning*) und dem nicht-überwachten Lernen (*unsupervised learning*) differenziert. Typische Aufgaben des überwachten Lernens umfassen (vgl. Talbi 2016, S. 201):

- **Klassifikation:** Das ist die Einteilung bzw. Zuweisung von bestimmten Attributen und Eigenschaften einer zu untersuchenden Instanz in zuvor definierte Kategorien bzw. Klassen. Die Partitionierung erfolgt häufig über ein Ähnlichkeitsmaß, welches über eine (z.B. euklidische) Distanz realisiert wird.
- **Erlernen von Klassifizierungsregeln:** Entdeckung einer Menge an Regeln (*Set* bzw. *Regelwerk*) anhand der Daten und Bildung eines genauen Klassifikators.

Die häufigsten Aufgaben des nicht überwachten Lernens stellen das (vgl. Talbi 2016, S. 201):

- **Clustering** (Die Partitionierung mittels Ähnlichkeits- oder Distanzmaß des Eingabedatensatzes in Teilmengen, sogenannte *Cluster*, sodass die darin enthaltenen Daten die gleichen bzw. ähnliche Aspekte aufweisen).
- **Erlernen von Assoziationsregeln** (Auffinden von Elementen, welche innerhalb eines gegebenen Datensatzes gemeinsam auftreten).

Eine weitere Aufgabe besteht in der Auswahl der wesentlichen Merkmale bzw. der Reduktion auf die entscheidenden Features für das vorliegende Modell, um dessen Komplexität zu reduzieren und gleichermaßen die Modellgenauigkeit zu erhöhen. Lingitz hat z.B. auf Basis von 41 entscheidenden Charakteristika, sogenannten *Features*, sein Durchlaufzeitprognosemodell erstellt, und im Verhältnis 70/30 die Datenmengen in Trainings- und Testdaten aufgeteilt. Einige Features beruhen auf statischen Eigenschaften, welche das jeweilige Los charakterisieren, beispielsweise Produkt(typ), Kund*in, Planlieferzeit, als auch dynamischen Eigenschaften, die den aktuellen Zustand des Produktionssystems reflektieren. Die Auswahl der entscheidenden Features durch den*die Analyst*in für ein Modell ist dabei entscheidend (vgl. Lingitz et al. 2018, S. 1051-1056; Zitzler et al. 2001, S. 107).

Teerasoponpong stellt in seinem Beitrag zur simulations- und datengestützten Optimierung eine Methodik für arbeitsintensive produzierende KMU vor. Das Simulationsmodell wird dabei durch ein künstliches neuronales Netz für die Simulationsabbildung und die Erkennung von Datenbeziehungen genutzt, die Optimierung und Konfiguration bzw. Anpassung des Produktionssystems erfolgt über einen GA. Die Methodik erlaubt eine akkurate Prädiktion des Durchsatzes mit einer bestimmten Menge an Arbeitskräften. Nachteilig wird in diesem Zusammenhang der zeitaufwändige Prozess für die Datensammlung- und Aufbereitung erwähnt (vgl. Teerasoponpong und Sopadang 2021, S.1-13).

Diese Aufgabe wird bei einem GA konzeptionell von *Holland's Schema-Theorem* übernommen (vgl. Whitley und Sutton 2012, S. 653-655). Garre verwendet ML Algorithmen, um Abweichungen in der Produktion vorherzusagen, wodurch sich die Unsicherheit in Bezug auf die produzierte Menge an Müll reduziert (vgl. Garre et al. 2020, S. 1ff.).

3.4.5 Spezifische Optimierungsstrategien

Dieser Unterabschnitt diskutiert die vielversprechendsten aktuellen und problemspezifischen für diese Arbeit relevantesten Ansätze und Methoden bzw. methodischen Erweiterungen oder Verfeinerungen im Kontext der Produktionsglättung. Die vorgestellten, teils hybriden, Strategien beziehen sich v.a. auf glättungsbezogene Aspekte, wobei die Methoden selbst auch *generisch* im Zuge der Optimierung einsetzbar sind (vgl. Chang et al. 2002, S. 171-183).

3.4.5.1 Gradual-Priority Weighting-Ansätze

Chang stellt einen multikriteriellen GA mit dem *Gradual-Priority Weighting*-Ansatz vor, wobei die Gewichtungen über die Laufzeit hinweg systematisch variiert werden, um die Suche in möglichst alle Regionen des Zielraumes zu führen. Als Anforderungen nennt er, dass die Priorität der einzelnen Teilziele vorab bekannt sind, sodass der GA anfangs mit der initial definierten Gewichtung arbeitet. Dann wird die Gewichtung des wichtigsten Teilziels sukzessive graduell verringert, während sie beim zweitwichtigsten Teilziel schrittweise entsprechend erhöht wird. Dieser Ansatz ist für beliebig viele Teilziele und auch auf jedes andere (metaheuristische) Verfahren übertragbar (vgl. Chang et al. 2002, S. 171-183).

Denselben Ansatz haben auch Jin, Okabe und Sendhoff verfolgt und haben ihren Buchbeitrag als *Adapted Weighted Aggregation for Multiobjective Evolution Strategies* bezeichnet. Die Gewichtungen werden auch hier schrittweise und periodisch im Zuge des Evolutionsfortschritts verändert. Im Gegensatz zum Ansatz von Chang verwenden sie dafür spezielle Funktionen für die beiden zu manipulierenden Gewichtungen, nämlich (vgl. Zitzler et al. 2001, S. 96-110):

$$w_1(t) = | \sin(2\pi t)/F | \quad (3.12),$$

$$w_2(t) = 1.0 - w_1(t) \quad (3.13),$$

wobei t für die Anzahl an Generationen und $| \cdot |$ für den Absolutbetrag steht. Die Änderungsfrequenz wird mittels F bestimmt, wobei die Autoren für $F = 400$ einsetzen. Das bedeutet, dass sich alle 400 Generationen w_1 von 0 nach 1 bewegt und anschließend wieder von 1 in Richtung 0, und das viermal. Abbildung 3.11 zeigt ein Beispiel für die dynamische Änderung der Gewichtungen im Laufe der Evolution. Die Autoren sind der Meinung, dass die Resultate des Algorithmus nicht sensitiv in Bezug auf F sind. Wichtig ist, dass alle Individuen jeder Generation dieselbe Gewichtung aufweisen (vgl. Zitzler et al. 2001, S. 96-110).

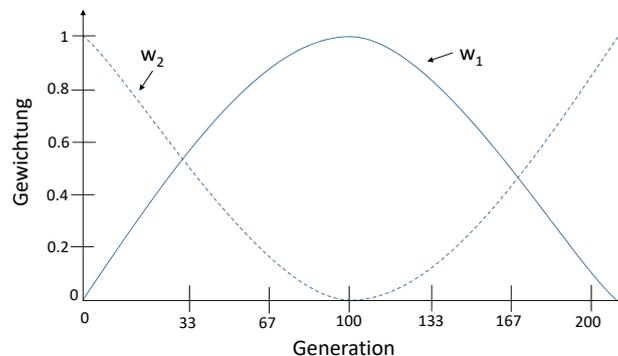


Abb. 3.11: Verlauf der adaptiven Gewichtung (Zitzler et al. 2001, S. 100)

Gemäß dem Autor dieser Arbeit spiegelt dieser Ansatz die Evolution wider, wo sich die Zielsetzungen der Gesellschaft im Rahmen der Jahrhunderte verändert haben. Während z.B. in der Steinzeit die Fitness anhand der Fähigkeit zu laufen, Wild zu jagen und Beeren zu sammeln geprägt war, hat sich die Fitness im Laufe der Zeit gewandelt hin zu einer intellektuellen-digitalisierten Gesellschaft, welche über die prinzipiellen Möglichkeiten verfügt, sich gesund und nährstoffreich durch eine ausreichende Versorgung mit Lebensmitteln zu ernähren. So erklärt sich der Wandel einer Gewichtung (dieser Menge an Fähigkeiten) von 1 in Richtung 0. Für die Optimierung trifft das nicht zur Gänze zu, weil sich die Zielsetzungen nicht in diesem Ausmaß ändern. Stattdessen geht es vielmehr darum, andere evtl. vielversprechendere Bereiche der Pareto-Front zu erreichen, was mit einer einzigen, vorab definierten Gewichtung per se ggf. nicht möglich ist.

Zitzler et al. vergleichen die periodisch-dynamisch angepassten Gewichtungen mit einem einfacheren Ansatz auf Basis von zufällig verteilten Gewichtungen innerhalb einer Population und kommen zu

dem Schluss, dass beide Ansätze die Pareto-Front erreichen. Speziell für Probleme mit geringer Dimension sind kaum Unterschiede feststellbar. Die Ansätze erreichen selbst konkave Pareto-Fronten. Bemerkenswert ist allerdings, dass für höherdimensionale Probleme der dynamische Ansatz dem einfacheren Ansatz überlegen ist. Außerdem ist die Qualität der Ergebnisse ihrer Meinung nach konkret von den gewählten Strategien hinter den genetischen Operatoren bzw. der gewählten Parametrisierung abhängig (vgl. Zitzler et al. 2001, S. 102-108).

3.4.5.2 Multi-objective Genetic Algorithm (MOGA)

Der MOGA-Ansatz reiht ein Individuum anhand aller anderen Individuen, von denen es dominiert wird, wobei allen nichtdominierten Individuen der beste Rang zugeordnet wird. Dominierte Lösungen werden dementsprechend gemäß der Populationsdichte im Lösungsraum ausgehend vom besten bis zum schlechtesten Individuum der Population gemäß einer (z.B. linearen) Funktion interpoliert bzw. bestraft (*Pareto Ranking*), wobei die Fitness von gleich bewerteten Individuen gemittelt wird. Der MOGA Ansatz ist effizient, vergleichsweise einfach zu implementieren und wird häufig im Steuerungsdesign angewendet (vgl. Zitzler et al. 2001, S. 27).

Der NSGA, ursprünglich von Srinivas und Deb als eine MOGA-Variante vorgeschlagen, basiert auf mehreren Klassifizierungsebenen bezüglich ihrer Individuen. Die Individuen der Population werden gemäß Pareto-Ranking gereiht. Alle nichtdominierten Lösungen eines Teils der Pareto-Front werden dabei in einer Kategorie (ähnlich einem Turnier) zusammengefasst klassifiziert und erhalten dabei einen Dummy-Fitnesswert proportional zur Populationsgröße. Um die Diversität der Population zu erhalten, werden die klassifizierten Individuen in Bezug auf ihre Dummy-Fitnesswerte im Lösungsraum verwendet. Anschließend wird diese Kategorie bzw. Gruppe an Individuen aus der Population entfernt und die nächste Ebene nichtdominierter Individuen wird betrachtet, bis alle Individuen klassifiziert sind. Individuen der ersten Ebene verfügen über die größte Fitness und erhalten mehr Kopien gegenüber dem Rest der Population. Die Suche in nichtdominierten Regionen erlaubt eine relativ starke Konvergenz, während die Mitbenützung von Individuen dazu beiträgt, dass sich die Population über die gesamte Region in dieser Ebene verteilt (vgl. Zitzler et al. 2001, S. 28).

Dieser Ansatz wurde zum NSGA-II erweitert, welcher Elitismus¹⁹ verwendet und die Diversität der Selektion über einen Vergleichsoperator (in Bezug auf den Abstand der Lösungen voneinander) gewährleistet. Yusoff hält fest, dass sich der NSGA-II v.a. für die Optimierung von Prozessparametern maschineller (z.B. für zerspanende) Bearbeitungsschritte eignet (vgl. Yusoff et al. 2011, S. 3978-3983).

Campos-Ciro bzw. Yi stellen eine berechnungseffiziente Erweiterung (NSGA-III) vor, wobei Yi eine adaptive Mutationsrate für den Mutations-Operator verwendet, welcher den traditionellen Ansatz übertrifft (vgl. Yi et al. 2018, S. 571-585; Campos-Ciro et al. 2016, S. 1272-1277).

Saracoglu verwendet im Rahmen der Modellierung seines MPMP-Problems einen GA mit einer in den GA eingebetteten lokalen Suche und erreicht dadurch qualitativ hochwertige Lösungen innerhalb relativ kurzer Zeit auch für größere Instanzen seines Problems (vgl. Saracoglu et al. 2014, S. 8189-8202).

Konak, Coit und Smith halten fest, dass sich der GA im Kontext realer technischer Probleme, bei denen es um die gleichzeitige Optimierung mehrerer Ziele geht, mit ihrer leistungsstarken, populationsbasierten Suche und ihrer Flexibilität, die sich aus der maßgeschneiderten Implementierung ihrer Operatoren ergibt, am vielversprechendsten im Vergleich mit anderen Metaheuristiken erweist. Im Zuge der Konzeptionierung eines GA erwähnen sie, dass ein auf Elitismus basierender GA einem GA, welcher ohne dem Konzept des Elitismus arbeitet, in der Regel überlegen ist. Die Autoren empfehlen außerdem, in der Selektion die Auswahl von lediglich nichtdominierten Lösungen, wie dies auch bei einem NSGA-II passiert. Der Grund liegt darin, dass eine auf Zufall basierende Selektion nicht garantiert, dass nur nicht-dominierte Lösungen in die nächste Generation mitgenommen werden. Im Gegensatz zum NSGA-II Konzept empfehlen die Autoren die Verwendung dynamischer Populationsgrößen²⁰. Dabei wird auf den Erhalt der Diversität innerhalb der Population geachtet, siehe den entsprechenden Paragraphen *Diversität und Fitness Sharing* in 3.3.1.1 (vgl. Konak et al. 2006, S. 992-1007).

¹⁹Elitismus bedeutet, dass die besten Individuen in die nächste Generation mitgenommen werden.

²⁰Beim Erreichen der jeweils oberen Schranke werden $N_{max} - N_{min}$ Individuen aus der Population entfernt.

3.4.5.3 Random Search Production Smoothing Heuristic

Karimi-Nasab stellt in seiner Arbeit eine auf Random Search basierende Heuristik vor, welche den zulässigen Lösungsraum effizient erforscht und dabei eine initiale Liste an Pareto-optimalen Lösungen erstellt. Die vorgestellte Methode arbeitet mit Pseudozufallszahlen und verbessert die Lösungen in Richtung der stärksten, globalen Verbesserungsrichtung für gleichzeitig möglichst viele Teilziele. In der zweiten Phase wird diese Liste finalisiert, indem nur die *global* Pareto-optimalen Lösungen in der Liste bestehen bleiben. In der finalen dritten Phase wird die beste Lösung gemäß der TOPSIS (technique for preference by similarity to the ideal solution)-Methode ausgewählt. Dieses Akronym steht für eine multikriterielle Präferenz ausdrückende Lösungsmethode, welche den*die Anwender*in unterstützt die bestmögliche Lösung aus einer begrenzten Lösungsmenge auszuwählen, wobei gleichzeitig der Abstand zu einem idealen Punkt minimiert und die Distanz zu einem Nadir Punkt maximiert wird (vgl. Karimi-Nasab und Konstantaras 2012, S. 479-490).

Olson vergleicht die Methodik für mehrere Gewichtungen bzw. Abstandsnormen (L_1 , L_2 , und L_∞) um die Distanz zu einem idealen bzw. Nadir-Vektor bewerten zu können. Die TOPSIS-Idee beruht auf den folgenden Schritten (vgl. Olson 2004, S. 722): (1) Lösungsnormierung bzw. Skalierung, (2) Entwicklung von Gewichtungs-Sets, (3) Identifikation der idealen bzw. Nadir-Alternative, (4) Entwicklung einer Distanzbewertung für die ideale Lösung D^+ bzw. die Nadir-Lösung D^- , (5) basierend auf einem definierten Verhältnis $R = \frac{D^-}{D^-+D^+}$ die gereichte Ermittlung möglicher Lösungsalternativen (vgl. Olson 2004, S. 721-727).

Karimi-Nasab hält abschließend fest, dass die vorgestellte problemspezifisch implementierte Random Search Heuristik für ein reales Problem einem MOGA hinsichtlich ihrer Effizienz, insbesondere für kleine bis mittelgroße Instanzen, überlegen ist. In großen Instanzen ist die Performance der Heuristik mit dem MOGA vergleichbar, wobei der MOGA einige prinzipielle Vorteile aufweist. Der MOGA hat (a) bestimmte Mechanismen, um sich neue Nachbarschaften durch die Kombination von Elementen aus bestehenden Lösungen zu generieren und arbeitet (b) populationsbasiert verglichen mit der einzellösungsbasierten Suche der Heuristik (vgl. Karimi-Nasab und Konstantaras 2012, S. 479-490).

3.4.5.4 Advance Order Production Smoothing Heuristic

Kogan und Tell stellen eine problemspezifische Heuristik vor, welche darauf beruht, Aufträge in kapazitiv weniger ausgelastete Perioden vorzuzuschieben und somit in diesem Zusammenhang auch die Kapazitätsauslastung zu glätten (vgl. Kogan und Tell 2009, S. 223-231). Dieser heuristische Ansatz verfolgt das Maximum-Prinzip, welches besagt, dass der Maximalwert einer Funktion an der Grenze des Definitionsbereichs auftritt (vgl. Danet 2013, S. 1ff.). Kogan und Tell modellieren den dynamischen Produktionsprozess über die folgende Gleichung: $\dot{X}(t) = u(t) - a(t)d$, wobei $X(t)$ den Überbestand sowie $u(t)$ die (nach oben durch die Kapazitätsgrenze $U(t)$ beschränkte) Produktionsrate zum Zeitpunkt t darstellen. Dieser Prozess wird einerseits durch die Produktionsrate, die durch die vordefinierte Maximumkapazität beschränkt ist, definiert und andererseits über den Initialbestand $X(0)$. Der Initialbestand ist an die Vorverschiebung von Produktionsaufträgen gebunden, welche dadurch bereits früher für den Verkauf verfügbar sind. Die Gleichung beschreibt den Materialfluss durch das Produktionssystem, wobei die Differenz zwischen der kumulierten Produktion und Nachfrage das Überschussniveau darstellt. Das Ziel besteht darin für das System geeignete Produktionsgeschwindigkeiten zu ermitteln sowie die optimale Anzahl an vorzuzusetzenden Produktionsaufträgen, sodass dabei die Gesamtkosten gemäß den einzuhaltenden Restriktionen minimiert werden. Das Modell berücksichtigt dabei Pönalen für nicht (zeitgerecht) ausgelieferte Aufträge sowie allgemeine Lagerbestandshaltungskosten in Anlehnung an Abb. 2.5. Mit Hilfe des Maximumsprinzip wird das Optimierungsproblem auf das Lösen nichtlinearer Gleichungen in Bezug auf Produktionsumschaltunkte sowie optimale vorverschobene Produktionsmengen reduziert und ist kosteneffizient und leicht zu implementieren. Mittels des Excel-Solvers wurden die resultierenden Gleichungen auf Basis eines industriellen Fallbeispiels aus der Bekleidungsindustrie gelöst, sodass sich für $X(0) = 53.78$ pro 1000 Anteile, d.h. 5.4%, als optimale vorversetzte Auftragsmenge ergibt, sowie eine Produktionsrate $u(t) = 16$ für $(3.36 \leq t < 83.18)$ auf Basis eines Zeitstrahls mit 100 Zeiteinheiten. An den jeweiligen Rändern ergibt sich $u(t) = 0$. (vgl. Kogan und Tell 2009, S. 223-231).

Dieser Ansatz ist um eine Betrachtung einer fixen Kapazität mit und ohne Überstunden sowie eines optimierten Bestellwesens als auch mit kleinstmöglichen Bestellungen (auf Basis einer Hersteller-Lieferantenbeziehung) erweiterbar. Die vorhandenen Modelle betrachten bereits Produktions- und Lagerkosten. Synáková schlussfolgert, dass sich sowohl bei dem Modell ohne als auch mit Überstunden ein möglichst hoher Personalfixkostenanteil kostenminimierend auf die Supply-Chain auswirkt. Diese Methoden vernachlässigen jedoch eine sorgfältige Modellierung und beschränken sich anstelle einer dynamischen Kapazitätsbetrachtung auf ein statisches Modell. Außerdem vermissen beide Modelle eine systematische Losteilung und verschieben stattdessen nur ganze Aufträge, ohne auf eine gleichmäßige, periodische Auftragsvoluminaverteilung einzugehen (vgl. Synáková 2017, S. 117-133).

3.4.5.5 Line Balancing-Ansätze

Line Balancing-Ansätze werden v.a. über den *Smoothing Index* (SI) bewertet, welcher als die Quadratwurzel aus der Summe der quadrierten Differenzen zwischen der maximalen Prozesszeit an einer Arbeitsstation und der jeweils anfallenden Stationsauslastung definiert ist (vgl. Lingitz et al. 2019, S. 1433).

Die Zykluszeit einer Montagelinie ist durch die gewünschte Produktionsrate vorbestimmt. Damit die gewünschte Menge an Endprodukten innerhalb einer bestimmten Zeitspanne produziert wird, gibt es drei Möglichkeiten, um die zeitliche Ausführung von Aufgaben innerhalb der Zykluszeit zu verbessern (vgl. Zupan und Herakovic 2015, S. 2306ff.):

1. Umverteilung bzw. Neuzuweisung von bestimmten Aufgaben an den vor- oder nachgelagerten Prozessschritt.
2. Verbesserung der Arbeitsmethoden und daher Optimierung der Stückzeit t_{ei} .
3. Einführung von parallelen Arbeitsplätzen und dadurch Aufteilung der Operation auf (möglichst) identische Jobs.

Nach der initialen, diskreten Modellierung einer Produktionslinie werden diverse Verbesserungen (zusätzliche Gabelstapler, Anpassung der Ankunftsrate von Rohmaterialien) im Zuge von *Was-Wäre-Wenn-Szenarien* schrittweise mittels DES durchsimuliert und in Bezug auf die Kriterien Durchsatz, Linienauslastung und Kosten bewertet. Auf diese Art und Weise können für die jeweilige Linienkonfiguration die dazugehörigen KPI (Zykluszeiten, Wartezeiten, Durchsatz) ermittelt werden. Dadurch konnte anhand eines realen Produktionssystems die Produktivität durch eine Vierteilung der Zykluszeit entsprechend gesteigert werden, während die Wartezeiten und notwendigen Pufferplätze nahezu vollständig reduziert wurden. Allerdings findet bei ausschließlich simulationsbasierter Optimierung auf Basis der Szenarienanalyse keine gezielte Optimierung der Produktionsplanung statt, sondern viel mehr eine rein strukturelle Anpassung des Produktionssystems in Bezug auf seine Dimensionierung bzw. Ressourcennutzung (Rohmaterialien, Pufferplätze, Gabelstapler) (vgl. Zupan und Herakovic 2015, S. 2305-2311).

Edis verwendet für ein disassembly line balancing and sequencing problem, also die Glättung und Reihenfolgeoptimierung einer Demontagelinie, einen Constraint Programming Ansatz, welcher sich v.a. für stark beschränkte Probleme eignet. Für die initiale Erzeugung einer gültigen Reihenfolge wird dabei eine fixe Zuordnung von Aufträgen an die jeweiligen Arbeitsstationen im Modell vorgenommen. Die Zielfunktion besteht aus 5 Teilzielen, welche lexikographisch optimiert werden. Charakteristisch für derartige Probleme glättet ein Teilziel die anfallende Arbeitslast unter den verschiedenen Arbeitsstationen. Ein weiteres Teilziel versucht gezielt Teile mit höherem Verbrauch an den Beginn der Auftragsreihenfolge zu platzieren. Dieses Teilziel hat eine starke Ähnlichkeit zu der Ausnutzung von Holland's Schema-Theorem bei einem GA, wenn im Zuge der Optimierung Schemata mit geringer Ordnung erzeugt werden. Rechnergestützte Studien zeigen, dass die Methode mittels CPLEX-Solver für bis zu 35 Demontage-Aufgaben effizient optimale Lösungen auffinden kann. Edis zeigt außerdem, dass für knapp 90% dieser Probleme Metaheuristiken bzw. heuristische Methoden zum Einsatz gelangen, während lediglich 4 Publikationen exakte Lösungsansätze vorstellen (vgl. Edis 2021, S. 1ff.). Lingitz demonstriert einen exakten Lösungsalgorithmus auf Basis einer Kombination aus binärer Programmierung sowie

Constraint Programming mit einem *Balancing Index* (BI), welcher die Lastverteilung an den einzelnen Arbeitsstationen bewertet. Dieser Index kann als Basis für die Bildung eines geeigneten Indikators für einen Optimierungsalgorithmus verwendet werden, sich dieser auf die Betrachtung der unterschiedlichen Auslastung an den jeweiligen Arbeitsstationen beschränkt (vgl. Lingitz et al. 2019, S. 1433; Whitley und Sutton 2012, S. 637-671).

3.4.5.6 Planung flexibler Produktionskapazitäten

Grundmann betrachtet flexible Produktionskapazitäten hinsichtlich der Definition und Planung sowie des Kapazitätseinsatzes aus einer kombinierten Logistik- und Kostensicht. Im Zuge dessen wird ein zweistufiger Algorithmus zur Anpassung der Kapazitäten unter Priorisierung der Kapazitätsoptionen auf Basis der zugehörigen Kostensätze entwickelt. Dadurch wird eine hohe Termintreue bei einer verbesserten Auslastungssituation kostengünstig realisiert wird. (vgl. Sven Grundmann 2007, S. IV u. S. 68).

Das in MS Excel definierte Optimierungsmodell wurde mit dem dazugehörigen GA (16 Chromosomen pro Population, Abbruch nach max. 250 Generationen) optimiert. Die GA-Ergebnisse liefern optimierte Werte für den mittleren internen und externen Bestand (Std.) sowie die mittlere interne und externe Monatsleistung (Std./Monat), den resultierenden Kostensatz in diesem Monat (€/Std.), und die Soll-Leistung (Std./Monat) (vgl. Sven Grundmann 2007, S. 125). Die Methode hat den Kapazitätsbedarf im Zeitraum um 17% bzw. die Reichweite um 18% reduziert, während die Auslastung um 4% erhöht werden konnte (vgl. Sven Grundmann 2007, S. 129).

Die entwickelte Methode richtet sich an der operativen Produktionsplanung aus und prüft den Nutzen flexibler Produktionskapazitäten. Der GA definiert einen geeigneten Betriebspunkt. Die Grenzen der Methode liegen v.a. in der Qualität der zur Verfügung stehenden Daten, wodurch sich eine Übertragung des Anwendungsbeispiels mit relativ verlässlicher Planung in die Konsumgüterindustrie herausfordernd gestaltet (vgl. Sven Grundmann 2007, S. 144f.). Für die in dieser Arbeit zu entwickelnde Methode ist die Methodik von Grundmann nicht anwendbar, weil ein variabilisierbarer Produktionsprozess modelliert wird und flexible Produktionskapazitäten nicht betrachtet werden. Die inhaltliche Nähe zum Thema Produktionsglättung ist durch die Betrachtung flexibler Produktionskapazitäten gegeben, die Zielfunktion ist (zeitlich) zu weit entfernt von den Anforderungen an eine rollierende Glättungsmethode für die operative Produktionsplanung.

3.4.5.7 Datengestützte evolutionäre Optimierung

Datengestützte, evolutionäre Optimierung besteht aus der Kombination von evolutionärer Optimierung (GA, EA, etc.) mit Machine Learning Methoden und Datenwissenschaften (vgl. Jin et al. 2019, S. 442-458). Dabei werden historische Daten gesammelt bzw. in Simulationsexperimenten aufbereitet und anschließend z.B. direkt (oder aufwändiger indirekt) verarbeitet, sodass diese in Form eines trainierten Ersatzmodells (engl.: surrogate model) für die Approximation der Zielfunktion bzw. Nebenbedingungen nutzbar vorliegen. Weiters wird zwischen offline nutzbaren und online vorliegenden Daten unterschieden. Während Offlinedaten nur vorab der Optimierung für die Erstellung von Ersatzmodellen verarbeitet werden, erhöhen online datengestützte EA die Flexibilität und Performance der Optimierung. Die entsprechenden Modelle werden z.B. im Falle des Vorliegens von Onlinedaten während der Optimierung laufend aktualisiert, verfeinert und dadurch zuverlässiger (vgl. Jin et al. 2019, S. 442-458). Die Konzepte sind mächtig und vielseitig einsetzbar. Dementsprechend erfolgt eine Abwägung, ob in dieser Arbeit gesammelte Daten offline oder online verwendet werden.

3.4.6 Fazit

Für den vorliegenden Anwendungsfall ist die Kombination des Optimierungsverfahrens mit einer vereinfachten Simulation zu einer sogenannten Simheuristik sinnvoll, wenn eines der definierten Teilziele den Einsatz von Simulation erfordert. Die Simulation würde dabei in Form der Variante als

Bewertungsfunktion der Optimierungsaufgabe integriert werden. Die Zweckmäßigkeit des Einsatzes von Simulation wird in Kapitel 4 entschieden. Spezifische Strategien, insbesondere die Berücksichtigung einer Variation der Gewichtungen in einem beschränkt erlaubten Wertebereich, werden mit in die Verfahrensentwicklung aufgenommen. Das Konzept des Smoothing Index aus 3.4.5.5. ist originell. Anstelle des SI wird in Kapitel 4, der Methodenentwicklung dieser Arbeit, ein Zielfunktionssystem aufgestellt, welches die Gleichmäßigkeit der Produktionsausstoßrate evaluiert.

3.5 Rollierende Planungsansätze

Sampaio beschreibt in seiner Arbeit, dass bestehende rollierende Ansätze für die Produktionsplanung einerseits nicht befriedigend mit Kapazitätsrestriktionen umgehen und andererseits den gesamten Planungshorizont betrachten bzw. optimieren, obwohl nur die jeweils zeitlich nächsten Planungsperioden von Interesse sind. Der Planungshorizont reduziert sich durch die Anwendung eines sogenannten *Zerlegungsschemas*²¹ jedes Mal um eine Periode, wenn eine Periode ausgeführt wird. Die Planungsabteilung interessiert der Produktionsplan nur bis zu einem bestimmten, fixierten Horizont im Voraus, wobei sich dieser Horizont (iterativ) nach Ablauf wieder um eine bestimmte Periodenlänge verlängert. Sein Modell koppelt ein lineares Optimierungsmodell mit einer externen *Übertragungsfunktion*²² (vgl. Selçuk et al. 2008, S. 206ff.), um die nichtlineare Variabilität des Produktionsdurchsatzes mit der Arbeitsauslastung und den Durchlaufzeiten des Systems darzustellen. Dieser dynamische Ansatz inkludiert, dass das Produktionsniveau x_{ij} für jedes Produkt i und jede Periode j geringer sein muss als die verfügbare Produktionskapazität γ_{ij} für jeden Artikel i und jede Planungsperiode j , d.h.: $x_{ij} \leq \gamma_{ij}$. Die formale Beschreibung lässt auf den Einsatz dynamischer Nebenbedingungen zurückschließen, allerdings fehlt die entsprechende Beschreibung im Umgang mit dynamischen Nebenbedingungen (vgl. Sampaio et al. 2017, S. 31f.).

Das vorgestellte Zerlegungsschema für rollierende Planung ist für die Anforderungen dieser Arbeit unzureichend, weil die zu entwickelnde Methodik Produktionsmengen entsprechend aufteilen bzw. um eine bestimmte Anzahl an Perioden vorverschieben muss. Außerdem ist ein fixierter Planungshorizont für den (mittelfristig agierenden) Einkauf von Nachteil, wenn sich der jeweilige Planungshorizont seinem Ende nähert.

Fazit: Prinzipiell ist ausschließlich die nähere Zukunft unmittelbar planungsrelevant. Je nach Art der Methodik (z.B. in Abhängigkeit vom Arbeitsplan bzw. der Durchlaufzeit) beeinflusst die nähere Zukunft auch die mittlere Zukunft, sodass der Betrachtungshorizont auf die mittlere Zukunft ausgedehnt wird. Hier wird im Zuge der Entwicklung der Methodik ein Kompromiss gefunden werden, wie viele Perioden in die Zukunft geplant wird. Die rollierende Planung erlaubt den Einsatz von Data-Mining, wenn z.B. bestimmte Teile von Stammdaten und der dazugehörige Zielerreichungsgrad laufend mitgespeichert werden. Die in dieser Arbeit entwickelte Methodik nutzt diesen Umstand aus und betrachtet das sich daraus bietende Potential.

3.6 Klassische und moderne kapazitive Losgrößenprobleme

Das klassische Losgrößenproblem bzw. das *Single-level Lot-sizing Model* (SLLSM) berücksichtigt Losgrößenentscheidungen für einen einzelnen Produktionsartikel über einen Planungshorizont, der aus vielen Planungsperioden besteht. Das Ziel besteht in der Suche eines Produktionsplans, welcher die Lager- und Rüstkosten gleichermaßen *heuristisch* unter der Bedingung minimiert, dass die Verbrauchsanforderungen jeder Periode erfüllt sind (vgl. Stadtler 2000, S. 320ff.). Das kapazitiv beschränkte Losgrößenproblem, auch bekannt als *capacitated lot sizing problem* (CLSP), minimiert die Gesamtkosten, klassisch bestehend aus Rüst- und Lagerkosten, gegen eine periodenweise beschränkte Kapazität. Nachdem eine

²¹engl.: decomposition scheme

²²engl.: clearing function

detaillierte Reihenfolgeplanung nicht in der Problemstellung enthalten ist, wird zuerst häufig das CLSP und konsekutiv das aus dem CLSP resultierende Reihenfolgeproblem gelöst. Neben problemspezifischen Heuristiken werden für derartige Probleme v.a. auch Metaheuristiken (GA, SA) angewendet. (vgl. Özdamar und Bozyel 2000, S. 1043-1057; Céline Gicquel et al. 2008, S.4-16; Masmoudi et al. 2016, S. 301-306).

Das Verfahren von Wagner-Whitin ist hingegen ein *exaktes* Lösungsverfahren zur Bestimmung der optimalen Losgröße für ein Produkt, das unter dynamischer Nachfrage bei einstufiger Fertigung ohne Berücksichtigung von Kapazitätsrestriktionen hergestellt wird. Stadtler beschreibt eine Modifikation des klassischen SLLSM, wobei Absatzprognosen außerhalb des Planungshorizonts in die Entscheidungsfindung miteinfließen. Sein Modell benötigt einen relativ umfangreichen Planungshorizont und (idealerweise) nahezu perfekte Absatzpläne, um kompetitiv zu sein. Außerhalb des Planungshorizonts wird nur ein relativ grobes Absatzplanmengengerüst benötigt, um bekannte Heuristiken, wie z.B. von Silver and Meal bzw. Groff, zu übertreffen. Das Prinzip von Stadtler, die Verbesserung rollierender Planungspläne durch Berücksichtigung nur eines Teils der Fixkosten, die mit einer Entscheidung verbunden sind, die sich auf Perioden jenseits des Planungshorizonts auswirken, kann auf eine Vielzahl von Entscheidungsmodellen angewendet werden. Dies entspricht dem Prinzip einer regelbasiert-rollierend vorausschauenden Planungslogik (vgl. Wagner und Whitin 2004, S.1770ff.; Stadtler 2000, S. 318-326).

Moderne Ansätze beschreiben auch kapazitätsbeschränkte, teils kombinierte Losgrößen- und Reihenfolgeplanungsprobleme, wie z.B. die heuristische 2-Phasen-Methodik von Marinelli (vgl. Marinelli et al. 2007, S. 177ff.). Dabei wird initial eine heuristische Zuordnung auf ein Gemenge aus Puffern durchgeführt. Anschließend werden die Aufträge an den jeweiligen Produktionslinien mittels lokaler Suchverfahren, siehe 3.2.2, gereiht. Das heuristische Optimierungsverfahren, welches im CPLEX-Solver modelliert und gelöst worden ist, erreicht nahezu optimale Lösungen, welche in relativ kurzer Zeit erzielt werden. Im Zuge der Arbeit werden diverse Relaxationen, z.B. die Nicht-Berücksichtigung von Rüstzeiten für Produktionslinien oder Puffertanks, untersucht. Im Zuge des Reihenfolgeproblems hat man sich bei der Suche auf den *shift-move* bzw. *swap-move* konzentriert. Während der *swap-move* (vgl. Abb. 3.7) im Zuge von GA diskutiert wird, transferiert bzw. überträgt der *shift-move* definierte Lösungsteile auf eine andere neue Lösung (vgl. Nickel et al. 2020, S. 9-23).

Absi und Kedad-Sidhoum beschreiben ein kapazitiv beschränktes Losgrößenproblem²³ mit real in der Produktion anfallenden Rüstzeiten. Die Autoren empfehlen eine gemischte, ganzzahlige Optimierungsheuristik auf der Grundlage einer strategischen Zerlegung des Planungshorizonts in drei Teile (*Eingefrorene Zone*, *Entscheidungsfenster*, *Approximationsfenster*), um machbare Lösungen zu finden. Die gewichtete Summen-Zielfunktion minimiert die Gesamtkosten bestehend aus Produktionskosten, Lagerhaltungskosten, Pönalen und Kürzungen, Abweichungen vom Sicherheitsbestand und Rüstkosten. Die Nebenbedingungen fordern die Erhaltung des Materialflusses durch die Sicherstellung einer bestimmten Mindestproduktionsmenge auf Artekelebene, wobei das Gesamtproduktionsniveau unter der Maximalkapazität bleibt. Das Prinzip der Horizontzerlegung besteht darin, die Integritätsbeschränkungen der binären Variablen, die zu einem festen Abschnitt des Horizonts gehören, beizubehalten und diese Beschränkungen für die anderen Variablen aufzulockern. Das Vorgehen besteht aus der Hybridisierung der BaB-Methode mit der Horizontzerlegung. Während in der eingefrorenen Zone die Werte teilweise oder zur Gänze fest zugewiesen und im Approximationsfenster Variablen oder Nebenbedingungen relaxiert werden, berücksichtigt das zentrale Entscheidungsfenster das originale Modell. Dadurch verringern sich einerseits die Rechenzeiten, andererseits wird der Suchbaum entsprechend reduziert. Der Produktionsplan wird durch eine iterative zeitlich vorwärts gerichtete Strategie bestimmt. Im Approximationsfenster werden die Nebenbedingungen derart gelockert, sodass prinzipiell ungültige Lösungen entstehen, welche dann als untere Kapazitätsschranke agieren. Danach werden 2 Heuristiken vorgestellt (Fix-and-Relax, Double-Fix-and-Relax) und untereinander verglichen, die sich in der Art und Weise unterscheiden, wie Rüst- und Produktionsvariablen jeweils eingefroren bzw. relaxiert werden (vgl. Absi und Kedad-Sidhoum 2007, S. 171-192).

Mercé stellt eine dem von Absi und Kedad-Sidhoum ähnliche MIP-basierte Heuristik vor, welche auf einanderfolgenden Auflösungen von verkleinerten MIP-Modellen basiert. Die Reduktion erfolgt

²³engl.: multi-item capacitated lot-sizing problem

durch eine sogenannte Einfrierstrategie sowie eine Vereinfachungsstrategie. Die iterative Prozedur selbst beruht auf hauptsächlich 2 Parametern, der Neuterminierungsperiodizität sowie der Länge des zentralen, detailliert geplanten Ausschnitts aus den drei Sektionen (Startsektion, Zentralsektion und Endsektion). Das vollständig dargestellte MIP-Optimierungsmodell wird nur im detailliert geplanten Ausschnitt berücksichtigt. Die Einfrierstrategie beruht auf der Reduktion der Anzahl an frei wählbaren Entscheidungsvariablen in der Startsektion des Planungshorizonts. Durch die Vereinfachung werden einige Details (Rüstkosten und binäre Variablen) in dieser Sektion ignoriert. Im generischen Ansatz können neue Charakteristika mit in die MIP-Formulierung mit aufgenommen werden, ohne dabei den Algorithmus zu verändern (vgl. Mercé und Fontan 2003, S. 97-111). Die Modellvereinfachungen in der Startsektion sind für das vorliegende Problem jedoch nicht zulässig.

Suwondo stellt eine umfassende Übersicht über dynamische Losgrößenprobleme und entsprechende Lösungsheuristiken zur Verfügung (vgl. E. Suwondo und H. Yulianto 2012, S. 36-49). Er hält tabellarisch fest, dass neben einfachen (Konstruktions- und Glättungs) Heuristiken (vgl. Minner 2009, S. 305-310) häufig Metaheuristiken (v.a. GA) als robuste Lösungsverfahren für diverse kapazitive Losgrößenprobleme eingesetzt werden (vgl. Gaafar 2006, S. 433-444). Nguyen vergleicht einen GA mit veränderlichen Ausführungsraten (für die einzelnen Operatoren) mit der (Greedy-Ansatz basierten) Silver-Meal Heuristik und unterstreicht die Überlegenheit des GA (vgl. van Hop und Tabucanon 2005, S. 129-135).

Fazit: Das Prinzip der Zerlegung des Planungshorizonts in verschiedene Teile wird im Zuge der methodischen Entwicklung des Verfahrens berücksichtigt. Der Ansatz von Mercé und Fontan in Bezug auf die Reduktion frei wählbarer Entscheidungsvariablen beinhaltet eine Reihe an Schlüsselementen eines dynamischen Planungsprozesses: Eine detaillierte Planung erfolgt in der zentralen Sektion, während die Startphase teilweise als fixierter Bereich modelliert wird und die Endphase relativ geringe Auswirkungen auf den Produktionsplan hat. Für den konkreten Anwendungsfall ist auf Ebene der Teilziele zu klären, in welchem Zeitfenster sich die zentrale Sektion befindet. Eine weitere Schlussfolgerung v.a. im Bereich der rollierenden Planung ist, dass der am weitesten in der Zukunft liegende Teil des Planungshorizonts vereinfacht modelliert werden kann. Dieser ist weder planungsrelevant, außerdem ändert sich die Planung mehrmals, bis dieser Teil der Planung in das relevante Zeitfenster vorrückt (vgl. Absi und Kedad-Sidhoum 2007, S. 171-192; Mercé und Fontan 2003, S. 97-111).

3.7 Umgang mit Nebenbedingungen

Die meisten realen Probleme weisen bestimmte einzuhaltende Nebenbedingungen auf. Konak, Coit und Smith empfehlen vier verschiedene Strategien zur Handhabung von Einschränkungen im Zuge der Optimierung (vgl. Konak et al. 2006, S. 1003f.):

1. Das Verwerfen von undurchführbaren Lösungen.
2. Die Verringerung der Eignung undurchführbarer Lösungen durch Verwendung einer entsprechenden Straffunktion (englisch: *penalty function*).
3. Die Anpassung der genetischen Operatoren in der Form, sodass sie immer durchführbare Lösungen produzieren.
4. Die Reparatur nicht durchführbarer Lösungen (englisch: *repair mechanisms*).

Die Autor*innen halten fest, dass die Strategien 1, 3 und 4 in der multikriteriellen Optimierung direkt anwendbar sind. Im Zuge der Umsetzung entsprechender Straffunktionen, welche die am häufigsten verwendete Strategie zur Handhabung von Einschränkungen bei einem Ein-Ziel-GA darstellen, ist die Implementierung für einen multikriteriellen GA nicht-trivial. Das liegt daran, dass die Fitnesszuweisung häufig auf dem Nicht-Dominanz-Rang einer Lösung geschieht und nicht auf dessen tatsächlichen Zielfunktionswert. Im Beitrag wird eine *Nischenselektionsstrategie* (siehe 3.3.1.1) erläutert, welche den Umgang mit nichtdurchführbaren Lösungen thematisiert (vgl. Konak et al. 2006, S. 997-1007):

1. Zuerst werden nach dem Zufallsprinzip zwei Lösungen x und y aus der Bevölkerung ausgewählt.

2. Wenn eine der Lösungen machbar und die andere nicht durchführbar ist, wird die machbare Lösung als Gewinner titulierte und abgebrochen. Wenn beide Lösungen undurchführbar sind, wird bei Schritt 3 fortgesetzt, ansonsten bei Schritt 4.
3. In diesem Fall sind die Lösungen x und y beide nicht durchführbar: Man wähle eine zufällige Referenzmenge C unter den in der Population nicht realisierbaren Lösungen. Dann werden die Lösungen x und y mit Lösungen der Referenz C hinsichtlich des Grades der Undurchführbarkeit verglichen. Um dies zu erreichen, wird ein Maß für die Undurchführbarkeit (z.B. die Anzahl verletzter Beschränkungen, die Höhe der Beschränkungsverletzung etc.) für die Lösungen x und y und in der Menge C berechnet. Wenn eine der Lösungen x und y besser (bzw. die andere schlechter) als die beste Lösung in C in Bezug auf das berechnete Machbarkeitsmaß ist, dann ist der Gewinner die weniger verletzende Lösung. Wenn es jedoch ein Unentschieden²⁴ gibt und die beiden Lösungen x und y weder besser noch schlechter als die beste Lösung in C sind, dann zählt ihre Nischenzahl für die Selektion. In diesem Fall wird die Lösung mit der geringeren Nischenzahl als Gewinner ausgewählt.
4. In diesem Fall sind die Lösungen x und y beide durchführbar. Man wählt eine zufällige Referenzmenge C in der Population unter den machbaren Lösungen aus. Dann werden die Lösungen x und y mit den Lösungen der Menge C verglichen. Wenn eine von ihnen in der Menge C nicht dominiert und die andere von zumindest einer Lösung aus C dominiert wird, ist die erste Lösung der Gewinner. Im Falle eines Unentschieden wird die Nischenanzahl der Lösungen im Raum der Entscheidungsvariablen berechnet. Die Lösung mit der kleineren Nischenanzahl wird ausgewählt.

Diese Strategie ist für die Arbeit von Bedeutung: Sie fordert nämlich prinzipiell dazu auf, eine aktuelle Lösung mit einer Lösungs-Referenz (z.B. einer erhaltenen Ausgangslösung) zu vergleichen, und die aktuelle Lösung in Bezug auf den verminderten Grad des Verstoßes gegenüber einer Nebenbedingung auszuwählen bzw. andernfalls nicht. Dieses Konzept gewinnt insbesondere in der Praxis an Bedeutung, wenn z.B. eine Ausgangslösung bereits gegen die vom Zielfunktionssystem geforderte Einhaltung der Nebenbedingung verstößt. Dies ist unter besonderen Planungsumständen zulässig, dass z.B. in einer Periode bewusst über die maximal definierte, zulässige Kapazität produziert wird. Diese Regel leitet ein Konzept ab, dass eine Lösung als machbarer bewertet wird, wenn diese z.B. weniger stark gegen eine Nebenbedingung verstößt als die erhaltene Basislösung bzw. eine vorherige Lösung.

3.8 Bewertung und Vergleich von Optimierungsverfahren

Zur Beurteilung eines Algorithmus werden allg. sechs Kriterien herangezogen (vgl. Sand 2013, S. 43f.):

1. Die Qualität der besten durch den Algorithmus gefundenen Lösung.
2. Die algorithmisch notwendige Laufzeit für das Auffinden der besten Lösung.
3. Die algorithmisch notwendige Laufzeit für das Auffinden guter Lösungen.
4. Die Robustheit des Verfahrens.
5. Die (Euklidische oder Hamming-) Distanz der besten Lösung zu guten Lösungen.
6. Der Trade-Off zwischen Gültigkeit und Lösungsgüte.

Diese Charakteristika beziehen sich allesamt auf die Lösungsgüte bzw. auf die Laufzeit. Allgemein wird die Laufzeitkomplexität auf Basis der Landau-Notation des ausgewählten Verfahrens sowie die Lösungsgüte $q(x)$ durch die Abweichung der durch die Metaheuristik erreichte Lösung $L(x_{heu})$ vom Optimum $L(x_{opt})$ bewertet (vgl. Merz 2003, F. 48):

$$q(x) = \left(\frac{L(x_{heu})}{L(x_{opt})} - 1 \right) \cdot 100\% \quad (3.14)$$

²⁴engl.: Draw

Dieser Wert kann nur ermittelt werden, wenn die optimale Lösung auch bekannt ist. Dies ist i.d.R. bei realen Problemen nicht der Fall, sodass dieser Wert zumeist nur bei theoretischen bzw. bekannten Problem instanzen als Vergleichsmaßstab herangezogen wird. Sand erwähnt, dass die Einordnung der Lösungsgüte ohne Vorliegen einer Optimallösung einerseits über obere bzw. untere Schranken ermöglicht wird. Andererseits kann ein Vergleich der Lösungsgüte des zu begutachtenden Verfahrens mit der Lösungsgüte der bisher besten bekannten Lösung erfolgen, wobei die bisher beste gefundene Lösung als Schranke angesehen wird. Bezüglich eines Vergleichs der Laufzeit empfiehlt es sich, die zu vergleichenden Verfahren in derselben Programmiersprache zu schreiben, mit dem gleichen Compiler in der gleichen Entwicklungsumgebung zu übersetzen und auf demselben Rechner mit den gleichen Problem instanzen auszuführen (vgl. Silberholz und Golden 2010, S. 635; Sand 2013, S. 44f.).

Im Zuge dieser Arbeit werden diese Regeln für den Vergleich von Optimierungsverfahren berücksichtigt, wobei sich der Vergleich auf die Kriterien (1), (2) und (4) beschränkt. Die Kriterien (3) und (4) können auf die Robustheit für das Auffinden guter Lösungen auf ein Kriterium (4) zusammengefasst werden, weil sich der*die Entscheidungsträger*in nicht für gute Lösungen interessiert, sondern nur für die jeweils beste Lösung. Die Robustheit des Verfahrens ist speziell für eine rollierende Anwendung, die im Zuge der vorliegenden Problemstellung vorliegt, wichtig, weil der Algorithmus jede Periode auf einen neuen bzw. aktualisierten Datensatz verlässlich funktionieren muss. Das Kriterium (6) ist nicht von Bedeutung, weil nur gültige Lösungen gemäß Zielfunktionssystem erstellt bzw. von der Produktionsplanung berücksichtigt werden. Die Beurteilung in dieser Arbeit konzentriert sich also auf die Güte der besten gefundenen Lösung, die Laufzeitkomplexität gemäß dem Verfahren nach Landau sowie die Robustheit der zu vergleichenden Verfahren. Je nach Verfahren kann die Speicherkomplexität anhand des notwendigen Speicherbedarfs direkt in der Entwicklungsumgebung ausgewertet werden, dies ist für die vorliegende Problemstellung nicht relevant.

Für die Evaluierung der Eignung von Optimierungsverfahren für multimodale Fitnesslandschaften existieren Testfunktionen, z.B. die nicht-konvexe stark multimodale *Rastrigin*-Funktion, die durch ein bananenförmiges Tal charakterisierte *Rosenbrock*-Funktion, die von Kalyanmoy Deb entwickelte *Deb2*-Testfunktion basierend auf der Himmelblau-Funktion, die *Alpental*-Funktion oder auch die *Miner-S-1* Funktion mit vielen, unregelmäßig verteilten lokalen Minima (vgl. Kux 2011, S. 68ff.).

Derartige Testfunktionen eignen sich sehr gut für den Vergleich von Optimierungsverfahren in Bezug auf diese mathematischen Funktionen. Allerdings sind diese Testfunktionen nicht für das vorliegende reale Optimierungsproblem geeignet. Im Gegensatz zur realen Aufgabenstellung verfügen die genannten Testfunktionen einerseits nicht über einen booleschen Anteil, darüber hinaus sind die heuristisch verarbeiteten Informationen einer hybriden Optimierung nicht auf diese Funktionen umlegbar.

3.9 Diskussion

Im Rahmen der Diskussion werden weitere Ansätze und Methoden, welche für diese Arbeit die größte thematische Relevanz in Bezug auf das Thema Produktionsglättung aufweisen, ergänzend reflektiert. Die auf diese Art und Weise gefilterten und in diesem Abschnitt behandelten Methoden werden anschließend auf Basis ihrer charakteristischen Eignung für die Nivellierung von Produktionssystemen gewichtet in einer Tabelle gegenübergestellt. Aus den Mängeln leitet sich der konkrete Forschungsbedarf für die methodische Entwicklung in dieser Arbeit ab.

Der Ansatz von *Rewers* beinhaltet eine Aufteilung der Produktionsartikel in Gruppen, welche die technologische Ähnlichkeit abbilden. Darauf basierend werden die Prozesszeiten, die benötigten Ressourcen und das Mengengerüst gebildet. Danach werden die Verbräuche diesen Gruppen zugeordnet, um einen sogenannten Abgleich zu erhalten. Im nächsten Schritt werden im Zuge der Wertstromanalyse die Produktionszeiten verifiziert und ggf. aktualisiert. Dies beinhaltet die Analyse von Rüst- und Wartezeiten im Produktionssystem sowie Produktions- und Transportkosten. Im Zuge der nächsten Phase werden die Losgrößen sowie die Häufigkeit und Reihenfolge der einzelnen Lose unter Berücksichtigung einer Kompetenzmatrix definiert. Abschließend wird ein gültiger Produktionsplan aufgesetzt.

Die sorgfältige Daten- und Wertstromanalyse liefert eine wichtige Grundlage, bevor mit der Modellierung und Optimierung begonnen wird. Obwohl die von Rewers vorgestellte Methode prinzipiell eine wichtige systematisch-methodische Basis darstellt, ist sie generisch, wenig konkret und unspezifisch und schneidet deshalb in Ermangelung einer spezifischen, algorithmischen Optimierungsmethode im Zuge des tabellarischen Vergleichs als schwächstes Verfahren ab (vgl. Rewers et al. 2019, S. 95ff.).

Al-Ashhab stellt ein Mixed-Integer Linear Programming (MILP) Modell für die Lösung seines MOMPMP Problems vor. Gelöst wird das Problem mittels des Fico-Xpress-Optimizers Solvers (vgl. FICO 2021). Das Modell beinhaltet eine mehrstufige Lieferkette und besteht aus mehreren Lieferant*innen, einem*einer Hersteller*in sowie mehreren Distributor*innen und Kund*innen. Dabei werden die Lokation der Lieferant*innen und die Liefermengen im Netzwerk entsprechend optimiert, wobei eine Gewinnmaximierung sowie eine Maximierung der Lieferquote unter den gegebenen kapazitiven und transporttechnischen Nebenbedingungen angestrebt wird. Die vorgestellte Methodik stellt eine möglichst optimale Supply-Chain-Konfiguration sicher und optimiert dadurch die Ausbringungsmenge und -rate im Netzwerk. Das Modell ist entsprechend umfangreich und hat auf Grund des Netzwerkcharakters nur bedingt Nähe zur vorliegenden Aufgabenstellung (vgl. Al-Ashhab 2016, S. 43ff.).

Karimi-Nasab verwendet für sein MOMPMP Problem einen GA, weil dieser in der Lage ist, Mehrzielprobleme effizient zu lösen. Die Optimierung sieht dabei v.a. den Erhalt eines geglätteten Produktionsplans in Folge der Berücksichtigung einer gleichmäßigen Produktionsausstoßrate als auch der Kostenminimierung dieses Produktionsplans vor. Für das Verhältnis von Kosten zu Zeit wird dabei ein linearer Zusammenhang angenommen. Die Abweichungen zwischen Absatz- und Produktionsrate werden quadratisch bewertet und minimiert. Sein Modell sieht im Sinne einer *realitätsnahen* Reihenfolgeplanung an der berücksichtigten Maschine *variable Prozessdurchlaufzeiten* in dem Sinne vor, dass die Maschine unterschiedliche Bearbeitungszeiten erhält und die Bearbeitungszeit eines Produkts an der Maschine damit variabel wird. In Bezug auf sein Modell wurden entsprechende Vereinfachungen getroffen, wenn diese in der realen Welt nicht von Bedeutung sind. Eine Schwachstelle im Modell besteht darin, dass Auftragsrückstand, resultierend in Fehlmengenkosten, in jeder Periode mit Ausnahme der letzten erlaubt ist (vgl. Karimi-Nasab und Aryanezhad 2011, S. 3596-3610). Dies trifft für das vorliegende Planungsproblem definitiv nicht zu.

Das Vorgehen von *Kogan und Tell*, siehe auch 3.4.5.4, beruht auf der Ausnutzung der zeitlichen Vorverschiebung von Aufträgen und ermittelt auf relativ einfache Art und Weise die Produktionsgeschwindigkeit sowie den Anteil der vorzuversetzenden Produktionsmengen. Die Kapazitätsgrenzen des Produktionssystems werden dazu verwendet, um Ungenauigkeiten in der Absatzplanung während den Absatzperioden auszugleichen. Das Ziel besteht darin, die Produktionsraten sowie die Kosten bzw. Menge an vorzuverschiebenden Aufträgen derart zu bestimmen, sodass unter Einhaltung sämtlicher Nebenbedingungen die zu erwartenden Gesamtkosten minimiert werden. Im Zuge dieses Ansatzes gilt es zu beachten, dass die Endprodukte früher fertiggestellt werden und dies mit einem höheren (Bestands-) Risiko verbunden ist. Diese Methodik bietet sich auch zur Bestimmung der vorzuverschiebenden Produktionsmengenanteile sowie ferner der Produktionsgeschwindigkeit an. Mit Zuhilfenahme des Maximumprinzips reduziert sich das Optimierungsproblem auf die Lösung nichtlinearer Gleichungen in Produktionsumschaltpunkten und die Ermittlung der optimalen vorzuverschiebenden Produktionsmengen. Für komplexere Produktionssysteme, welche z.B. verschiedene Produktionsgeschwindigkeiten in Teilen des Produktionssystems erfordern, ist diese Methodik nur bedingt anwendbar (vgl. Kogan und Tell 2009, S.225ff.). Außerdem wird die Losgrößenthematik bzw. die Glättung auf Produktionsartikelebene nicht behandelt.

Marinelli entwickelt für ein Problem im Bereich der Reihenfolge- und kapazitiven Losgrößenbestimmung eine 2-stufige Optimierungsheuristik, welche die Zuordnung von Produktionsaufträgen auf die jeweiligen Puffer im ersten Schritt vornimmt und diese anschließend mittels lokaler Suche auf den Produktionslinien reiht und terminiert. Im Zuge des ersten Schritts wurden die Operatoren *Shift-Move* und *Swap-Exchange* verwendet. Die Methode wird als Werkzeug zu einer optimierten Produktionslinienauslastung und im weiteren Sinne als Produktionsglättung verstanden. Die Zielfunktion berücksichtigt Rüstkosten, Lagerhaltungskosten sowie Produktionskosten. Die Freiheitsgrade der Optimierung sind für das vorliegende Problem aus der Lebensmittelindustrie v.a. durch verderbliche Güter sowie entsprechend

hohe Rüstkosten stark beschränkt. Daher wurden diverse Modellvereinfachungen vorgenommen. Wenn dies für die Anforderungen des vorliegenden Problems zutrifft, erscheint es zweckmäßig, im Zuge der Lösung des Problems auf diese Vereinfachungen zurückzugreifen. Der Algorithmus akzeptiert beispielsweise nur Verbesserungszüge in Bezug auf die Bewertung der Rüstkosten. Die Modellierung des Problems mittels CPLEX zerlegt das Problem in 2 Teilprobleme und löst diese zeiteffizient. (vgl. Marinelli et al. 2007, S. 177-192). Allerdings mangelt es dieser Methode an einem expliziten Glättungsansatz.

Wang und Wu stellen für ihr MOMPMP Problem einen 2-stufigen Lösungsansatz vor, wobei in der ersten Phase eine Heuristik als Eröffnungsverfahren zur Erstellung gültiger Lösungen verwendet wird, bevor in Phase 2 ein stochastisches, globales Optimierungsverfahren, ein hybrider Genetischer Algorithmus, eingesetzt wird. Die Autoren betonen, dass der vorgestellte hybride GA mit lokaler Suche einem reinen GA (bzw. einem BaB) deutlich überlegen und entsprechend effektiver ist, während ohne globaler Suche groß skalierte Probleme nicht effektiv lösbar sind. Wang und Wu unterscheiden bei der Modellierung der Ressourcen zeitbasiert zwischen regulärer Arbeitszeit, Überstunden und Auslagerungszeiten. Modellierungsspezifisch ist dabei die Annahme einer kontinuierlichen Produktion²⁵ sowie der Vorgabe, dass die Anzahl an Produkttypen pro Periode vorgegeben und beschränkt ist. Die Autoren bewerten in der Zielfunktion die Kriterien Lieferverzugskosten sowie die Kosten für Outsourcing, anfallende Bearbeitungskosten an jeder Arbeitsstation und Überstunden. Das Produktionssystem ist durch eine untere und obere Kapazitätsgrenze beschränkt. Die Methodik verkörpert die klassische Lösung eines MOMPMP gemäß dem Stand der Technik. Die Autoren betonen den Vorteil des Konzepts des lokalen Verbesserungsmechanismus im GA. Dies ermöglicht eine weitere Verbesserung der Lösung am Ende des GA-Suchprozesses (vgl. Wang und Wu 2003, S. 297-309). Die Modellierung lässt dynamische Nebenbedingungen sowie eine auf die Nivellierung von Produktionsmengen konkret zugeschnittene Zielfunktion vermissen und bewertet entsprechend Lieferverzugskosten in der Zielfunktion. Die Methodik bedient sich, mit Ausnahme der Bewertung von Überstunden im Modell, keiner speziellen über den SOTA hinausgehenden Techniken.

Rager legt den Betrachtungsschwerpunkt auf die Konzeption einer energieorientierten Maschinenbelegungsplanung und erweitert die klassischen Zielsetzungen in der Belegungsplanung (hohe Termintreue, niedrige Bestände in der Produktion, kurze Durchlaufzeiten, möglichst geringe Anzahl an belegten Maschinen) um die Betrachtung des zeitlichen Verlaufs des Einsatzenergieträgerbedarfs für identische, parallele Maschinen (vgl. Rager 2008, S. 9-34). Dabei werden Produktionsaufträge auf der Ebene einzelner Operationen modelliert, außerdem wird eine nichtlineare Zielfunktion eingeführt. Das NP-harte Problem, das dem eindimensionalen Bin-Packing-Problem (vgl. Korte und Vygen 2012, S. 503ff.) ähnelt, wird zuerst über ein Eröffnungsverfahren heuristisch einer gültigen Ausgangslösung zugeführt. Dabei findet bereits eine Zuordnung der Produktionsaufträge zu den Maschinen statt. Anschließend wird eine Pareto-optimale Lösung unter der Anwendung eines hybriden EA erreicht. Dadurch werden 10 – 20% der Endenergieträgerkosten eingespart sowie die Lastspitzen in der Produktion gleichzeitig um bis zu 30% reduziert. Die Minimierung der (gleichzeitig) belegten Maschinen glättet implizit auch den zeitlichen Verlauf des Einsatzenergieträgerbedarfs (vgl. Rager 2008, S. 110-129). Seine Arbeit ist methodisch relativ nahe mit dem vorliegenden Problem verwandt, wobei im vorliegenden industriellen Anwendungsfall vorrangig die Produktionsausstoßrate nivelliert werden muss. Als Hauptanforderung werden für die Anwendung der Methode von Rager einerseits die Verfügbarkeit der Einsatzenergieträgerbedarfsprofile vorausgesetzt. Andererseits ist methodisch ein entsprechender Fokus auf eine energieorientierte Maschinenbelegungsplanung (als Input) gegeben. Die Lastspitzenglättung wirkt sich dabei sowohl auf die laufende Maschinenbelegungsplanung als auch auf die Höhe der Investitionen für entsprechende (Energie-) Umwandlungsanlagen kostengünstig aus.

Karner stellt im Zuge seiner Dissertation ein Phasenmodell zur Entwicklung einer werkzeug- und maschinenzustandsbedingten Produktionsreihenfolgeoptimierung zur Verfügung (vgl. Karner 2019, S. 52ff.). Dabei verwendet er für die PPS relevante, zustandsbasierte Daten von Produktionswerkzeugen und -maschinen, um daraus Werkzeug- und Maschinenzustandsinformationen in die Produktionsplanung zu integrieren. Im Zuge der Modellierung des Anlagenzustands definiert und verwendet er dabei sogenannte *Healthpoints*. In Bezug auf die Optimierung werden logistische Zielgrößen zu einer

²⁵ist ein Produkt einmal in einer Periode gestartet worden, so wird dessen Menge bis zur Fertigstellung zu Ende produziert

kostenbasierten Fitnessfunktion aggregiert (vgl. Karner 2019, S. 75). Die Zielfunktion ergibt sich aus der Summe der gewichteten Teilzielfunktionen, wobei die Parametrierbarkeit durch die Gewichtung der Zielfunktion erhalten bleibt. Karner hält darüber hinaus in Bezug auf den Einsatz des jeweiligen Optimierungsalgorithmus fest, dass in den meisten Fällen ($\sim 20\%$) Eigenentwicklungen publiziert werden, welche in Grundbausteinen oft auf GA basieren, gefolgt von Vertretern aus der Gruppe Genetischer Algorithmen ($\sim 10\%$) sowie SA und sonstigen Ansätzen. Zusammenfassend überwiegt der Einsatz von Metaheuristiken. Dies bestätigt sich auch im Zuge seiner systematischen Literaturanalyse für das vorliegende Produktionsproblem. Karner stellt in diesem Rahmen fest, dass der Planungsbereich der Grobplanung mit rund 20% leicht unterrepräsentiert ist, verglichen mit Publikationen betreffend die Produktionsfeinplanung sowie Produktionssteuerung (vgl. Karner 2019, S. 181). Der Ansatz von Karner zielt v.a. auf die Optimierung der kurzfristigen Produktionsreihenfolge ab. Entsprechend fehlen in der Zielfunktion Charakteristika betreffend die Produktionsglättung bzw. ein mittelfristiger Planungshorizont. Diese Lücke im Bereich der Produktionsgrobplanung wird durch diese Arbeit geschlossen. An dieser Stelle wird festgehalten, dass in der Wissenschaft Eigenentwicklungen sowie Metaheuristiken, v.a. GA und SA, in der Anwendung dominieren.

Sobottka diskutiert und bewertet aktuell bestehende Planungsansätze in Bezug auf eine multikriterielle Anforderungsliste bestehend aus der Modellierung des Produktions- und Energiesystems, den Wechselwirkungen zwischen Materialfluss und thermisch-physikalischem Energiesystem, der automationsgestützten Optimierung sowie der Einsatzbereitschaft für die Industrie. Die Produktionsanlagen werden dabei im Zuge der Optimierung flexibel an- und abgeschaltet. Daraus resultiert die gesamtzeitliche Erfüllung der Anforderungen. In diesem Zusammenhang schlussfolgert er, dass bestehende Ansätze (Thiede, Junge etc.) und Methoden zwar bedingt in Bezug auf die Simulationstiefe für die Feinplanung geeignet sind, allerdings wichtige Eigenschaften für die direkte Verwendung bzw. den operativen Einsatz im Zuge der Co-Simulation fehlen. Allen voran fehlt die Integration der einzelnen Module in *einem hybriden Simulator für die Feinplanung* mit ausgeprägter Optimierung für den diskreten und kontinuierlichen Teil, berücksichtigend die Energieeffizienz von Maschinen und TGA sowie Gebäudehülle. Die Kombination aus diskreter Prozessablauflogik und kontinuierlicher Simulation ist dem Ansatz von *Sobottka* gelungen (vgl. *Sobottka* 2017, S. 66-97). Der Methode fehlt auf Grund der intensiven Datenanforderungen heute noch die Anforderungsbereitschaft sowie ein gezielter Fokus auf die reine Produktionsoptimierung bzw. konkret die Produktionsglättung. Daher soll die in dieser Arbeit *zu entwickelnde Methodik für die optimierte Mittelfristplanung*, bestehend aus einem mathematischen Modell, einer vereinfachten Simulation sowie einer dazugehörigen Optimierungsaufgabe, in einer Anwendungsumgebung mit diesem Fokus ausgestattet, implementiert, getestet und validiert werden.

3.10 Entwicklungsbedarf

Dieser Abschnitt definiert, basierend auf dem Abgleich der in den Abschnitten 3.4 - 3.6 und 3.9 untersuchten und diskutierten, vielversprechendsten Ansätzen, den von der vorliegenden Arbeit zu leistenden Entwicklungsbedarf für die Erstellung einer Produktionsglättungsmethodik, welche die in Abschnitt 1.2 definierten Anforderungen an die allgemeine Problemstellung bzw. das entsprechende Planungswerkzeug bestmöglich erfüllt. Die vollständige und eindeutige Ermittlung dieser Anforderungen bildet die Grundlage für die Funktionalität und Leistungsfähigkeit der zu entwickelnden Methodik (vgl. *Dick et al.* 2017, S. 2).

Tabelle 3.3 bewertet gemäß aktuellem Stand der Technik die verfügbare, relevante und thematisch passendste, diskutierte Primärliteratur in Bezug auf den folgenden vom Autor festgelegten Anforderungskatalog für die Eignung als Produktionsglättung. Die Anforderungen wurden im Zuge des Abgleichs der in den Abschnitten 3.4 - 3.6 und 3.9 vorgestellten Lösungsansätze aus der Literatur mit der allgemeinen Problemstellung in Abschnitt 1.2 aufgestellt. Diese leiten sich aus den charakteristischen Merkmalen an eine zu entwickelnde Optimierungsmethode mit dem Fokus auf die dynamische Glättung bzw. Nivellierung von Produktionsmengen ab. Die allgemeinen Anforderungskriterien in Bezug auf die Eignung für die zu entwickelnde Produktionsglättung werden derart festgelegt, sodass v.a. die

inhaltlichen Schwerpunkte Modellkomplexität und Problemklasse sowie die Art und Weise (Modelldynamik, Charakteristika) bzw. die Fristigkeit bezüglich der Modifikationen von Produktionsmengen und Losen hinsichtlich von Merkmalen einer Produktionsglättung umfassend berücksichtigt bzw. für jedes Vorgehen bewertet werden:

- **Modellkomplexität:** Durch dieses Kriterium werden Umfang und Detailgrad des abgebildeten Produktionssystems inklusive Zielfunktion und Nebenbedingungen berücksichtigt.
- **Problemklasse und Optimierungsmethode:** Dieses Kriterium bewertet die Schwere der Problemklasse sowie den Reifegrad der dazugehörigen Optimierungsmethode.
- **Quantitativer Modifikationsgrad:** Dieses Kriterium bewertet, in welchem Umfang Produktionsmengen bzw. Lose terminiert und sequenziert werden. Darüber hinaus wird auch berücksichtigt, ob und ggf. in welchem Umfang eine Vorverschiebung oder Aufteilung von Losen stattfindet bzw. ob oder auf welche Weise unterschiedliche Mengendimensionen (z.B. Kurantmengen und Aktionsmengen) betrachtet werden (gemeinsam oder separat).
- **Modelldynamik:** Dieses Kriterium bewertet, ob und in welchem Umfang das Modell z.B. dynamische Prozesszustände, dynamische Restriktionen, beschränkte Kapazitäten bzw. stochastische Verbräuche abbildet.
- **Produktionsglättungscharakteristika:** Dieses Kriterium bewertet, ob die Optimierungsmethode, die Zielfunktion oder die Lösung Charakteristika einer Produktionsglättung, also insbesondere der Nivellierung von Produktionsmengen, beinhalten.
- **Datenanforderungen:** Dieses Kriterium bewertet den Ansatz hinsichtlich seiner Anforderungen bezüglich Umfang und Granularität der benötigten Daten. Dies umfasst einerseits Stammdaten, Prozessdaten, Arbeitspläne, als auch kontinuierlich zu speichernde und auszuwertende Prozessdaten bzw. Datenanforderungen für die Bereitstellung von z.B. Einsatzenergeträgerbedarfsprofilen.
- **Fristigkeit:** Bewertung des Modellansatzes hinsichtlich seiner Fristigkeit (kurzfristig, mittelfristig, langfristig). Gemäß Abb. 2.8 eignen sich Ansätze mit mittel- und langfristigen Planungshorizont besser für die Glättung als kurzfristige Ansätze aus dem Bereich der Feinplanung.
- **Prozessflexibilität:** Dieses Kriterium bewertet den Grad der Prozessflexibilisierung, der sich über eine flexible Anlagenverfügbarkeit bis zu einer Variabilisierung der Prozessdurchlaufzeit erstreckt.
- **Integration von Simulation:** Über den Umfang und Detailgrad der Integration von Simulation wird die Realitätsnähe des Ansatzes in Bezug auf den jeweiligen Anwendungsfall bewertet.
- **Anwendungsbereitschaft und Laufzeiteffizienz:** Die Anwendungsbereitschaft leitet sich einerseits aus der Laufzeiteffizienz gemäß Landau (siehe 2.3.1.4) sowie andererseits aus dem Integrationsgrad und Umfang von Simulation ab. Laufzeiteffiziente Verfahren mit einem, im Verhältnis zur Laufzeit der Optimierung, nicht überproportional großen Simulationsanteil eignen sich eher für die Produktionsglättung.
- **Integration von Datenanalyse-Methoden:** Diese Anforderung bewertet die Methodik hinsichtlich der Integration einer statistischen Datenanalyse bzw. von Algorithmen für das maschinelle Lernen.

Die genannten Kriterien werden mittels, nach Anforderung absteigend sortiertem, in Summe 1 ergebenden Gewichtungsvektor $G = \{0.15, 0.10, 0.10, 0.10, 0.10, 0.10, 0.10, 0.10, 0.05, 0.05, 0.05\}$ zu einer Gesamtpunkteanzahl je Vorgehen aufsummiert. Die Abstufung der 3-teiligen Skala erfolgt von besonders relevant (Gewichtung 0.15) über relevant (Gewichtung 0.10) bis hin zu weniger relevant (Gewichtung 0.05). Hinsichtlich des Erfüllungsgrades eines jeden einzelnen Kriteriums werden ganzzahlig bis zu 4 Punkte vergeben, wobei dies symbolisch durch einen vollen Kreis dargestellt ist, während am unteren Ende der Skala ein leerer Kreis 0 Punkte für dieses Kriterium repräsentiert. Die Methoden werden gemäß ihrer erreichten Gesamtpunkteanzahl als auch der gewichteten Punkteanzahl von links nach rechts absteigend in Tab. 3.3 dargestellt.

Tab. 3.3: Vergleichende Bewertung methodisch relevanter Modellierungs- und Optimierungsansätze (Quelle: eigene Darstellung)

Gewichtung	Anforderungen	Vorgehen															
		Karimi-Nasab et al. (2012, 2011)	Wang und Wu (2002)	Sihn et al. (2018), Sobottka (2017)	Rager (2006)	Karner (2019)	Kogan et al. (2009)	Marinelli et al. (2007)	Absi und Kedad-Sidhoum (2007), Mercé et al. (2003)	Grundmann (2007)	AL-Ashhab (2006)	Stadler et al. (2000)	Sampaio et al. (2017)	Zupan et al. (2015)	Rewers et al. (2019)		
0,15	Modellkomplexität	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0,10	Problemklasse und Optimierungsmethode	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○
0,10	Quantitativer Modifikationsgrad	●	●	●	●	●	●	●	●	●	●	●	○	○	○	○	○
0,10	Modelldynamik	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0,10	Produktionsglättungscharakteristika	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0,10	Datenanforderungen	●	●	○	●	●	●	●	●	●	●	●	●	●	●	●	●
0,10	Fristigkeit	●	●	●	●	●	●	●	●	●	●	●	●	●	●	○	●
0,10	Prozessflexibilität	●	●	●	●	●	●	●	●	●	●	○	○	○	○	○	○
0,05	Integration von Simulation	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○
0,05	Anwendungsbereitschaft & Laufzeiteffizienz	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
0,05	Integration von Datenanalyse-Methoden	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Gesamtheitliche Erfüllung der Anforderungen	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Erreichte Punkteanzahl (absolut)	30,0	26,0	24,0	21,0	21,0	21,0	19,0	19,0	19,0	17,0	16,0	16,0	16,0	16,0	13,0	
	Erreichte Punkteanzahl (gewichtet)	3,00	2,55	2,35	2,15	2,00	2,00	1,95	1,90	1,85	1,80	1,65	1,55	1,45	1,45	1,15	

Die Methoden von Karimi-Nasab (vgl. Karimi-Nasab und Aryanezhad 2011, S. 3596-3610) sowie Wang und Wu (vgl. Wang und Wu 2003, S. 297-309) erreichen mit Abstand die höchsten absoluten als auch gewichteten Punkte. Karimi-Nasab berücksichtigt variable Prozesszeiten in seinem Modell als Vorgabe an die jeweilige Maschine. Beiden Methoden fehlt, neben den bereits erwähnten Punkten im Zuge der Diskussion (mangelnde Prozessvariabilisierung, charakteristische Zielfunktion bei Wang und Wu), die Realitätsnähe durch die Integration von Simulation bzw. statistischer Datenanalyse. Rager (vgl. Rager 2008) wendet das Konzept der Glättung im zeitlichen Verlauf an. Kogan (vgl. Kogan und Tell 2009) verwendet das Konzept der Vorverschiebung von Produktionsaufträgen, ohne dabei die Durchlaufzeit zu variabilisieren. Dem Ansatz von Sihm mangelt es an Produktionsglättungscharakteristika in der Zielfunktion und, bedingt durch die Integration hybrider Simulation (vgl. Sihm et al. 2018), der Mittelfristigkeit in der Methodik (vgl. Sobottka 2017). Das liegt v.a. in der Kurzfristplanung mit höherem Detailgrad und der damit verbundenen Integration der Energieeffizienz begründet. Mit Ausnahme des Ansatzes von Karner fehlt in jeder Optimierungsmethode der umfangreiche Einsatz von Datenanalyse-Methoden. Die Methodik von Karner zielt, wie auch der Ansatz von Sihm, auf eine effiziente Feinplanung ab. Der Ansatz von Karimi-Nasab berücksichtigt, neben dem rollierenden Planungsansatz von Sampaio, Produktionsglättungscharakteristika in der Zielfunktion und erhält durch die Modellkomplexität, Problemklasse und Prozessflexibilität mit den gestellten Anforderungen insgesamt die größte Übereinstimmung. Der Ansatz weist noch Entwicklungspotential in Bezug auf die Integration von Simulation bzw. Datenanalyse-Methoden auf.

Die zu entwickelnde Produktionsglättungsmethodik wird einerseits, sofern sinnvoll, durch die Integration von Simulation und Datenanalyse-Methoden möglichst realitätsnahe entwickelt. Andererseits wird der Anteil an Simulation im Verhältnis zur Gesamtlaufzeit der Optimierung relativ gering ausfallen, sodass die Optimierungsmethodik gleichzeitig auch eine hohe Anwendungsbereitschaft aufweist. Alle

vorgestellten und diskutierten Optimierungsmethoden vermissen außerdem die Variabilisierung eines Produktionsprozesses sowie die im Zuge der Optimierungsmethodik algorithmisch getrennte Behandlung von Aktions- und Kurantmengen.

3.11 Fazit

Abschnitt 3.10 bzw. Tab. 3.3 stellt die allgemeinen Anforderungen an eine zu entwickelnde Produktionsglättungsmethodik mit den dazugehörigen, in der Literatur existierenden, methodischen Ansätzen gegenüber. Dabei stellen sich diverse Mängel aus dem Stand der Technik heraus. Die schwerwiegendsten Mängel stellen die fehlende Variabilisierung der Durchlaufzeit von Produktionsprozessen sowie die methodisch nicht getrennte Behandlung von Aktions- und Kurantmengen dar. Weitere Defizite beziehen sich auf den notwendigen Simulationsanteil für eine an z.B. Ressourcen bzw. Kapazitäten berücksichtigende Mittel- bis Langfristplanung sowie die in diesem Zusammenhang mangelnde Verwendung von (z.B. Bestands-) Stammdaten im Rahmen der Methodik.

Die spezifischen Anforderungen an die zu entwickelnde Methodik in Kapitel 4 werden darüber entscheiden, ob und in welchem Umfang zur Auswertung eines Zielfunktionsbestandteils eine (vereinfachte) dynamische Simulation notwendig ist. In diesem Fall wird die zu entwickelnde Optimierungsmethode entsprechend mit dem gemäß Zielfunktionssystem und Periodizität notwendigen Anteil an Simulation ausgestattet und damit zu einer Simheuristik hybridisiert. Darüber hinaus ergibt sich ggf. ein Mehrwert durch eine statistisch-rollierende Datenauswertung, welche gleichzeitig durch die technische Infrastruktur einer rollierenden Produktionsplanung prinzipiell (automatisch) ermöglicht wird. Im Gegensatz zu den vorgestellten größtenteils starren Algorithmen, siehe Tab. 3.3, bzw. kommerziellen Solvern, welche aus verfügbaren historischen Daten bzw. Stammdaten nicht lernen, wird die in dieser Arbeit entwickelte Produktionsglättungsmethodik durch den rollierenden Planungsansatz adaptiv, wenn die für die Auswertung bzw. das Lernen notwendigen Daten periodisch abgespeichert, ausgewertet und genutzt werden. Dieser Umstand ist für die technische Infrastruktur wichtig, weil in einem frühen Stadium bereits die Daten mitgespeichert werden, welche zu einem späteren Zeitpunkt für die Auswertung benötigt werden, um daraus nützliche Informationen (mit entsprechendem Mehrwert für das Unternehmen) zu generieren.

Gemäß der Literaturanalyse stellen sowohl der GA als auch SA die am häufigsten eingesetzten Metaheuristiken dar und werden deshalb in dieser Arbeit für die Hybridisierung angewendet, spezifisch entwickelt und untereinander verglichen. Da ein Benchmark mit theoretischen Instanzen nicht (sinnvoll und zweckmäßig) möglich ist, erfolgt ein Vergleich der beiden Metaheuristiken gemäß der in Abschnitt 3.8 definierten Kriterien. Abschließend wird aus dem methodischen Vergleich eine Empfehlung für eine der beiden Metaheuristiken für die vorliegende Problemklasse hervorgehen. Die Konzepte des MOGA bzw. NSGA (II) verdeutlichen, dass eine zielgerichtete Suche auf vielversprechende Bereiche der Pareto-Front ein grundsätzlich erfolgreiches und gegenüber einem traditionellen GA häufig überlegenes Handeln darstellt, um qualitativ bessere Lösungen zu finden bzw. zu erreichen. Die Diversität der Population eines GA entspricht in etwa der Fähigkeit des SA, aus lokalen Optima ggf. wieder zu entkommen, wenn der SA entsprechend gut parametrisiert ist. Diese Fähigkeit wird beim SA, z.B. über eine Parameteroptimierung mittels Grid-Search, sichergestellt.

Der Gradual-Priority Weighting-Ansatz ist ein weiteres Mittel zum Zweck, um vielversprechende Bereiche der Pareto-Front zielgerichtet abzutasten. Es empfiehlt sich sowohl eine problemspezifisch gleichwertige Implementation des SA bzw. GA als auch eine Variation der einzelnen Gewichtungen, wie in 3.4.5.1 vorgestellt (vgl. Chang et al. 2002, S. 171-183). Dies erfolgt am Ende der Optimierung innerhalb eines definierten Bereichs, um die letztgefundene Lösung des jeweiligen Verfahrens neben der global besten Lösung sowie weiters z.B. der besten (schwach) Pareto-optimalen Lösungen hinsichtlich ihrer Robustheit bezüglich verschiedener Sets an Gewichtungen zu bewerten. Die jeweils besten (schwach) Pareto-optimalen Lösungen werden dabei in Bezug auf z.B. die wichtigsten 2 Teilziele herangezogen. Karimi-Nasab ermittelt die beste Lösung mittels TOPSIS anhand der finalen Liste Pareto-optimaler Lösungen (vgl. Karimi-Nasab und Konstantaras 2012, S. 479ff.).

4

Kapitel 4

Produktionsproblembeschreibung und Anforderungsübersicht

Kapitel 4 gliedert sich in den Produktionsablauf und die Spezifika des konkret vorliegenden Produktionssystem. Darauf basierend wird eine Anforderungsübersicht an die zu entwickelnde Produktionsglättungsmethodik erstellt.

Das Vorgehen besteht allgemein in der Ist-Prozessaufnahme und Analyse des vorliegenden Produktionssystem. Dabei werden die wesentlichen Produktionsschritte (vom Wareneingang bis zum Warenausgang) und Kostentreiber sowie die Engpässe in der Produktion identifiziert und analysiert. Das können sowohl Produktionsanlagen als auch räumlich begrenzt zur Verfügung stehende Zwischenspeicher darstellen. Danach werden die Systemgrenzen aufgestellt. Anschließend werden die Kernanforderungen an die Methodik, die das entsprechende Problem löst, formuliert und priorisiert. Für ein gutes Problemverständnis ist eine umfangreiche Datenanalyse und Aufbereitung mit dem Fokus auf die für die Problembearbeitung relevanten Daten erforderlich, um den Input für ein entsprechend zu formulierendes Modell möglichst realistisch und vollumfänglich darzustellen (vgl. Hillier und Lieberman 2015, S. 12).

4.1 Produktionsablauf des vorliegenden Produktionssystem

Der im Zuge dieser Arbeit betrachtete industrielle Anwendungsfall ist ein in Österreich neu gebautes Produktionswerk der fleischverarbeitenden Industrie. Basierend auf einer täglichen Warenanlieferung (Frischfleisch) werden *Schlögel*¹ angeliefert und in das Wareneingangslager eingelagert, bevor diese binnen 48 Stunden weiterverarbeitet werden müssen. Die Einsalzanlage bildet den ersten Schritt der mehrstufigen sogenannten *Grundproduktion*, siehe auch Abb. 4.1, und stellt gleichzeitig den primär identifizierten, kapazitiven Produktionsengpass im betrachteten Produktionssystem dar. Auf Basis der erhaltenen Anlagenleistungsdaten erfolgt eine (Wertstrom-) Analyse, welche diesen Produktionsengpass als Resultat ausweist. Die Gründe liegen einerseits an der durch die Anlage angebotene Leistung sowie dem Raumbedarf für die gesamte Einsalzanlage als auch im Investitionsbedarf für diese Anlage und die damit unmittelbar verbundenen operativen Kosten durch die manuellen sowie automatisierten Prozesse. Nach der manuellen Entnahme von der Palette wird jeder Schlögel auf der Anlage automatisiert auf seine geometrische Form, Blut-, pH- und Fettwerte sowie sein Gewicht geprüft, bevor dieser mit einer speziellen Einsalzmischung automatisiert eingesalzen wird. Am Ende der Anlage werden die Schlögel nach einer manuellen Sichtprüfung vom Band genommen und auf entsprechende, fahrerlose Transportfahrzeuge gepackt, welche im Anschluss gemäß dem jeweils definierten Arbeitsplan die weiteren Schritte der mehrstufigen Grundproduktion (Pökeln, Riposo, Selch, Klimalager, Reifelager) bedienen. Nach dem Reifelager endet mit der Grundproduktion gleichzeitig auch der Ausschnitt des betrachteten Produktionssystem. Die Grundproduktion erfolgt kundenanonym bis zum Kundenentkopplungspunkt, welcher unmittelbar nach dem (verlängerten) Reifelagerprozess angesiedelt ist. Nach dem Kundenentkopplungspunkt erfolgt der Zuschnitt bzw. die Verpackung auf der jeweils geeigneten

¹in etwa 7 - 12 kg schwere Teile des Hinterbeins von Tieren bei der Fleischproduktion.

Slicer-Anlage² sowie die dazugehörig nachgelagerte Kommissionierung für den*die Endkund*in aus dem Lebensmitteleinzelhandel (vgl. Kamhuber et al. 2020, S. 4).

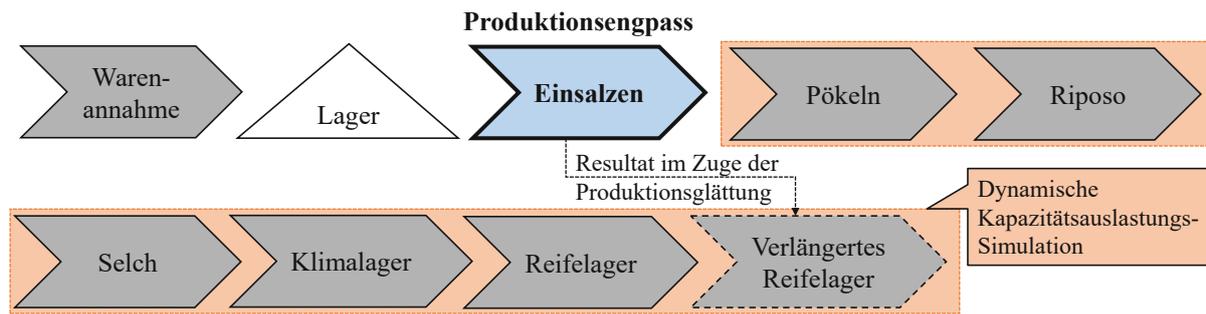


Abb. 4.1: Betrachteter Ausschnitt des Produktionssystems: Vom Einsalzen bis zum Abschluss der Grundproduktion (vgl. Kamhuber et al. 2019, S. 2110)

Das übergeordnete Hauptziel der rollierenden Planungsmethodik liegt gemäß Engpasstheorie (vgl. Lenort et al. 2012, S. 225), siehe Abschnitt 1.5, in einer möglichst gleichmäßigen (idealerweise gleichzeitig nahezu maximalen) mittel- und langfristigen Auslastung des zuvor identifizierten Produktionsengpasses (Einsalzanlage) als auch der sämtlichen weiteren Grundproduktionschritte, damit sich der Engpass nicht nur verschiebt, sondern bestmöglich ganzheitlich durch die zu entwickelnde Planungsmethodik aufgelöst wird. Dafür ist konkret die Entwicklung einer integrierten, dynamischen Ablaufsimulation zur möglichst akkuraten Entwicklung der Kapazitätsauslastung für alle Grundproduktionsbereiche über den gesamten Planungshorizont hinweg notwendig. Allgemein wird bei Planungsproblemen mit begrenzt zur Verfügung stehenden Kapazitäten, siehe die allgemeine Problemstellung aus Abschnitt 1.4, eine entsprechende Kapazitätsauslastung der Aggregate benötigt. Konkret wird diese Kapazitätsauslastung im Zeitverlauf für den vollständigen mittel- bis langfristigen Planungshorizont benötigt, siehe Abschnitt 2.5. Auf Grund der Fristigkeit der vorliegenden Problemstellung wird in der Simulation für die Periodizität der Arbeitspläne der Produktionsartikel die Dimension Woche verwendet. Die Engpasstheorie sieht gemäß Schritt 5 vor, dass es nicht genügt, lediglich den ersten Kapazitätsengpass zu identifizieren und zu beheben, sondern auch sämtliche weitere danach (durch die Behebung des vorherigen Produktionsengpasses) entstandenen Engpässe. Der mittelfristige Planungshorizont sieht die Produktionsplanung aller ca. 25 Produktionsartikel (je nach Saison) über einen Planungshorizont von 78 Wochen vor (vgl. Kamhuber et al. 2019, S. 2110; Kamhuber et al. 2020, S. 4).

4.2 Spezifika des vorliegenden Produktionssystems

Die Einsalzanlage ist eine hochtechnologische Anlage, welche die genannten Tätigkeiten großteils automatisiert übernimmt und über eine variable, d.h. einstellbare, Produktionsgeschwindigkeit verfügt. Dadurch wird dieser Produktionsengpass als dynamische Nebenbedingung modelliert, also im Zeitverlauf pro Periode schwankend berücksichtigt. In diesem Zusammenhang ist das Personal (für manuelle Tätigkeiten wie z.B. das Auflegen auf bzw. Abpacken von der Anlage) entsprechend dynamisch verfügbar. Auf Grund der Investitionskosten sowie des Raumbedarfs wurde nur eine Anlage für das neu errichtete Werk angeschafft. Entlang des betrachteten Ausschnitts in der Werkschöpfungskette ist der *verlängerte Reifelagerprozess* von zentraler Bedeutung: Dieser Prozess stellt allgemein eine Art *Vergütungslager* dar und ist methodisch notwendig bedingt durch die Variabilisierung der Reifelagerdauer (siehe Reifelagerprozess in Abbildung 4.1 gemäß Arbeitsplan) und dient als Puffer für die Optimierung, welche sich der technologisch variablen Prozessdurchlaufzeit an dieser Stelle bedient.

Ein zweites wichtiges Merkmal des Produktionssystems ist die Möglichkeit der Variabilisierbarkeit des Reifelagerprozesses. Als weiteres, spezielles Merkmal des Produktionssystems ist die Nebenbedingung

²Slicer-Anlagen sind automatisierte Zuschnitt- und Verpackungsanlagen in der Lebensmittelindustrie, um für den LEH fertige Verkaufseinheiten zu bilden, welche in Kartons und Paletten verpackt vertrieben werden.

nicht auf Artekebene definiert, sondern unterscheidet auch zwischen Kurantmengen und Aktionsmengen. Aktionsmengen dürfen i.d.R. pro Artikel über einen längeren Zeitraum (max. 12 Wochen) hinweg versetzt werden, während die Standardmengen, welche in der Fachsprache auch als *Kurantmengen* bezeichnet werden, über einen deutlich kürzeren Zeitraum (max. 4 Wochen) vorverschoben werden dürfen. Der Hauptgrund dafür liegt darin, dass die wenigen aber großen Aktionsmengen auf jeden Fall bestmöglich aufgeteilt werden, während die Kurantmengen lediglich saisonale Effekte beinhalten, für die es keiner Verschiebung über einen längeren Zeitraum hinweg bedarf. Kurantmengen sind im Gegensatz zu Aktionsmengen bestandswirksam. Diese Einstellung kann für jeden Artikel gesondert erfolgen, wobei bestimmte Artikel derart eingestellt werden, dass deren Produktionsmengen nicht versetzt werden dürfen.

Eine weitere Besonderheit des Produktionssystems stellt die Produktionskapazität der gesamten Grundproduktion (Pökeln, Riposo, Selch, Klimalager, Reifelager) dar. Auf Grund der räumlich-kapazitativ beschränkten Gegebenheiten sind diese Produktionsräume als statische Nebenbedingungen zu berücksichtigen, wobei jede Raumgruppe ihrerseits über eine spezifische Gesamtkapazität verfügt. Die Grundproduktion verfügt in Summe über eine relativ lange prozessbedingte Durchlaufzeit, welche in Abhängigkeit des Arbeitsplans eines jeden Artikels zwischen ca. 10 und max. 32 Wochen liegt. Durch die Variabilisierung eines Arbeitsschrittes kann diese *Mindestdurchlaufzeit* für bestimmte Teilmengen jedes Produktionsartikels überschritten bzw. verlängert (*relaxiert*) werden. Die Artikel verfügen über unterschiedlich lange Teildurchlaufzeiten in den einzelnen Downstream Prozessen nach dem Einsalzen, siehe auch Tabelle 4.1. Die größten Schwankungen der einzelnen Artikel weist dabei das Klimalager auf.

Tab. 4.1: Übersicht über die min. und max. Teilprozessdurchlaufzeiten (Quelle: eigene Darstellung)

Downstream Kernprozesse	Pökeln	Riposo	Selch	Klimalager	Reifelager
Minimum [Tage]	14	7	7	7	14
Maximum [Tage]	21	21	7	140	35

Im Zusammenhang mit dem industriellen Anwendungsfall aus der Lebensmittelindustrie ergibt sich außerdem, dass Umrüstvorgänge an der Einsalzanlage relativ einfach und kurzfristig umsetzbar sind. Dabei muss lediglich das verwendete Einsalzprogramm bzw. die dazu notwendige Einsalzmischung entsprechend geändert werden, sobald das Einsalzband leer ist und den vorherigen Artikel abgearbeitet hat. In der Vorbereitung müssen auch die Mengen der neuen Artikelnummer ausgelagert und vor der Einsalzanlage vorhanden sein, damit diese anschließend manuell aufgelegt werden. Der tatsächliche Umrüstvorgang ist damit in Bezug auf seine Ausführungsdauer mit einigen Minuten verglichen mit der Produktivzeit der Anlage vergleichsweise gering.

Losteilung: Im Zuge der Lösung des gegebenen Produktionsproblems kann und darf explizit das Konzept der Losteilung angewendet werden. Low hält im Zuge seiner Experimente fest, dass eine Batch-Zuordnung mit gleich großen Teillosten eine kürzere Produktionsdauer als eine Zuordnung mit ungleichen Teillosten ergibt. Dieser Effekt nimmt mit der zunehmenden Anzahl an Teillosten ab. Low schlägt für NP-harte Planungsprobleme außerdem die Entwicklung von heuristischen Algorithmen vor, welche sich des Konzepts der Losteilung bedienen, um den Suchraum entsprechend effizient zu durchsuchen. Der Beitrag von Low betrachtet den Einfluss des Zusammenhangs zwischen Rüstkosten und Bearbeitungszeiten nicht und belässt diese Fragestellung für die zukünftige Forschung offen (vgl. Low et al. 2004, S. 773).

Im Rahmen der in dieser Arbeit zu entwickelnden Methodik wird die Losteilung in Bezug auf das gestellte Problem für zwei parallel zu betrachtende Dimensionen angewendet und mit den dadurch entstehenden zusätzlichen Rüstkosten gegenübergestellt. Die Forschungslücke, dass Produktionsmengen zur Verfahrenslaufzeit einerseits in zwei Betrachtungsebenen separat bzw. eigenständig optimiert und anschließend wieder gemeinsam als Produktionslosmenge betrachtet werden, um z.B. die gesamten anfallenden Produktionsbedarfe für die Kapazitätsauslastung entsprechend beurteilen zu können, ist eine bedeutende Fragestellung, die in dieser Arbeit für das vorliegende Produktionssystem untersucht

wird. Rewers hat im Zuge seiner industriellen Studie festgestellt, dass die Einführung einer geglätteten Produktion am Beispiele einer Produktion für chirurgische Instrumente einerseits die Durchlaufzeit um 30% verkürzt sowie die Effizienz der Ressource Mensch um 20% zugenommen hat (vgl. Rewers et al. 2017, S. 619-626).

Problemkomplexität: Die Komplexität von diesem Teil des gesamten Problems hängt zum einen von der aufgespannten Matrix ab, welche die Anzahl an Artikeln mit dem betrachteten Planungshorizont ergibt. Zweitens stellt sich auch die Frage, ob und unter welchen Bedingungen ein Produktionslos geteilt bzw. aufgesplittet wird. An dieser Stelle ist jedenfalls angemerkt, dass der aufgespannte Suchraum deutlich geringer ist, wenn Produktionsaufträge von bestimmten Artikeln nur binär, d.h. zur Gänze oder überhaupt nicht, versetzt werden, als wenn auch eine (nach bestimmten Regeln nicht beliebige) Teilung der Auftragslosgröße möglich ist. Im Falle der Möglichkeit der Losteilung stellt sich die nächste Frage, in welcher *Diskretisierung* bzw. Granularität dies möglich ist. Unabhängig davon ist auch die Frage zu beantworten, wie viele Produktionsvolumina (in einer Periode von einem Artikel) durchschnittlich produziert werden. Es erscheint logisch, dass sich im Falle von kleinen Produktionsmengen die Suchlandschaft entsprechend reduziert, weil in jeder Periode eine Menge ≥ 0 produziert wird. Das bedeutet, dass in keiner (Zwischen-) Lösung eine Periode mit negativen Produktionsmengen existiert. Die letzte vom industriellen Anwendungsfall zu beantwortende Frage bezieht sich auf die Anforderungen aus dem Fazit von Kapitel 2 bzw. konkret Abbildung 2.4, also die Variabilisierbarkeit des Produktionsprozesses. Für die Auswahl des Verfahrens macht es darüber hinaus auch einen Unterschied, ob Produktionsaufträge bzw. Lose - in Bezug auf eine gültige Ausgangslösung - nur früher oder auch später produziert werden können. Dementsprechend muss das Kriterium Liefertermintreue mit in das Zielfunktionssystem aufgenommen werden oder ist andererseits in der Betrachtung vernachlässigbar. Im konkreten industriellen Anwendungsfall liegt die Beschränkung vor, dass Produktionsmengen zeitlich nur nach vorne versetzt werden. Dies schränkt den Freiheitsgrad der Optimierung entsprechend ein und erlaubt dafür, dass das angesprochene Kriterium der Liefertermintreue im Rahmen des Zielfunktionssystems in dieser Arbeit nicht explizit betrachtet bzw. berücksichtigt wird. Dadurch entfällt für die Optimierung eine häufig widersprüchliche Zielfunktionsgröße, siehe Unterabschnitt 2.3.2, wodurch das Zielfunktionssystem des Problems vereinfacht wird (vgl. Fu et al. 2019, S. 515-525).

4.3 Spezifische Anforderungsübersicht an das zu entwickelnde multikriterielle Optimierungsverfahren

Gemäß der in Abschnitt 4.2 konkretisierten Spezifika des Produktionssystems wird eine leistungsfähige und gleichzeitig für den rollierenden Einsatz *laufzeiteffiziente* Planungsmethode mit den folgenden Anforderungen, welche sich aus dem Entwicklungsbedarf in Abschnitt 3.10 bzw. Tab. 3.3 ableiten, entwickelt:

- Die Planungsmethode unterscheidet bei Produktionsmengen zwischen sogenannten *Kurantmengen* und *Aktionsmengen* derart, sodass diese in der Optimierung separat betrachtet werden. Kurantmengen stellen gewöhnlich anfallende Produktionsmengen dar, welche kontinuierlich benötigt werden, um den Absatzmarkt entsprechend zu befriedigen. Diese repräsentieren das laufende, durch Listungen im LEH bedingte, normale Geschäft. Dieses unterliegt geringeren Schwankungen als das Geschäft über Aktionsmengen, welche vertriebsseitige kurzfristige Vereinbarungen³ mit dem LEH mit dem übergeordneten Ziel darstellen, innerhalb eines definierten Zeitraums mehr Volumina abzusetzen⁴. Die Kategorie der Aktionsmengen umfasst über den normalen Bedarf hinausgehende Mengen, die im Zuge einer vom Vertrieb mit dem*der Kund*in vereinbarten und zeitlich abgestimmten Aktion zusätzlich zum normalen laufenden Geschäft anfallen. Aktionsmengen werden jeweils mit einem*einer bestimmten Kund*in vereinbart und sind in der Methode mit einer zugeordneten Aktionsnummer zu kennzeichnen.

³Diese Vereinbarungen beinhalten Aktionen mit einem reduzierten Preis, Zweitplatzierungen im Geschäft bzw. Änderungen an Menge oder Design der Verpackung.

⁴Diese Zusatzvolumina werden im Nachgang an eine Aktion vom LEH ausgewerte, z.B. über die Scannermengen.

- Der multikriterielle Ansatz bedingt zusätzliche Restriktionen auf der jeweiligen Planungsebene und eröffnet zudem die Erschließung des damit verbundenen Potentials auf der jeweiligen Planungsdimension (z.B. Produktionsmengen, Kapazitäten, Bestand etc.).
- Produktionsauftragsmengen werden im Zuge dieser Methodik dynamisch, also rollierend, verändert bzw. in kleinere Auftragsmengen auf die einzelnen Perioden aufgeteilt und dadurch mathematisch feiner optimiert werden. Dieses Merkmal ist speziell im Zuge der rollierenden Betrachtung notwendig, weil sich die tatsächlich in einer Periode produzierten Mengen von den geplanten unterscheiden oder die Absatzplanung aktualisiert worden ist.
- Das Optimierungsverfahren ist möglichst transparent und nachvollziehbar für den*die Benutzer*in betreffend die Generierung von Lösungen als auch laufzeiteffizient gemäß der Landau-Notation, siehe 2.3.1.4, und modular aufgebaut.
- Darüber hinaus analysiert die Methodik, bedingt durch ihren rollierenden und selbstbewertenden Charakter, gesammelte Produktionsdaten aus der Vergangenheit und wertet diese laufend aus. Dadurch passt sich das Verfahren dynamisch an die Zukunft an und erschließt die sich dadurch ergebenden Potentiale selbständig.

Tabelle 4.2 liefert eine modulare Übersicht bezüglich der gestellten Kernanforderungen an das zu entwickelnde Optimierungsverfahren. Die dritte Spalte gibt den Abschnitt an, in welchem die entsprechende Anforderung im Zuge der Modellbildung umgesetzt wird. Die zweite Spalte stellt die Kategorie dar, in welche das Modul hineinfällt. Grundsätzlich verfügt gemäß des Operations Research Modelling Approach jedes Optimierungsproblem über Module, welche in den Bereich der **Modellierung** des Problems hineinfallen, als auch Module betreffend die Umsetzung der dazugehörigen **Optimierung**, siehe dazu auch die entsprechenden Kategorien in Tab. 4.2.

Tab. 4.2: Anforderungsübersicht an das Optimierungsverfahren (Quelle: eigene Darstellung)

Anforderungsbezeichnung	Kategorie	Abschnitt(e)
Zielfunktionssystem	Modellierung	5.3
Statische Nebenbedingungen	Modellierung	5.3
Dynamische Kapazitätsgrenze	Modellierung	5.3
Kapazitätsauslastungsvorschau	Modellierung	5.4
Hybride Optimierung	Optimierung	5.6
Aktionsmengenglättung	Optimierung	5.6
Kurantmengenglättung	Optimierung	5.6
Bestandsverlaufsglättung	Optimierung	5.6
Laufzeiteffizienz	Modellierung	5.3 - 5.6, 6.3
Anpassungsfähigkeit	Optimierung	5.7, 6.4

Aufbau des Zielfunktionssystems

Das Zielfunktionssystem ist derart aufgebaut, dass gemäß der Kernanforderung eine klare Fokussierung auf die Nivellierung der Produktionsausstoßrate auf Artikel- und Werksebene mit der Differenzierung zwischen jeweils Aktions- und Kurantmengen gegeben ist. Darüber hinaus wird die Auslastung der Kapazitäten an der Engpassanlage dynamisch bzw. an den zur Verfügung stehenden Raumkapazitäten statisch bewertet, damit die Schwankungen entlang der gesamten Grundproduktion betrachtet und in weiterer Folge minimiert werden können. Außerdem haben Expert*innenbefragungen ergeben, dass die zu entwickelnde Methodik die Zielbestandsreichweite jedes Artikels dynamisch berücksichtigt. Im Zuge der Systemintegration wird außerdem die Anforderung an die Optimierung gestellt, dass sämtliche Schritte der Methodik einzeln nachvollziehbar und jeweils begründbar sind. Um diese Zielsetzungen transparent und verständlich zu gestalten, wird wie folgt vorgegangen:

1. Entwicklung einer lexikographisch wissensbasierten und laufzeiteffizienten Heuristik.

2. Darauf aufsetzend: Entwicklung und Vergleich zweier Metaheuristiken, welche mit der gewichteten Zielfunktion arbeiten.

Statische Nebenbedingungen

Artikelspezifische Nebenbedingungen beinhalten statische Stammdatenvorgaben:

- Definierter, nichtveränderlicher Mindestbestand pro Artikel.
- Statische Vorgaben aus der Raumkapazitätsverfügbarkeit.
- Vorgaben aus der Optimierung betreffend die Rundung der Ergebnisse auf Partien (C-Artikel) bzw. halbe (B-Artikel) oder volle Racks (A-Artikel) gemäß der von dem*der Anwendungspartner*in bereitgestellten ABC-Einteilung der Artikel.

Dynamische Kapazitätsgrenze

Die dynamische Kapazitätsgrenze wird im Zuge der Optimierung prinzipiell (de-) aktivierbar ausgestaltet und im Zuge der Optimierung grundsätzlich aktiviert und auch berücksichtigt.

Allerdings existiert eine Ausnahme am Ende eines jeden Optimierungsaufbaus auf Artekebene in beiden Mengendimensionen (Kurant- und Aktionsmengen): Kurantmengen werden im Falle des Verstoßes gegenüber einer gesetzten dynamischen Kapazitätsgrenze in einem relativ engen (Versatz-) Zeitfenster trotzdem produziert, während Aktionsmengen über einen deutlich längeren Zeitraum versetzt und eingelätet werden. Zumeist lässt sich über einen längeren Zeitraum eher eine (kapazitive) Lücke finden. Prinzipiell sieht die Methodik entsprechende Mengenkürzungen bei Nichteinhaltung der Nebenbedingung vor. Allerdings werden die gesamten Produktionsmengen konstant gehalten. Durch die Methodik dürfen daher automatisiert keine Mengen aus dem erhaltenen Absatzplan gelöscht bzw. reduziert werden. Diese Anforderung ist ein zentrales Element der Produktionsglättung. Dementsprechend wird diese Nebenbedingung entsprechend *relaxiert*. Wenn sich für eine Kurant- oder Aktionsmenge am Ende einer jeden Optimierungsroutine keine Lücke im Planungshorizont finden lässt, wird die Differenzmenge daher auf alle Perioden möglichst gleichmäßig aufgeschlagen und die Verstöße werden im Zuge der Ausgabe angezeigt. Die Planungsabteilung kann den Produktionsvorschlag entsprechend manuell anpassen, z.B. mittels einer

- Erhöhung der dynamischen Kapazität der Einsalzanlage in der betroffenen Woche: Dies führt i.d.R. zu einem neuen Planungslauf mit aktualisierten Stammdaten (*Szenario*).
- Reduktion der Produktionsartikelmengen von einem bzw. mehreren definierten Artikel (Dies kann nur manuell erfolgen, weil eine Reduktion von initial erhaltenen Mengen nicht als Freiheitsgrad für die Optimierung freigegeben ist).
- Erhöhung des Schiebefensters für den betroffenen Artikel (*Szenario*).

Während die Erhöhung der dynamischen Kapazität bzw. des Schiebefensters als Szenario automatisiert abbildbar ist, erhält die Methode nach Abstimmung mit den Planer*innen nicht die Möglichkeit der automatisierten Reduktion einer bestimmten Produktionsmenge. Die Planungsabteilung möchte sich vorbehalten, ob ggf. weniger Mengen produziert bzw. an der Engpassanlage aufgelegt werden oder ob die Kapazität an der Anlage in der betroffenen Periode hochgefahren wird, um den Absatzplan einzuhalten. Die Problematik besteht darin, dass der Absatzplan nicht mit der verfügbaren Kapazität abgestimmt ist und somit ggf. mehr Mengen in das System gelangen, als an der Engpassanlage verarbeitet werden können. Eine ähnliche Relaxation existiert auch betreffend die Nebenbedingung für die Raumkapazitätsverfügbarkeit.

Integrierte Simulation: Dynamische Kapazitätsauslastungsvorschau

Die Hauptanforderung an die dynamische Kapazitätsauslastungsvorschau stellt die auf Palette aggregierte Vorschau der Kapazitätsauslastung der Grundproduktionsprozesse (Pökeln, Riposo, Selch,

Klimalager, Reifelager) dar, siehe auch Abb. 4.1. Die Aggregation auf Palettenebene ergibt sich über die Definition von Artikelgruppen. Es können nur Produktionsartikel derselben Artikelgruppe zusammen auf eine Palette gepackt werden. Dementsprechend wird die Artikelgruppe eines jeden Produktionsartikels im Zuge der Stammdaten auf der Datenbank zur Verfügung gestellt. Die Kapazitätsauslastungsvorschau ergibt sich somit über den individuellen Arbeitsplan eines jeden Produktionsartikels. Die Simulation erstellt über den gesamten Planungshorizont eine Kapazitätsvorschau auf Wochenebene für jede einzelne Raumgruppe, um im Einklang mit der Planung (auf Wochenebene) die Ergebnisse darzustellen.

Hybride Optimierung

Basierend auf Kapitel 3 wird im Zuge der Optimierungsmethodik zuerst eine problemspezifische, zugeschnittene Heuristik, welche in der Abarbeitungsreihenfolge und Programmlogik zwischen Aktions- und Kurantmengen differenziert, für die Glättung von Produktionsmengen entwickelt. Anschließend werden im Rahmen der metaheuristischen Optimierung ein Simulated Annealing Verfahren und ein Genetischer Algorithmus vergleichend implementiert. Die Verfahren werden möglichst *gleichwertig* unter Berücksichtigung der jeweiligen Spezifika umgesetzt. Beide Verfahren setzen auf die beste durch die Heuristik zuvor gefundene Lösung auf und erhalten eine Parameteroptimierung, die speziell beim SA erforderlich ist, weil dieses Verfahren nur auf die jeweils vorangegangene, entwickelte Trajektorie aufbaut.

Aktionsmengenglättung

Aus der Analyse der Spezifika des vorliegenden Produktionssystems geht hervor, dass, in Abhängigkeit von Saison und Datensatz, bis zu 30% der Volumina Aktionsmengen darstellen. Das Verfahren verteilt diese Mengen systematisch in einer effizienten Art und Weise, sodass die Mengen möglichst zeitnah unter Berücksichtigung der artikelspezifischen Nebenbedingungen produziert werden. Bei der Aktionsmengenglättung wird im Zuge der Anwendung der Losteilung stets die Aktionsnummer mitgeführt. Bei jedem Glättungsschritt wird sowohl die Originalwoche als auch die neu vorgeschlagene Woche mitgespeichert. Aktionsmengen sind nicht bestandswirksam, dementsprechend werden Aktionsmengen nach Produktionsabschluss nicht im Bestand geführt, bevor sie weiterverarbeitet werden. Dementsprechend haben Aktionsmengen keinen Einfluss auf den Zielbestand.

Kurantmengenglättung

Die Kurantmengenglättung stellt ähnliche Anforderungen an die Optimierung wie die Aktionsmengenglättung. Der Hauptunterschied liegt in der unterschiedlichen Wertebelegung für artikelspezifische Nebenbedingungen. Diese Arbeit stellt sich im Zusammenhang mit der Optimierung auch der Frage, ob die Glättung der Aktionsmengen prinzipiell vor oder nach der Kurantmengenglättung stattfindet. Mangels Vorhandensein entfällt das Mitführen einer definierten Aktionsnummer. Kurantmengen sind bestandswirksam, d.h. diese werden am Ende der Produktion bzw. konkret nach dem Abschluss des Reifelagerprozesses auf Lager gelegt bzw. im Bestand geführt, bevor sie weiterverarbeitet werden.

Bestandsverlaufsglättung

Die Glättung des Bestandsverlaufs ist aus Sicht des*der Entscheidungsträger*in v.a. für die beeinflussbare nahe bevorstehende Zukunft notwendig. Dementsprechend erscheint es wichtig, diese Anforderung gezielt umzusetzen. Bei Minderbestand wird schrittweise auf den Mindestbestand bzw. Sollbestand aufgepuffert. Überbestand wird relativ rasch, d.h. in wenigen Perioden, abgebaut. Das Modul wird heuristisch in der Form implementiert, sodass zuerst alle Produktionsartikel identifiziert werden, welche eine Abpufferung erfordern. Das Ziel besteht darin, Kapazitäten freizugeben, welche zusätzlich für die Aufpufferung von den verbleibenden Artikeln verwendet werden kann. Dabei werden die Artikel

vorgereicht, welche zuerst (nur mittels Kurantmengen) auf den Mindestbestand aufgepuffert werden, bevor die Kapazität für Artikel verwendet wird, die eine Aufpufferung auf den Sollbestand erfordern⁵.

Laufzeiteffizienz

Die planungsseitige Vorgabe an die Programmlaufzeit sieht eine einstellbare, maximale Ausführungsdauer von 15 Minuten pro Planungslauf vor. Dasselbe betrifft die prinzipielle (De-) Aktivierbarkeit der Metaheuristik. Um eine effiziente Methodik zu entwickeln, wird zuerst eine Heuristik entwickelt, welche ohne Zielfunktion und Simulation auskommt und teilweise mehrere Optimierungsschritte pro heuristischer Iteration auf einmal ausführen kann. An dieser Stelle sei erwähnt, dass durch die Losteilung eine ursprüngliche Produktionsmenge durch die Heuristik auf mehrere Perioden aufgeteilt wird. Dies entspricht mehreren Optimierungsschritten einer Metaheuristik, wobei es für die Heuristik nur eine Iteration darstellt. Die Effizienz der zu entwickelnden Methodik wird theoretisch ermittelt und anschließend praktisch validiert sowie mit anderen prinzipiell möglichen Lösungsverfahren hinsichtlich der Laufzeiteffizienz gemäß Landau-Notation verglichen.

Statistische Datenanalyse: Grundlage für die Anpassungsfähigkeit

Im Zuge der Anpassungsfähigkeit und der Berücksichtigung von (alternativen Produktions-) Szenarien ist eine Datenbank-Umgebung notwendig, um (veränderliche) Nebenbedingungen für die Optimierung oder neue bzw. aktualisierte Planungsdaten zur Verfügung zu stellen. Der*die industrielle Anwendungspartner*in verwendet bereits eine Oracle Datenbank und stellt diese zur Anbindung der Planungsmethodik zur Verfügung. Der Aufruf erfolgt über eine Init-Datei mit einstellbaren Parametern. An ein entsprechendes, statistisches Datenanalyse-Modul zur Erhebung und Auswertung des Bestandsrisikos werden die folgenden Anforderungen gestellt:

- Rollierende Anpassung der kategorischen ABC-Zuordnung auf Artikelebene.
- Rollierende Bestandsverlaufsauswertung, um die Zielbestandsniveaus aktualisiert anzupassen.
- Eine rollierende Auswertung des Produktionsglättungsvorschlags gegenüber der von der Planungsabteilung umgesetzten Lösung, um die Produktionsstabilität zu bewerten.
- Eine Auswertung eines einmaligen Glättungsvorschlags inklusive Vorschau gegenüber eines rollierend angepassten Produktionsglättungsvorschlags, um zu bewerten, ob und in welchem Maß eine rollierende Planung genauer ist.

Im Zuge der Erfüllung der Anforderungen wird das zusätzliche Optimierungspotential durch die Nutzung der Ergebnisse aus der Datenanalyse im Ergebnisteil dieser Arbeit diskutiert.

4.4 Fazit

Während die Produktionsglättung klassisch durch eine Auftragsvorverschiebung bei konstanter Durchlaufzeit beschrieben wird (vgl. Kogan und Tell 2009, S. 223–231), liegt der Fokus der in dieser Arbeit problemspezifisch zu entwickelnden Methodik auf der Vorverschiebung und gleichzeitigen Losgrößenveränderung von Produktionsaufträgen bei (teilweise) variabler, d.h. verlängerter, Durchlaufzeit, siehe Abschnitt 4.2. Die Spezifika des vorliegenden Produktionssystems beinhalten einen ökonomisch variabilisierbaren Produktionsprozess sowie eine dynamische Kapazitätsobergrenze. Außerdem ist die Erstellung einer integrierten Simulation zur Kapazitätsauslastungsvorschau notwendig. Die Anforderungsübersicht in Tab. 4.2 resultiert einerseits aus allgemeinen Anforderungen, die sich v.a. aus dem Anforderungskatalog, siehe Tab. 3.3 bzw. Abschnitt 3.10, zusammensetzen und andererseits aus branchen- und anwendungsfallspezifischen Anforderungen, siehe Abschnitt 4.3, ableiten. In diesem Zusammenhang wird in Bezug auf die Methodenentwicklung in Kapitel 5 insbesondere auf die getrennte Betrachtung von Aktions- und Kurantmengen Rücksicht genommen.

⁵Absolute oder relative Abweichungen vom Sollbestand würden diesen Kenntnisstand pro Artikel nicht aufzeigen, dementsprechend ist hier eine wissensbasierte Implementierung notwendig.

5

Kapitel 5

Entwicklung der Produktionsglättungsmethodik

Kapitel 5 beschäftigt sich mit der Entwicklung der Produktionsglättungsmethodik. Auf Basis der Zielsetzung in Abschnitt 5.1 wird in Abschnitt 5.2 das Zielfunktionssystem abgeleitet und anschließend in Abschnitt 5.3 vollständig aufgestellt. Dieser Schritt entspricht einer klassisch mathematischen Modellbildung gemäß Phase 2 aus dem ORMA. Dabei ist eine integrierte Simulation, die in Abschnitt 5.4 erläutert wird, erforderlich. Abschnitt 5.5 skizziert die Systemintegration der zu entwickelnden Methodik in die IT-Landschaft des industriellen Anwendungsfalles. Anschließend werden gemäß Phase 3 aus dem ORMA die dazugehörigen Optimierungsalgorithmen in Abschnitt 5.6 entwickelt, wobei die Optimierung von einer statistischen Datenanalyse in Abschnitt 5.7 ergänzt wird.

5.1 Zielsetzung und erwartetes Ergebnis der in dieser Arbeit zu entwickelnden Methodik

Die Zielsetzung der vorliegenden Arbeit ist die Erschließung der sich in einem Produktionssystem mit schwankendem Produktionsausstoß ergebenden Kostenreduktionspotenziale durch die Entwicklung und Umsetzung einer hybrid-optimierten, rollierenden Planungsmethodik für eine dynamische, multikriterielle Produktionsglättung am Beispiel der fleischverarbeitenden Industrie. Das Verfahren schlägt am Ende eines Optimierungslaufes einen näherungsweise kostenoptimalen Produktionsplan vor, welcher die kapazitiven Restriktionen bzw. Spezifika der jeweiligen Produktionsartikel einerseits sowie die saisonalen Bestandsvorgaben auf Produktionsartikelebene andererseits berücksichtigt. Die Lösung berücksichtigt außerdem eine entsprechende Losgrößenoptimierung in Bezug auf die Planungsdimension Kalenderwoche. Die optimale Lösung wird den Planer*innen in ihrer Planungsumgebung aufbereitet zur Verfügung gestellt, sodass sie diese direkt übernehmen oder ggf. nachträglich adaptieren können. Dadurch verringert sich auch der periodisch anfallende manuelle Planungsaufwand signifikant. Das entwickelte Verfahren ist zudem in der Lage, aktualisierte Werte für bestimmte (Stammdaten-) Parameter selbständig vorzuschlagen. Dadurch entfällt oder minimiert sich der Anpassungsaufwand für die Aktualisierung von inzwischen veralteten Bestandsstammdaten. Als Folge erhalten die Planer*innen regelmäßig aktualisierte Werte für bestimmte Planungsparameter für die Bestandsreichweitenplanung angezeigt. Dementsprechend reduziert sich die manuelle Stammdatenverwaltung.

Um diese Zielsetzung zu erreichen, wird eine leistungsfähige und effizient implementierte Methodik auf Basis der problemspezifischen Anforderungen mit den folgenden Merkmalen entwickelt:

- Die Methodik fokussiert sich im Betrachtungsschwerpunkt auf die Minimierung der Produktionsschwankungen in Bezug auf das gesamte Produktionssystem.
- Der multikriterielle Ansatz bedingt einerseits Restriktionen (z.B. Produktionsmenge, Kapazität, Personaleinsatz, etc.) und eröffnet gleichzeitig die Erschließung des damit verbundenen Potentials durch die Reduktion des Suchraums.

- Die Kapazitätsauslastung an der Engpassanlage sowie der darauf anschließenden Aggregate stimmt, auf den jeweiligen Planungsdimensionen (Stellplatz bzw. Rack, Planungsperiode), möglichst genau mit der zu erwartenden Realität überein.
- Produktionsaufträge können im Zuge dieser Methodik einerseits zeitlich versetzt als auch hinsichtlich der Auftragsgröße verändert bzw. aufgeteilt und dadurch mathematisch feiner optimiert werden (Losgrößenplanung).
- Die Methodik verarbeitet rollierend gesammelte Daten aus der Historie, generiert Vorschläge für Parameteranpassungen, und erschließt die sich dadurch ergebenden Potentiale.

Abb. 5.1 skizziert beispielhaft und visionär den grundlegenden, schematischen Ablauf der zu entwickelnden Produktionsglättungsmethodik:

1. Erstellung bzw. Erhalt einer in Bezug auf alle zu optimierenden Planungsdimensionen gültigen Basislösung.
2. Ausführung der multikriteriellen Produktionsglättungsmethodik:
 - a) Gezielte Lastspitzenreduktion und Glättung des Produktionsprogramms,
 - b) unter Berücksichtigung der vorhandenen Kapazitäten,
 - c) sowie Optimierung des Bestandsverlaufs je Produktionsartikel.
 - d) Überprüfung und ggf. Adaption optimierungsrelevanter Stammdaten.
3. Erstellung bzw. Konstruktion einer möglichst verbesserten Lösung → Erhalt *optimierter* Glättungsergebnisse sowie Stammdaten durch die Methodik.

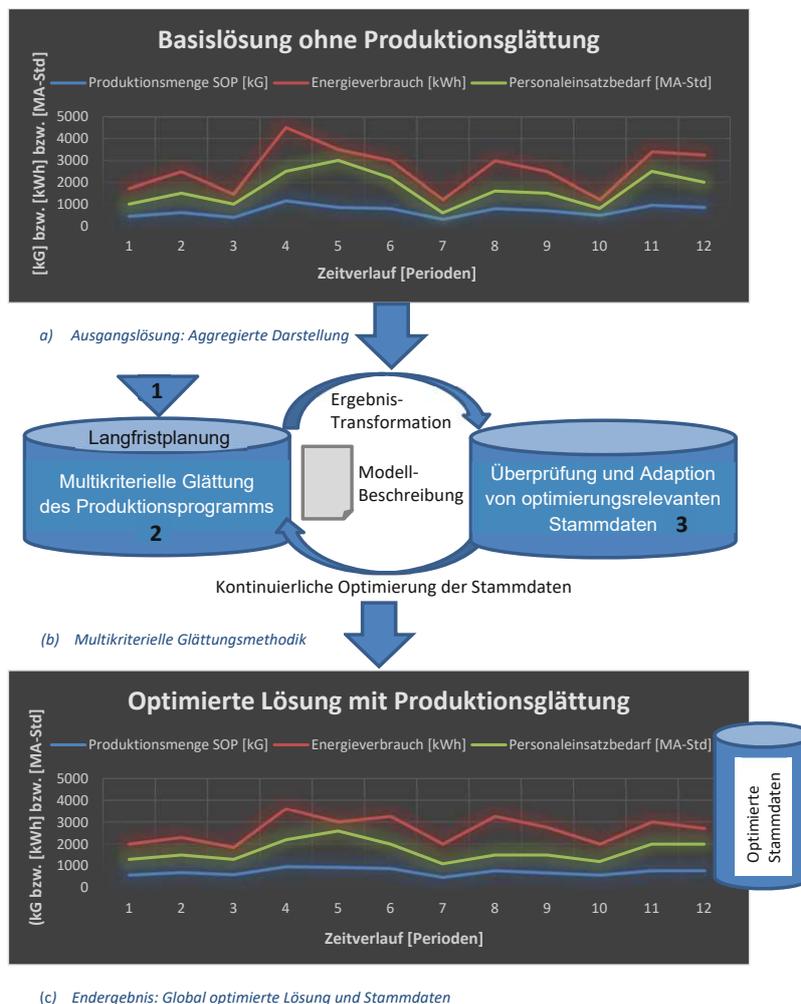


Abb. 5.1: Beispielhafter Ablauf der Produktionsglättungsmethodik (Quelle: eigene Darstellung)

Das Hauptergebnis dieser Arbeit stellt die Entwicklung und Validierung einer effizienten Methodik zur möglichst optimal geglätteten Produktionsplanung dar, welche hochwertige Vorschläge in Bezug auf einen möglichst kostenoptimalen, ausführbaren Produktionsplan, v.a. aus der Sicht einer mittelfristig nivellierten Produktionsauslastung, unter gleichzeitiger Berücksichtigung mehrerer Zielsetzungen, vorsieht. Dieses Vorgehen wird in dieser Arbeit unter dem Begriff multikriterielle Optimierung zusammengefasst. Die multikriterielle Optimierung umfasst die heuristische und metaheuristische Optimierung in Abschnitt 5.6. Basierend auf den Ergebnissen einer statistischen Datenanalyse von Bestandsstammdaten wird in Abschnitt 5.7 ein Konzept für Vorschläge zur Parametrisierung erstellt, um diese Parameter laufend auszuwerten und automatisiert anzupassen.

Zusammenfassend wird das im Zuge dieser Arbeit als Anwendungsfall vorliegende kapazitiv beschränkte Produktionssystem im Rahmen der mittelfristigen Produktionsplanung, welche sich aus vielen Produktionsartikeln und Planungsperioden zusammensetzt, in Bezug auf mehrere Zielsetzungen gleichzeitig derart optimiert, sodass dabei eine möglichst gleichmäßige Produktionsausstoßrate erreicht wird. Neben der Minimierung dieser Produktionsausstoßschwankungen auf Produktionartikel- und Werksebene besteht eine weitere zentrale Zielsetzung aus der möglichst gleichmäßigen Auslastung aller zur Verfügung stehenden Produktionskapazitäten.

5.2 Ableitung des konkreten Zielfunktionssystems

Das konkrete Zielfunktionssystem leitet sich aus der zentralen zu beantwortenden Problemstellung im Einklang mit den Zielen dieser Arbeit ab. Die Kernaufgabe besteht darin, einen möglichst gleichmäßig ausgelasteten mittel- bis langfristigen Produktionsplan¹ rollierend für alle Produktionsartikel des neuen Werks zu erstellen. Der Plan berücksichtigt die Spezifika des vorliegenden Produktionssystems entsprechend den Anforderungen des*der industriellen Anwendungspartner*in, welche einerseits statische als auch dynamische sowie artikelspezifische Nebenbedingungen umfassen. In Zusammenarbeit mit der Planungsabteilung und Informationstechnologie wurden die folgenden Optimierungskriterien definiert und entwickelt:

1. Minimierung der Lastspitzen pro Planungsperiode und Produktionsartikel auf der Einsalzanlage: Dabei werden Aktions- und Kurantmengen getrennt behandelt.
2. Minimierung der Lastspitzen über alle Produktionsartikel (d.h. auf Werksebene kumuliert) pro Planungsperiode in Bezug auf die Einsalzanlage für den gesamten Planungshorizont.
3. Optimierung der Kapazitätsauslastung für die Einsalzanlage sowie sämtliche nachgeschaltete Produktionsanlagen in der Grundproduktion.
4. Optimierung der Bestandsniveaus durch Annäherung an definierte Soll-Bestandsniveaus.
5. Minimierung der Anzahl an Rüstvorgängen für den gesamten Planungshorizont.

Die ersten beiden Kriterien unterscheiden sich hinsichtlich der Betrachtungsebene. Während das erste Kriterium eine möglichst gleichmäßige Produktionsausstoßrate für einen jeden Artikel pro Periode fordert, setzt sich das zweite Kriterium eine nivellierte Produktionsausbringungsrate auf Werksebene zum Ziel. Diese beiden Teilziele haben i.d.R. einen positiven Überlagerungseffekt, obwohl sie unterschiedliche Zielsetzungen verfolgen. Das erste Teilziel verfolgt eine möglichst stetige Beschaffung und Verarbeitung eines jeden einzelnen Produktionsartikels, während das zweite Teilziel die Zielsetzungen des ersten Teilziels auf das gesamte Werk übersetzt. Das dritte Teilziel zielt in der Betrachtung, gemäß Goldratt, darauf ab, den Fokus nicht ausschließlich auf den aktuellen Produktionsengpass zu legen, sondern auch mögliche neu entstehende Produktionsengpässe vorausschauend zu bewerten und bestmöglich zu minimieren (vgl. Lenort et al. 2012, S. 225). Eine optimierte Kapazitätsauslastung bedeutet für den konkreten industriellen Anwendungsfall, dass die der Einsalzanlage nachgelagerten Produktionsaggregate ebenfalls möglichst gleichmäßig ausgelastet werden unter der Restriktion, dass sie dabei nicht selbst

¹In diesem industriellen Anwendungsfall werden, bedingt durch die relativ lange, prozessbedingte Durchlaufzeit der gesamtheitlich betrachteten Grundproduktion, 78 Wochen als Planungshorizont definiert.

zum Engpass werden. Das vierte Teilziel sieht vor, dass für jeden Produktionsartikel nach Möglichkeit die Soll-Bestandsniveaus in jeder Produktionsperiode erreicht werden. Das fünfte Teilziel bewertet und minimiert die Anzahl an notwendigen Rüstvorgängen in Bezug auf den betrachteten Planungshorizont.

Die multikriterielle Optimierungsmethode genügt den vorliegenden Produktionsspezifika in Bezug auf die Werkskapazitäten, die Produktionsprozesse und Arbeitspläne, die Spezifika der Produktionsartikel, die spezifischen Bestandsniveaus und Rüstzeiten. Der multikriterielle Optimierungsalgorithmus führt unter Berücksichtigung der Teilziele (1) und (2) zu einer Reduktion der Lastspitzen in der Produktion. Die Methode ist außerdem in der Lage, durch die Berücksichtigung von Teilziel (3) Schwankungen der Kapazitätsnachfrage über alle betrachteten Produktionsstufen hinweg auszugleichen. Dies resultiert in einem ressourceneffizienteren Produktionssystem. Eine Produktionsglättungsmethodik, welche die Lastspitzen effizient reduziert, verlängert darüber hinaus auch die Zeitspanne, bis Investitionen in neue und kostspielige Anlagen notwendig werden. In Bezug auf die Bestandsoptimierung muss die Methodik in der Lage sein, den Fokus auf einen wesentlichen Ausschnitt zu legen. Die Rüstzeiten werden im Zuge einer mittel- bis langfristigen rollierenden Erstellung eines Produktionsplans nicht exakt im Voraus kalkuliert. Deshalb werden diese von der Methode mit einer akzeptablen Genauigkeit berücksichtigt.

Eine zentrale Herausforderung für die Planungsaufgabe stellt die Variation der artikelspezifischen prozessbedingten Durchlaufzeiten innerhalb der mehrstufigen Produktion (siehe Abb. 4.1) dar. Die fest vorgegebenen artikelspezifischen, individuellen Teilprozessdurchlaufzeiten führen i.d.R. zu einer unterschiedlichen Kapazitätsauslastung in der Grundproduktion. Dies begründet sich einerseits durch die verschiedenen Arbeitspläne der Artikel selbst oder durch eine Änderung des Produktionsvolumens bzw. des dazugehörigen Produktmixes pro Periode und Prozessschritt. Ein weiterer Grund liegt in der zeitlichen Vorverschiebung von Produktionsmengen von einer Periode in eine andere. Die Kapazitätsauslastung für die nachgeschalteten Produktionsaggregate wird über eine integrierte, dynamische Kapazitätsauslastungssimulation ausgewertet.

5.3 Notation des Zielfunktionssystems

Das Zielfunktionssystem lässt sich aus den bereits definierten Optimierungskriterien bzw. Zielsetzungen, den dazugehörigen Nebenbedingungen, und damit verbunden einer aufzustellenden Zielfunktion mathematisch formulieren (vgl. Kamhuber et al. 2019, S. 2110ff.; Kamhuber et al. 2020, S. 5f.). Zuvor wird an dieser Stelle die im Zuge dieser Arbeit verwendete Notation erläutert, siehe die Tab. 5.1 - Tab. 5.3. Tabelle 5.1 beschreibt die verwendeten, abhängigen Variablen und Indizes für das zu definierende Zielfunktionssystem, untergliedert in die auf Basis der Optimierungskriterien formulierten Teilzielfunktionen (TZF) $f_1 - f_5$. Die wöchentlichen Produktionsmengen der Artikel p_{ij} bzw. die Werksproduktionsmenge q_i werden jeweils in Bezug zur Engpassanlage betrachtet, während die Kapazitätsauslastung c_{ik} pro Periode auf Kapazitätsgruppenebene bewertet wird. Tabelle 5.2 beschreibt sämtliche im Rahmen der durch Nebenbedingungen beschränkten Optimierung verwendeten Variablen. Tabelle 5.3 beinhaltet eine Übersicht aller verwendeten Matrizen zur Darstellung der Lösung.

Tab. 5.1: Verwendete Notation für das Zielfunktionssystem (vgl. Kamhuber et al. 2020, S. 5)

Variablen und Indizes	Beschreibung
$\omega_1 - \omega_5$	Individuelle Gewichtungen der einzelnen Teilziele
$s_1 - s_5$	Teilzielspezifische Skalierungsfaktoren
W	Gewichtungsvektor für $\omega_1 - \omega_5$, $W = [\omega_1, \omega_2, \omega_3, \omega_4, \omega_5]$
i, j, k	Indizes für Periode (Woche), Produktionsartikel, Kapazitätsgruppe
m, n, o	Gesamtanzahl an Produktionsartikeln, Perioden und Kapazitätsgruppen
p_{ij}	Produktionsmenge p von Artikel j in Periode i
p_{ijt}	Produktionsversatzmenge von Artikel j in Periode i um t Periode(n)
p_{ijz}	Produktionsmenge p_{ij} , die noch um z Periode(n) vorversetzbar ist
q_i	Werksproduktionsmenge q aller m Artikel in Periode i
c_{ik}	Kapazitätsauslastung c in Periode i auf Kapazitätsgruppe k
s_{ij}	Bestandswirksame, fertig produzierte Menge s von Artikel j in Periode i
$f(p_{ij}, q_i, c_{ik}, s_{ij})$	Die aufgestellte Zielfunktion f in Abhängigkeit ihrer Variablen
$f_1(p_{ij})$	TZF 1: Produktionsartikelgradient in Periode i für Artikel j
$f_2(q_i)$	TZF 2: Werksgradient in Periode i
$f_3(c_{ik})$	TZF 3: Kapazitätsauslastungsgradient in Periode i für Kapazitätsgruppe k
$f_4(s_{ij})$	TZF 4: Bestandsverlaufsgradient s in Periode i für Artikel j
$f_5(p_{ij})$	TZF 5: Anzahl an Rüstvorgängen in Periode i für Artikel j

Tab. 5.2: Verwendete Notation für die Variablen betreffend die Optimierung und Nebenbedingungen (vgl. Kamhuber et al. 2020, S. 5)

Variablen	Beschreibung
q_{ik}	Maximal zulässige Werksproduktionsmenge q in Periode i auf Kapazitätsgruppe k
q_{max_i}	Maximal zulässige (dynamische) Werksproduktionsmenge q in Periode i
$c_{max_{ik}}$	Maximal zulässige (statische) Raumkapazität c von Kapazitätsgruppe k in Periode i
$t_{p_{ij}}$	(Aktuelle) Anzahl an Versatzperioden t von Produktionsmenge p_{ij}
t_{max_j}	Maximal zulässige (ganzzahlige) Anzahl an Versatzperioden t_{max} von Artikel j
s_{ij}	Bestandswirksame (fertig produzierte) Menge s von Artikel j in Periode i
$S_{opt_{ij}}$	(Dynamisches) Zielbestandsniveau S_{opt} von Artikel j in Periode i
S_{min_j}	(Statisches) Mindestbestandsniveau S_{min} von Artikel j
α	Abkühlungsgeschwindigkeit bzw. Ausglührate (SA)
I_T, T_{init}, T_{min}	Wiederholungen pro Temperatur, Starttemperatur, Endtemperatur (SA)

Tab. 5.3: Matrizen zur Darstellung der Lösung (vgl. Kamhuber et al. 2020, S. 5)

Lösungsmatrix	Beschreibung
ppm	Produktionsplanmatrix (beinhaltend sämtliche Produktionsartikelmengen p_{ij})
stm	Bestandsverlaufsmatrix (beinhaltend sämtliche Bestandsmengen s_{ij})
cum	Matrix zur Vorschau des Kapazitätsauslastungsverlaufs pro Raumgruppe und Periode
vm	Versatzmatrix zur Darstellung der (bereits) versetzten Produktionsmengen p_{ijt}

5.3.1 Zielfunktionssystem: Teilzielfunktionen und Nebenbedingungen

Im Zuge von Phase 2 des ORMA, siehe auch 1.5.1, wird ein mathematisches Modell für das vorliegende Problem in Abstimmung mit dem*der industriellen Anwendungspartner*in formuliert. Das Modell besteht aus einer kostenbasierten Zielfunktion, welche auf eine heuristisch optimierte Basislösung von der jeweiligen Metaheuristik (GA, SA) angewendet wird. Diese Zielfunktion kann auch für die Heuristik

ausgewertet werden, um vollständige Zielfunktionsverläufe zu erstellen. Dadurch werden die Ergebnisse der Methodik ganzheitlich bewertet und visualisiert dargestellt. Die Zielfunktion besteht, gemäß den Kriterien aus Abschnitt 4.3, aus fünf absteigend sortiert gewichteten ($\omega_1 - \omega_5$) Teilzielen, welche unterschiedliche Aspekte der Produktionsplanung betreffen: Die Hauptzielsetzungen einer geglätteten Produktion werden über zwei Teilziele (f_1, f_2) sichergestellt und bewertet. Eine möglichst gleichmäßige Raumkapazitätsauslastungsvorschau der gesamten Grundproduktion (siehe Abb. 4.1) wird mittels f_3 evaluiert. Die Gleichmäßigkeit des Bestandsverlaufs gegenüber dem definierten Zielbestandsniveau wird mittels f_4 bewertet. Der letzte Gesamtzielfunktionsbestandteil, f_5 , berücksichtigt die Bewertung der minimal erforderlichen Rüstvorgänge für den jeweiligen Produktionsplan. Die vollständige Zielfunktion f , dargestellt in Gleichung 5.1², skalarisiert und überträgt das multikriterielle Problem in ein singlekriterielles Problem durch die Berechnung eines gewichteten und skalierten Fitnesswerts. Die Skalierung erfolgt in der Metaheuristik auf Basis der Teilzielfunktionsbewertungen der besten durch die Heuristik gefundenen Lösung.

$$\begin{aligned} \text{Minimiere } f(p_{ij}, q_i, c_{ik}, s_{ij}) = & \\ \frac{\omega_1}{s_1} \sum_{i=1}^n \sum_{j=1}^m f_1(p_{ij}) + \frac{\omega_2}{s_2} \sum_{i=1}^n f_2(q_i) + \frac{\omega_3}{s_3} \sum_{i=1}^n \sum_{k=1}^o f_3(c_{ik}) + & \quad (5.1) \\ \frac{\omega_4}{s_4} \sum_{i=1}^n \sum_{j=1}^m f_4(s_{ij}) + \frac{\omega_5}{s_5} \sum_{i=1}^n \sum_{j=1}^m f_5(p_{ij}), & \end{aligned}$$

$$\text{Gemäß } \sum_{j=1}^m p_{ij} \leq q_{max_i}, \quad \forall i \in \{1, \dots, n\}, \quad (5.2)$$

$$\sum_{k=1}^o c_{ik} \leq c_{max_{ik}}, \quad \forall i \in \{1, \dots, n\}, \quad (5.3)$$

$$t_{p_{ij}} \leq t_{max_j}, \quad \forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\}. \quad (5.4)$$

Die Teilzielfunktionen $f_1 - f_4$ werden mittels Absolutabweichungen jeweils benachbarter Perioden bzw. dem Ist- zum Zielbestandsniveau dargestellt und ausgewertet, während f_5 über eine boolesche Gleichung berechnet wird:

$$f_1(p_{ij}) = Abs(p_{i+1,j} - p_{ij}), \quad (5.5)$$

$$f_2(q_i) = Abs(q_{i+1} - q_i), \quad (5.6)$$

$$f_3(c_{ik}) = Abs(c_{i+1,k} - c_{ik}), \quad (5.7)$$

$$f_4(s_{ij}) = Abs(s_{ij} - S_{opt_{ij}}), \quad (5.8)$$

$$f_5(p_{ij}) = \begin{cases} 0, & \text{falls } p_{ij} = 0, \\ 1, & \text{falls } p_{ij} > 0, \end{cases} \quad (5.9)$$

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\}, \forall k \in \{1, \dots, o\}. \quad (5.10)$$

Die Teilzielfunktionen $f_1 - f_3$ werden durch die Absolutdifferenzen der jeweiligen Variablen (Artikelproduktionsmenge, Werksproduktionsmenge, Kapazitätsauslastung) zwischen den jeweils angrenzenden Planungsperioden innerhalb des gesamten Planungshorizonts berechnet, während bei f_4 laufend der aktuelle Bestand s_{ij} mit dem Zielbestand $S_{opt_{ij}}$ verglichen wird. Durch die Verwendung von Absolutdifferenzen ist gewährleistet, dass eine positive als auch negative absolute Veränderung, z.B. der Produktionsausstoßrate für f_1 bzw. f_2 , gleichermaßen im Zuge der Optimierung bestraft wird. Dies

²die entsprechende (Java-) Methode zur Bewertung der Zielfunktion befindet sich ausformuliert im Anhang A.16.

entspricht auch exakt dem Streben der Planung, die einen Produktionsvorschlag mit einer möglichst gleichmäßigen Produktionsausstoßrate sowohl auf Artikelebene als auch auf Werksebene in Bezug auf die identifizierte Engpassanlage (Einsalzprozess) erhalten möchte. Im Zuge dieser Arbeit werden die Absolutabweichungen in $f_1 - f_4$ als Gradienten der jeweiligen Variable bezeichnet. Der Werksgradient bezieht sich dabei (wie der Artikelgradient auch) auf die periodische Ausbringungsmenge der Einsalzanlage (pro Artikel bzw.) als Summenbetrachtung über alle Artikel, d.h.: $q_i = \sum_{j=1}^m p_{ij} \forall i \in \{1, \dots, n\}$.

Teilziel f_3 benötigt das Ergebnis einer integrierten, dynamischen Kapazitätsauslastungssimulation. Diese gibt den Auslastungsverlauf der einzelnen Produktionskapazitätsgruppen (siehe Abb. 4.1) in Bezug auf den entsprechenden Produktionsvorschlag an der Einsalzanlage zurück.

Die Produktionsglättungsmethodik sieht vor, dass die Optimierung über eine Minimierung der Absolutdifferenzen der Teilzielfunktionen f_1 bis f_3 erfolgt. Der Bestandsgradient, f_4 , ergibt sich aus der Differenz zwischen dem je Periode aktuellen Fertigwarenbestand³ pro Artikel und dem dazugehörigen dynamisch gepflegten Ziellagerbestand für diese Periode. Die ersten vier Teilziele des aufgestellten Zielfunktionsmodells, $f_1 - f_4$, sind in den Gleichungen (5.5) – (5.8) definiert und basieren auf der gleichen Maßeinheit ([kg]) und begünstigen dadurch die Anwendung der Gewichteten-Summen-Methode. Das Teilziel f_5 ist eine vereinfachte Funktion zur Abschätzung der zu erwartenden (Um-) Rüstkosten. Gemäß Gleichung (5.9) ist in einer Produktionsperiode für einen Artikel mindestens einmal zu rüsten, wenn der entsprechende Artikel in dieser Periode Produktionsmengen aufweist. Für den Fall, dass in einer Periode keine Produktionsmengen eines Artikels vorliegen, muss dafür entsprechend nicht (um-) gerüstet werden. Das Teilziel f_5 kann im Zuge der Mittel- und Langfristplanung nicht genauer ermittelt werden, weil dazu eine Feinplanung (z.B. auf Tagesebene) notwendig ist, welche die Anlieferzyklen innerhalb einer Woche sowie diverse weitere Umstände detailliert berücksichtigt. Für die Berücksichtigung der einzelnen Skalierungsfaktoren $s_1 - s_5$ in der anschließenden Metaheuristik werden die jeweiligen Teilziele der besten, durch die Heuristik ermittelten, Lösung verwendet. Der mit der Planungsabteilung abgestimmte Gewichtungsvektor, der die Präferenzen des Managements am besten widerspiegelt, lautet: $W = [1.25, 1.25, 1, 1, 0.5]$. Dieser wurde nach mehreren Testläufen mit verschiedenen Parametrisierungen als der die Präferenzen am besten ausdrückende Vektor ermittelt.

5.3.2 Berücksichtigung von Nebenbedingungen im Zuge der Optimierung

In Abhängigkeit von den Werktagen einer Woche, der geplanten Verfügbarkeit von Mitarbeiter*innen und des verwendeten Schichtmodells ergibt sich eine periodisch-dynamische, kapazitive Leistungsgrenze in Bezug auf die Engpassanlage des Werks. Diese Leistungsgrenze wird in Form einer maximal möglichen, dynamischen Gesamtproduktionsmenge für jede Woche für dieses Aggregat angegeben, sodass eine Lösung nur gültig ist, sofern die zu produzierende Wochenmenge unterhalb dieser Kapazitätsgrenze liegt. Nebenbedingung (5.2) betreffend die Kapazitätsbeschränkung an der Engpassanlage muss in jeder Periode strikt eingehalten werden, um eine gültige Lösung, d.h. einen realisierbaren Produktionsplan, zu generieren. Die einzige Ausnahme bildet⁴ dabei die von der Planung über die Datenbank zur Verfügung gestellte Basislösung: Wenn die von der IT übergebene Lösung in einer Periode bereits einen Verstoß gegen diese Nebenbedingung generiert, muss es der Optimierung nicht mehr zwangsläufig gelingen, diesen Verstoß an dieser Stelle vollständig zu eliminieren. In dieser Arbeit wird dafür eine Strategie verwendet, die der in Unterabschnitt 3.3.2 vorgestellten Nischenselektionsstrategie von Jimenez (vgl. Jimenez et al. 2002), beschrieben in dem Beitrag von Konak (vgl. Konak et al. 2006, S.1003f.), inhaltlich sehr nahe kommt. Der Hauptunterschied besteht prinzipiell darin, dass nur jeweils eine heuristisch weiterentwickelte Lösung jeweils mit der Basislösung verglichen wird, ob der Grad des Verstoßes der aktuellen Lösung sich entsprechend ggü. der Initiallösung reduziert. Die Zielfunktion bedingt in diesem Anwendungsfall bereits, dass sich mit einer besseren Zielfunktionsbewertung von f_1 und f_2 bzw. resultierend f automatisch auch der dazugehörige Verstoß gegenüber (5.2) insgesamt reduziert. Dies erklärt sich durch das Konzept der regelbasierten Glättung (siehe Abb. 5.5) und dazugehörig der

³ex Reifelager: Das bedeutet, dass die Mengen den Reifelagerprozess bereits abgeschlossen haben müssen und noch kundenanonym für die weitere Verarbeitung vorliegen.

⁴nach Rücksprache mit der Planung und IT

Losteilung, wodurch automatisiert Mengen bevorzugt in auslastungsschwächere Wochen vorverschoben werden. Dadurch reduziert sich auch der Grad der Verletzung, wenn er an einer oder mehreren Stelle(n) innerhalb einer Lösung (bereits) auftritt.

Der entwickelte Letztstand ermöglicht eine gesonderte Abfrage der Nebenbedingung in Bezug auf Aktions- und Kurantmengen, wobei Aktionsmengen bei einer Überschreitung der Kapazitätsgrenze nicht in eine Woche hineinverschoben werden dürfen, während dies für Kurantmengen ggf. ermöglicht wird. Dies liegt v.a. an den unterschiedlichen, max. erlaubten Verschiebezeiträumen für Aktions- und Kurantmengen, siehe Nebenbedingung (5.4), wodurch Aktionsmengen mehr Verschiebepotential erhalten als Standardmengen.

Die Gründe für diese Ausnahmeregelung liefern u.a. weitere Nebenbedingungen, welche es z.B. nicht ermöglichen würden, eine Belastungsspitze abzubauen. In diesem Fall wird der Verstoß gemäß der Strategie bestmöglich reduziert und in der Lösung angezeigt, wenn die Erstellung einer gültigen Lösung nicht möglich ist⁵. Darüber hinaus legt diese Beschränkung im Sinne der Planung fest, dass die Produktionsmengen nur bestimmte, ganzzahlige Werte entsprechend einer A/B/C-Kategorisierung der Artikel annehmen, um volle (A-Artikel) oder halbe (B-Artikel) bzw. angefangene (C-Artikel) Palettenstellplätze, sogenannte *Racks*, zu erstellen. Der Hauptgrund für diese Logik - zusätzlich zur Stabilisierung der Produktionsmengen innerhalb jeder Periode - liegt in der Optimierung der Lager- und Palettenstellplatzkapazitäten in den einzelnen Produktionsschritten: Je besser die Racks in jedem Raum gefüllt sind, desto geringer fällt die Auslastung der entsprechenden Kapazitätsgruppe und die mit der Ein- und Auslagerung verbundene (mittels fahrerlosen Transportsystemen automatisierte, in begrenztem Umfang mögliche) Anzahl an Lagerspielen aus.

Die Nebenbedingung (5.3) prüft, ob die Anzahl der benötigten Racks für jede Periode und jeden Grundproduktionsprozess die entsprechend statisch definierte, maximale Rack-Kapazität des jeweiligen Prozesses nicht überschreitet. Die Nebenbedingung ist sowohl im Zuge der Heuristik als auch Metaheuristik global einstellbar, d.h. deaktivierbar als auch aktivierbar, gestaltet. Eine Aktivierung bedeutet, dass eine Änderung im Falle eines Verstoßes nicht durchgeführt wird. Dies entspricht Strategie 1 aus Abschnitt 3.7. Im Zuge der Betrachtung von (un-) möglichen Produktionsszenarien ist ein Überschreiten dieser Nebenbedingung in einzelnen Perioden durch die Optimierung nicht ausgeschlossen. Dementsprechend haben sich die Planung und die IT für eine (vorläufige) Deaktivierung der Nebenbedingung entschieden. Nachdem die Optimierung - in Analogie zu Nebenbedingung (5.2) - von selbst keine Modifikationen an den Stellhebeln der Optimierung bzw. an der Basislösung vornehmen kann, werden Perioden mit einer Überlastung der Rack-Kapazität entsprechend gekennzeichnet ausgegeben. Die Planungsabteilung möchte im Zuge der (teilweise manuellen) abschließend Erstellung einer als für gültig empfundenen Lösung selbst entscheiden, ob das Schiebefenster für einen oder mehrere Artikel verlängert oder ausnahmsweise zu produzierende Mengen von einem oder mehreren Artikeln reduziert und stattdessen vom Puffer gezogen wird, oder die zusätzlichen Mengen auf eine andere Art und Weise berücksichtigen möchte. Der Umgang mit dieser Nebenbedingung entspricht also der manuellen Form der Strategie 4 aus Abschnitt 3.7. Es besteht auch die Möglichkeit, die Nebenbedingung im Zuge der Optimierung in der Form zu aktivieren, sodass keine Modifikationen ausgeführt werden, welche eine Verletzung der Nebenbedingung mit sich bringen würden.

Sowohl für Nebenbedingung (5.2) als auch (5.3) wurde im Zuge von Expert*innenbefragungen mit der Planungsabteilung eine automatisierende Berücksichtigung im Rahmen der Optimierungsmethode angeboten: Dies umfasst konkret die Definition einer Hierarchie, nach welcher automatisiert das Schiebefenster ausgedehnt wird oder die Produktionsmenge entsprechend der Hierarchie gekürzt wird. Diese Automatisierung möchte die Planungsabteilung jedoch nicht allgemein formuliert haben, sondern angepasst an die jeweilige Situation selbständig treffen.

Nebenbedingung (5.4) legt fest, dass aus Qualitätsgründen jede Produktionsartikelmenge p_{ij} maximal um eine bestimmte (vordefinierte) Anzahl an Perioden t_{max_j} vorverschoben werden darf. Dieser gemeinsam durch die Planung und IT definierte Grenzwert bestimmt somit für jeden Artikel, dass

⁵Der Hauptgrund dafür liegt in der manuellen Berücksichtigung von ggf. längeren Schichtzeiten im Notfall, sodass diese durch die Planung bereits in der Basislösung vorgesehen ist, in Ausnahmesituationen über das definierte (Kapazitäts-) Limit hinaus zu produzieren.

eine Produktionsmenge, ausgehend von einer Originalperiode, den maximal möglichen, definierten Versatz in Bezug auf eine andere, zeitlich betrachtet frühere Periode, nicht überschreitet. Diese hart definierten Grenzwerte werden sowohl für Standard- als auch für Aktionsvolumina definiert, so dass jede Mengendimension gesondert behandelt wird. Dies entspricht exakt der Strategie 3 aus Abschnitt 3.7.

5.4 Modellierung der Kapazitätsauslastung: Integrierte dynamische Simulation

Gemäß Abschnitt 5.3 erfordert das Teilziel f_3 aus Formel (5.1) für die Zielfunktion eine Bewertung durch eine integrierte, dynamische Simulation. Das primäre Ziel dieses diskreten Simulationsmoduls besteht darin, eine möglichst realistische und akkurate dynamische Kapazitätsauslastungsvorschau (auf Ebene eines Racks bzw. Palette) für die dem Einsalzen nachgelagerten Kernprozesse der Grundproduktion zu erstellen, siehe Abb. 4.1. Im Zuge dieser Vorschau wird auch die Nebenbedingung (5.3) geprüft und ein entsprechender Verstoß in der Lösung (einerseits bereits durch eine höhere Gradientenabweichung bestraft) ausgegeben. Diese Prozesse sind durch die jeweils maximal verfügbare Raumkapazität begrenzt. Wie in Abbildung 5.2 dargestellt, kommen die auf Racks gepackten Artikel aus dem entsprechenden vorgelagerten Prozess. Die Modellierung dieser Simulation fasst verschiedene Artikel, die zu derselben Artikelgruppe gehören, auf einem Rack zusammen, bevor sie in prozessspezifischen Produktionsräumen gemäß Arbeitsplan weiterverarbeitet werden. Danach gelangen sie in den nächsten nachgeschalteten Prozess, wo sie entsprechend ihres individuellen Arbeitsplans und ihrer Artikelgruppe wieder neu gruppiert werden. Diese Gruppierung findet in der Realität mehrmals unter der Woche statt und stellt das heuristische Wissen der Mitarbeiter*innen dar. Im Zuge dieser Heuristik kann jeweils eine kurze Zeitspanne (z.B. 1 – 2 Werkzeuge) nach der initialen Rack-Bildung die ursprüngliche Rack-Auslastung noch verbessert, d.h. erhöht, werden, wenn Mengen von Produktionsartikeln derselben Artikelgruppe produziert werden. Die Simulation bildet direkt den optimalen Rack-Auslastungsgrad, der sich zuzüglich aller möglichen Einlagerungen sowie abzüglich aller Auslagerungen ergibt, ab. Das Simulationsmodul wird für den gesamten Planungshorizont ausgeführt und nutzt die aktuell in jeder Raumgruppe befindlichen Produktionsmengen zur Erstellung einer Ausgangslösung. Diese Lösung wird nun verwendet, um eine Simulation für den dazugehörigen Produktionsplan pro Periode durchzuführen. Die Ergebnisse umfassen neben einer Teilzielfunktionsbewertung von f_3 insbesondere die prognostizierte Auslastung aller wesentlichen, nachgelagerten Grundproduktionsprozesse im Zeitverlauf. Diese Simulation liefert die Grundlage für die Bewertung von verschiedenen Planungsszenarien, v.a. hinsichtlich der jeweiligen kapazitiven Umsetzbarkeit (vgl. Kamhuber et al. 2019, S. 2113).

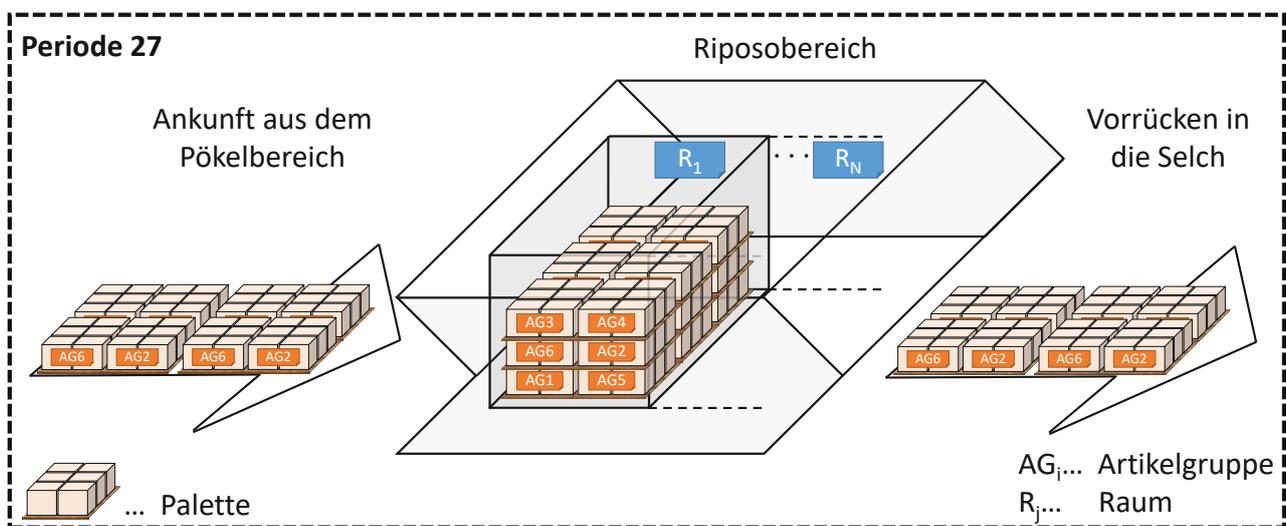


Abb. 5.2: Dynamische Simulation zur Erstellung einer Kapazitätsauslastungsvorschau (vgl. Kamhuber et al. 2019, S. 2113)

5.5 Systemintegration und technische Anbindung

Die Systemintegration und technische Anbindung, siehe Abb. 5.3, zeigt die Einbindung der zu entwickelnden rollierenden Planungsmethodik an die grundsätzliche Ablaufplanung bei dem*der industriellen Anwendungspartner*in. Das Ergebnis der rollierend auszuführenden Produktionsglättung stellt eine rollierend geglättete und aktualisierte Produktionsplanung inklusive einer dazugehörigen Kapazitätsauslastungsvorschau dar. Das Pufferlager wird laufend einer Bestandsrisikoanalyse unterzogen und ggf. entsprechend angepasst. Die Ergebnisse der Methodik werden den Produktionsplaner*innen sowie dem Frischfleisch-Einkauf laufend zur Verfügung gestellt. Der Einkauf möchte auch entsprechende *Szenarien* bezüglich einer Optimierung seiner Einkaufsstrategie in der Planung vorausschauend kostentechnisch bewertet abbilden. Im Zuge der rollierenden Planung wird laufend die tatsächlich beauftragte Ist-Produktionsmenge sowie der dazugehörige jeweils aktuelle Ist-Bestand einbezogen. Das Planungstool wird in Java implementiert und (als ausführbare .jar-Datei) an eine Oracle Datenbank-Umgebung sowie an die Planungsstammdaten sowie den für die Planung konkret verwendeten *Speckplan* (siehe Abb. 5.3) eingebunden. Im Speckplan wird die Planung in einer grafischen Benutzeroberfläche entsprechend für die Verwendung übersichtlich auf Produktionsartikelebene pro Periode aufbereitet.

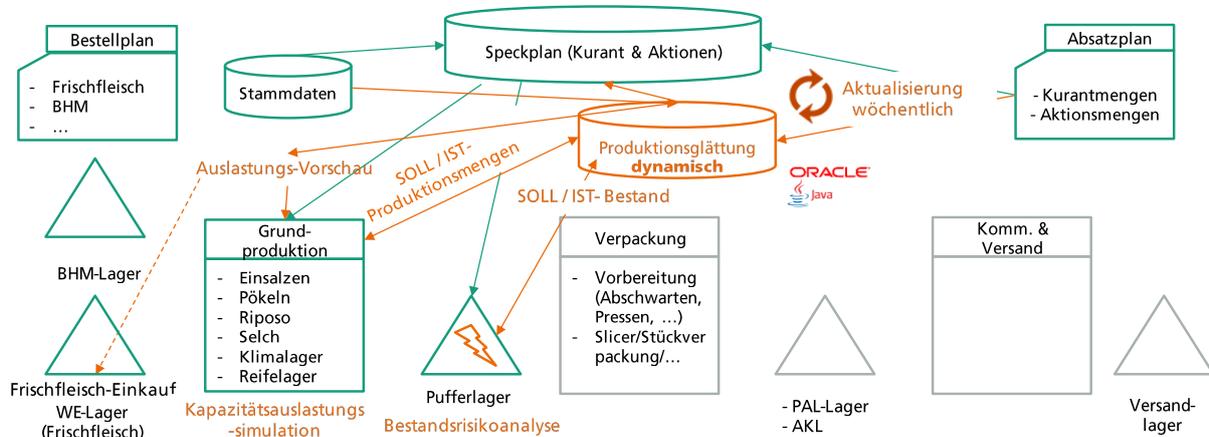


Abb. 5.3: Systemintegration und technische Anbindung (Quelle: eigene Darstellung)

Die Systemintegration hat dabei wenig Einfluss auf die methodische Entwicklung selbst. Es werden die Schnittstellen, v.a. die zu lesenden bzw. zu beschreibenden Datenbank-Tabellen, definiert. Der Einleseprozess umfasst neben den Bewegungsdaten (zukünftige Produktionsstartmengen gemäß Absatzplan) Artikel- und (statische bzw. dynamische) Kapazitätsstammdaten, Arbeitspläne für alle Artikel und die Übermittlung der initialen Kapazitätsauslastung pro Raumgruppe auf Artikelebene. Die Ausgabe bzw. der schreibende Prozess umfasst die Übergabe der geglätteten Lösung, die Rückmeldung der aktiven Bestandsstrategie sowie den dazugehörigen, aktualisierten Bestandsverlauf und die, gemäß Glättungsergebnis, aktualisierte Raumkapazitätsauslastung. Außerdem wird ein Übergabezeitpunkt spezifiziert. Dies bedeutet, dass die Methode in jeder Periode an einem bestimmten Zeitpunkt automatisch ausgeführt wird und auf einem zu diesem Zeitpunkt aktuellen Datenstand zurückgegriffen wird. Durch die Glättung werden jedes Mal erneut in der Zukunft liegende Produktionsmengen zeitlich vorversetzt, aufgeteilt und entsprechend verschoben. Für bereits beauftragte Produktionsmengen, siehe auch den orange markierten Bereich von Abb. 4.1, werden von dem*der industriellen Anwendungspartner*in geglättete Produktionsmengen in den Datenbank-Tabellen entsprechend gekennzeichnet, damit für die Methode klar ist, dass diese Mengen mit einer entsprechend längeren Durchlaufzeit versehen worden sind. Diese Berücksichtigung ist kritisch, weil der Methodik dadurch angegeben wird, welche Mengen von einer jeden Woche tatsächlich geglättet, d.h. vorversetzt mit verlängerter Durchlaufzeit gestartet, worden sind bzw. welche Mengen mit der regulären Durchlaufzeit beauftragt wurden.

5.6 Entwicklung des hybriden Optimierungsverfahrens

Metaheuristiken wurden in der Vergangenheit erfolgreich auf komplexe Produktionsplanungsprobleme angewandt, wie in Kapitel 3 aufgezeigt wird. Bei umfangreichen Optimierungsaufgaben mit einem ähnlichen Komplexitätsgrad zeigen die Autoren, dass Metaheuristiken in alleiniger Ausführung vor der Herausforderung stehen, gute bzw. Pareto-optimale Lösungen in einem riesigen Suchraum in angemessener Zeit⁶ zu finden, um für eine reale rollierende Planung der wöchentlichen Produktionsmengen relevant zu sein. In den dort beschriebenen Anwendungsfällen sind die Implementierung eines gezielten Suchmechanismus innerhalb der Metaheuristiken und die Eingrenzung des Suchraumes mittels Nebenbedingungen durch den Einbezug von Wissen über das zugrundeliegende, reale System vielversprechende Mittel, um sowohl die Leistung als auch die Qualität der Optimierungsergebnisse erheblich zu verbessern (vgl. Sobottka et al., 2018, S. 413-420; Sihm et al., 2018, S. 447ff.).

Für die vorliegende Aufgabenstellung ist der Einbezug von Wissen über das Produktionssystem nicht nur ein Mittel zur Leistungssteigerung, sondern auch zur Gewährleistung, dass die Optimierungsergebnisse vertrauenswürdig sind und idealerweise ohne weitere manuelle Begutachtung für die Realplanung übernommen werden und ausführbar sind. Daher wird ein hybrides Optimierungsverfahren in zwei Phasen konzipiert: Die erste Phase verwendet eine auf wissensbasierten Regeln basierende heuristische Implementierung, gefolgt von einer metaheuristischen Optimierung in einer zweiten Phase, siehe auch Abb. 5.4. Die zu entwickelnde hybride Optimisierungsmethodik beabsichtigt, die Vorteile einer gezielt deterministischen, heuristischen Optimierung mit den Fähigkeiten eines stochastisch-metaheuristischen Algorithmus zu kombinieren.

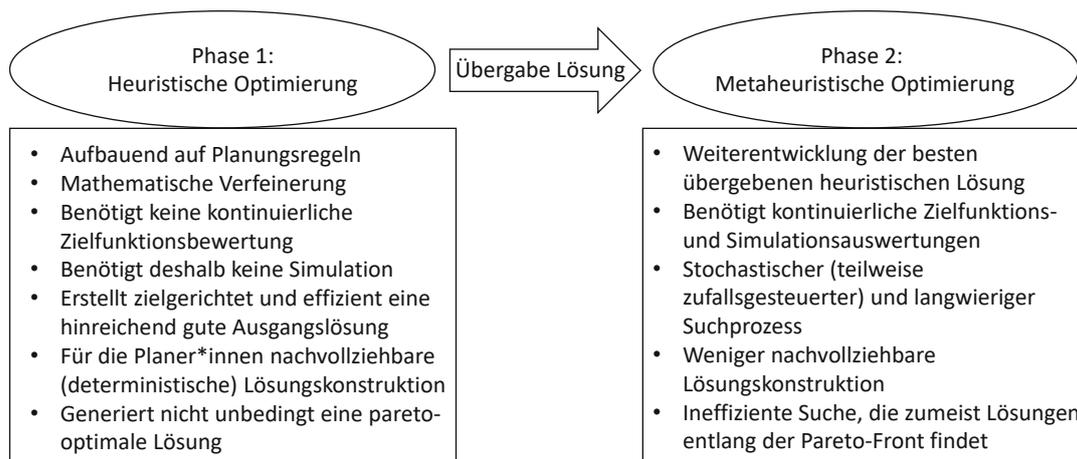


Abb. 5.4: Hybrides Optimierungs-konzept (Quelle: eigene Darstellung)

5.6.1 Phase 1: Heuristische Optimierung

Im Zuge von Interviews und Workshops mit der Planung und IT wurden die von der Planungsabteilung durch langjährige Erfahrung entwickelten Planungsregeln formalisiert. Diese Regeln basieren in den meisten Fällen auf dem Prinzip, mit großen Vertriebsaktionen beim LEH in der Art und Weise umzugehen, sodass diese in kleinere Produktionsaufträge aufgeteilt werden und entsprechend lange voraus geplant werden. In weiteren Gesprächen mit den Prozessexpert*innen wurde zusätzliches Potential für die Glättung durch die Variabilisierung des Reifelagerdauer-Prozessschrittes identifiziert. Die Möglichkeiten für eine maximale Überschreitung der minimalen Reifelagerdauer wurden auf Articlebene formalisiert und für Kurant- und Aktionsmengen separat festgehalten. Auf Grund der teilweise außerordentlich großen Volumina wird zumeist für Aktionsmengen ein größeres Schiebefenster erlaubt als für laufend

⁶das entspricht einigen Minuten in dem konkreten, industriellen Anwendungsfall bzw. höchstens ein paar Stunden in der Generalisierung.

bediente Kurantmengen. Im Rahmen der Entwicklung des hybriden Optimierungsverfahrens empfiehlt es sich zuerst die formalisierten Planungsregeln nachzubilden und mathematisch in einer ersten Phase in einer wissensbasierten Heuristik abzubilden und entsprechend zu verfeinern, bevor in der zweiten Optimierungsphase entsprechende Metaheuristiken entwickelt und untereinander verglichen werden. Der Hauptgrund in der gewählten Vorgehensweise liegt darin begründet, auf bestehendem Wissen aufzubauen und dieses regelbasiert zu erweitern und damit eine auf Basis der Regeln nachvollziehbare gute Lösung zu generieren, welche die Präferenzen der Planungsabteilung widerspiegeln. Auf Basis einer oder mehrerer als bereits für gut empfundenen Lösung(en) wird anschließend im Zuge der stochastischen Optimierung aufgesetzt. Die resultierende wissensbasiert entwickelte gemischt-ganzzahlige *Produktionsglättungsheuristik* (PGH) besteht aus vier modularen Optimierungsphasen (vgl. Kamhuber et al. 2018, S. 103ff.; Kamhuber et al. 2019, S. 2112; Kamhuber et al. 2020, S. 6f.):

1. Aktionsmengenglättung.
2. Kurantmengenglättung.
3. Rack-Optimierung.
4. Optimierung des Bestandsverlaufs.

Jede Phase manipuliert das Produktionsprogramm zur Erreichung verschiedener Teilziele innerhalb der Zielfunktion, wobei ein lexikographischer Ansatz im Zuge der multikriteriellen Optimierung verfolgt wird. Dadurch wird sichergestellt, dass die Optimierung nach einer initialen Datenaufbereitung dem höchstgereihten Teilziel, der Glättung sämtlicher Produktionsartikel, entsprechend Vorrang einräumt. Jede einzelne Phase ist durch ein übergeordnetes Ziel definiert sowie durch den Umstand, dass jeweils mehrere regelbasierte, deterministisch formulierte Optimierungsschritte benötigt werden, um dieses Ziel zu erreichen.

Optimierungsphasen 1 – 2: In den ersten beiden Optimierungsphasen werden die Auslastungsspitzen, resultierend jeweils aus den jeweiligen Aktionsmengen (Phase 1) bzw. Kurantmengen (Phase 2), minimiert. Dies passiert durch eine Vorverschiebung von Spitzenlastmengen in frühere Perioden. Dabei werden außerdem Lücken in der Kapazitätsnachfrage, sogenannte *Auslastungstäler*⁷, geschlossen. Abb. 5.5 verdeutlicht den Zusammenhang zwischen der Vorverschiebung von Belastungsspitzen, resultierend aus Aktionsmengen in Abb. 5.5, in bevorzugt auslastungsschwächere Perioden. Zuerst werden dabei die lokalen Produktionsartikelgradienten (f_1) verbessert, gefolgt von der gezielten Optimierung des globalen Werksgradienten (f_2) gemäß der folgenden Schritte der *Produktionsglättungsheuristik* (PGH):

1. Die Heuristik sortiert alle Artikel gemäß Zuordnung zu einer ABC-Kategorie und iteriert pro Artikel von der jeweils letzten Periode des Planungshorizonts zeitlich nach vorne (Alg. 5.1, Zeile 2) und betrachtet jede Periode in Bezug auf die jeweils anliegenden Aktions- bzw. Kurantmengen. Die Heuristik beginnt mit der Abarbeitung von A-Artikeln, bevor B- und C-Artikel betrachtet werden.
2. Die PGH bildet einen über einen längeren Zeitraum (konkret 26 Wochen) gültigen Lastmittelwert *apl* (engl.: *average production load*) für diesen Artikel. Der jeweilige Lastmittelwert berechnet sich aus der kumulierten Summe an Aktions- und Kurantmengen dividiert durch die Summe an den zu betrachtenden Perioden⁸. Zuletzt wird ein kleiner (prozentualer) Faktor⁹ (z.B. 5%) berücksichtigt, ab dem ein (verschiebender) Optimierungsschritt ausgelöst wird.
3. Produktionsglättung: Unter der Bedingung, dass die anstehende Gesamtproduktionslast in der jeweils aktuellen Periode den Lastmittelwert übersteigt, siehe Periode 30 in Abb 5.5 (links),

⁷Dabei handelt es sich um Perioden bzw. Wochen mit geringerer Auslastung, welche unter dem Durchschnitt im jeweiligen Betrachtungszeitraum liegt.

⁸Den berechneten Lastmittelwert versteht man auch als durchschnittliches Produktionsniveau oder Wochendurchschnitt in diesem Zeitraum.

⁹Diese Herangehensweise empfiehlt sich, um nicht bei bereits geringen Lastspitzen eine Vorverschiebung von Produktionsmengen auszulösen, welche dann aus einer geringen Lastspitze ein erhebliches Auslastungstal generieren und so eine größere Schwankung in das Produktionssystem einbringen bzw. verursachen würde.

werden die Lücken in den Zeiträumen, die der aktuell betrachteten Periode am nächsten liegen, möglichst zeitnah (in Bezug auf die aktuelle Periode) unter Einhaltung der ausformulierten und aktivierten Nebenbedingungen mit Aktionsmengen (Phase 1) oder Kurantmengen (Phase 2) aufgefüllt, bis der jeweils aktuelle Lastmittelwert erreicht ist. Abb. 5.5 sowie Alg. 5.1 (Zeilen 10 - 12) verdeutlichen diesen methodischen Kern innerhalb der PGH.

4. Im Falle von verbleibenden, sogenannten *Überschussmengen* nach Schritt 3 werden diese gleichmäßig auf alle möglichen Ausgleichsperioden¹⁰ verteilt. Dies resultiert in einem höheren Lastmittelwert für all diese betrachteten Perioden.

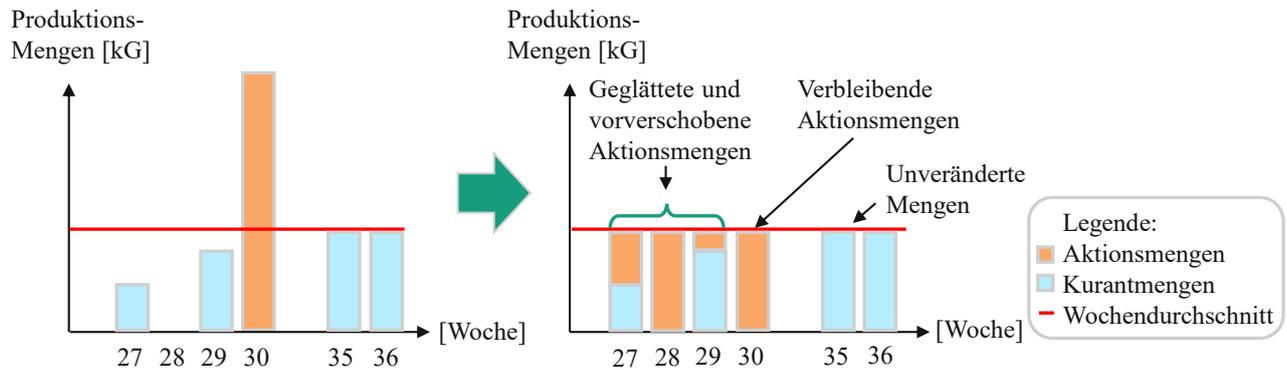


Abb. 5.5: Regelbasierte Produktionsglättung mittels Produktionslückenfüllung durch Versatz von Produktionsbelastungsspitzen (vgl. Kamhuber et al. 2019, S. 2112)

Die Schritte 2 – 4 unterscheiden im Rahmen der Verschiebung von Produktionsmengen (p_{ij}), siehe auch die Zeilen 5 - 13 aus Alg. 5.1, zwischen Aktions- und Kurantmengen. Zuerst werden in Phase 1 die Aktionsmengen aller Artikel entsprechend den einzuhaltenden Nebenbedingungen vorversetzt, anschließend werden im Zuge von Phase 2 die Kurantmengen geglättet. Also bezieht sich der in Alg. 5.1 formulierte Parameter *delta* ggf. sowohl auf Aktionsmengen als auch Kurantmengen. Die Annahme dieser Reihenfolge beruht darauf, dass die Aktionsmengen für den Großteil der Unruhe im Produktionssystem verantwortlich sind und deshalb priorisiert, d.h. zuerst, zu betrachten sind.

Im Zuge von Schritt 3 unterscheidet die Heuristik zwischen lokalen und globalen Lücken: Lokale Produktionslücken stellen Lücken auf Artekebene (in Bezug auf f_1) dar, während globale Produktionslücken in Bezug zum Lastmittelwert der jeweilig produzierten Werksproduktionsmenge (f_2) gebildet werden. Die im Zuge der Berechnung vorzuverschiebenden Lastspitzenmengen werden in beiden Fällen so nah wie möglich in Bezug auf ihre jeweilige Ursprungsperiode versetzt. Dabei werden, in Einklang mit der Hierarchie des Zielfunktionssystems, zuerst lokal günstige Wochen verwendet, bevor auf globale Lücken zurückgegriffen wird (Algorithmus 5.1, Zeilen 10 – 12).

Die PGH berücksichtigt, je nach den gewünschten globalen Einstellungen, alle gegebenen Einschränkungen (5.2) - (5.4). Um die Nebenbedingung (5.3) exakt zu berücksichtigen, müssen vorab einmalig eine Simulation (siehe Abschnitt 5.4) ausgeführt und die Ergebnisse des dazugehörigen Kapazitätsauslastungsverlaufs in Bezug auf die jeweiligen (kapazitiven) Änderungen laufend aktualisiert werden. Ein Versatz ist, bedingt durch die Charakteristik der Methode, nur in zeitlich frühere Perioden möglich, um die Produktions- und Liefertermine zu jedem Zeitpunkt einzuhalten. Jede Mengenverschiebung bedingt, entsprechend des Versatzes, eine Verlängerung der jeweiligen Prozessdurchlaufzeit. Algorithmus 5.1 beschreibt den methodischen Kern (Phase 1), also die Aktionsmengenglättung, der PGH prägnant in Form von Pseudocode. Es gilt zu beachten, dass die PGH im Zuge der Phasen 1 und 2, jeweils ausgehend von der letzten bis zur ersten Periode, zeitlich nach vorne iteriert. Während in Phase 1 Aktionsmengen zeitlich vorversetzt werden, widmet sich Phase 2 der Vorverschiebung von Kurantmengen.

¹⁰Diese Kategorie umfasst alle Perioden, in welche gemäß Nebenbedingung (4.4) eine Produktionsmenge theoretisch versetzt werden kann mit Ausnahme der Mengen in den Perioden, welche Nebenbedingung (4.2) verletzen.

Algorithmus 5.1 Multikriterielle Produktionsglättungsheuristik, Phase 1 (vgl. Kamhuber et al. 2018, S. 6)

Input: Initiallösung x bestehend aus der Produktionsplan-Matrix (ppm), Bestandsverlaufsmatrix (stm) und Kapazitätsverlaufsauslastungsmatrix (cum) sowie dem Lastmittelwert (apl)

Output: Geglättete Lösung $x' = (ppm', stm', cum')$

```

1: for  $j := 1$  to  $m$  do
2:   for  $i := n$  to 1 do // Diese Heuristik iteriert von der letzten bis zur ersten Periode
3:     Update  $apl \leftarrow apl'$ 
4:     Initialize HashMaps: bestLocalWeeks ( $bLW$ ), bestGlobalWeeks ( $bGW$ )
5:     if  $p_{ij} > apl'$  then
6:       Calculate  $delta = p_{ij} - apl'$ 
7:       Determine and save locally and globally (favorable) weeks to shift peak load quantities
8:       Update  $bLW, bGW$ 
9:     end if
10:    while ((( $bLW \parallel bGW$ ) is not empty) and ( $delta > 0$ )) do
11:      Update  $p_{ij}$  and  $q_i$  according to ( $bLW \parallel bGW$ ),  $q_{max_i}$  and constraints (5.2) and (5.4)
12:      Update  $c_{ik}$  according to constraint (5.3), Update  $s_{ij}$ , Update  $delta$ 
13:    end while
14:    Update  $ppm \leftarrow ppm', stm \leftarrow stm', cum \leftarrow cum'$ 
15:  end for
16: end for
17: return  $x' = (ppm', stm', cum')$ 

```

Optimierungsphasen 3 – 4: In den nächsten beiden Phasen werden die Produktionsvolumina entsprechend ihrer ABC-Klassifikation taktisch auf jeweils halbe oder volle *Racks* gerundet. Aufgrund der bereits in den Produktionsaggregaten befindlichen Volumina werden die Lagerbestandsniveaus innerhalb eines definierten Entscheidungsfensters optimiert, um den entsprechenden Ziellagerbestand innerhalb des Einfluss-Zeitfensters zu erreichen, siehe Abb. 5.6. Im konkreten, industriellen Anwendungsfall ist diese Zeitspanne als die artikelspezifisch jeweils dritte Durchlaufzeit in der Zukunft definiert. Dies wird durch Berechnung eines durchschnittlichen, erwarteten Lagerbestands über eine bestimmte Anzahl von Perioden (einstellbar, z.B. eine ganze Durchlaufzeit, siehe Abb. 5.6) und durch das Hinzufügen bzw. Entfernen der Differenz zwischen dem berechneten und dem Ziellagerbestandsniveau zum bzw. vom Produktionsplan gewährleistet. Absil bezeichnet dieses Vorgehen als *Horizontzerlegung* (vgl. Absil und Kedad-Sidhoum 2007, S. 176f.), siehe auch die Abschnitte 3.6 bzw. 3.9.

Der Ansatz der Aufteilung des Planungshorizonts ermöglicht eine effiziente Vorwärtsiteration durch den gesamten Planungshorizont, wobei der dynamische Zielbestand entsprechend optimiert wird. Diese Optimierung des Lagerbestandsverlaufs folgt den Regeln einer **heuristisch basierten, phasenweisen Priorisierung**:

1. Zuerst wird der Lagerbestand von Artikeln, die einen Überschuss an fertigem Reifelagerbestand haben, reduziert bzw. abgepuffert, um Kapazitäten freizugeben (Phase 1). Dies passiert durch eine Reduktion der jeweiligen Produktionsmengen der Artikel in Durchlaufzeit 2. Die erste Durchlaufzeit wird bereits in den jeweiligen Räumen fertiggestellt und kann nicht mehr abgebrochen bzw. reduziert werden. Dadurch werden Kapazitäten während des Entscheidungsfensters (bestimmte Perioden von Durchlaufzeit 2) prinzipiell reduziert bzw. freigegeben. Die Änderungen werden jeweils ab der entsprechenden Periode im Einfluss-Zeitfenster (Durchlaufzeit 3) wirksam.
2. Danach werden die freigesetzten Kapazitäten für andere Artikel verwendet, deren Bestand unter dem Mindestbestand liegt. Zuerst werden diese auf den jeweiligen Mindestbestand aufgepuffert (Phase 2). Im Zuge dessen werden entsprechend die jeweiligen Produktionsvolumina dieser Artikel im Entscheidungsfenster (Durchlaufzeit 2) angehoben, sofern dies (in einem bestimmten Umfang) kapazitativ auf Basis der Ergebnisse von Phase 1 möglich ist.

3. Sofern noch ungenutzte Kapazitäten aus den vorherigen Phasen existieren, werden diese in Phase 3 zuerst für die auf Mindestbestand aufgepufferten Artikel sowie in weiterer Folge ggf. für die übrigen Artikel (analog zu Phase 2) verwendet, um deren individuellen, optimalen Ziellagerbestand ($S_{opt_{ij}}$) zu erreichen bzw. darauf hinzuarbeiten.

Dieses phasenweise Regelwerk stellt sicher, dass die anfänglich freigegebenen Kapazitäten bis zum Ende möglichst vollständig und v.a. zweckmäßig genutzt werden, wobei durch die Heuristik denjenigen Artikeln Vorrang eingeräumt wird, die eine (zeitliche) Verbesserung ihres aktuellen Bestandsniveaus in Relation zum Sollbestandsniveau am dringendsten notwendig haben. Abbildung 5.6 verdeutlicht den zeitlichen Zusammenhang zwischen den einzelnen Zeitabschnitten bezüglich der Lagerbestandsentwicklung: In der Vergangenheit beauftragte Produktionsmengen befinden sich bereits in Produktion und können nicht mehr (weder zeitlich noch mengenmäßig) verändert werden. Änderungen in der Produktionsplanung im Entscheidungsfenster betreffen daher aus Sicht der Lagerbestandsentwicklung Perioden ab Durchlaufzeit 3¹¹.

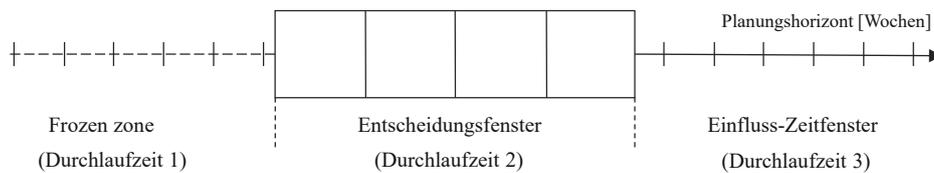


Abb. 5.6: Lagerbestandsentwicklung: Schematische Aufteilung des Planungshorizonts (vgl. Kamhuber et al. 2020, S. 7; Absi und Kedad-Sidhoum 2007, S. 177)

Fazit: Das Hauptergebnis der Heuristik ist ein *effizient* optimierter Produktionsplan. Die Optimierung unterscheidet die Produktionsmengen der Artikel hinsichtlich ihrer Ausprägung als entweder Kurant- oder Aktionsmengen. Die Optimierung beginnt lexikographisch mit der Glättung der einzelnen Produktionsartikel. Vorab werden alle Artikel gemäß ihrer ABC-Klassifikation absteigend sortiert, sodass die Abarbeitung die A-Artikel vor die B- und C-Artikel reiht. Dabei werden alle Artikel innerhalb des gesamten Planungshorizonts regelbasiert auf Basis von heuristischen Informationen *geglättet*. In diesem Rahmen werden die Konzepte der Variabilisierbarkeit der jeweiligen, artikelspezifischen Durchlaufzeit sowie der dynamischen Losteilung (auf Basis heuristischer Informationen) angewendet. Die Heuristik schließt mit der Bildung von Racks und taktischen Rundungen zur Verbesserung der artikelspezifischen Ziellagerbestandsniveaus ab. Die beste von der Heuristik ermittelte Lösung stellt, nach einer Zielfunktionsbewertung gemäß Glg. (5.1), die Ausgangslösung für die darauf aufsetzende Metaheuristik dar. Diese bewertet eine jede neue (potentielle) Lösung in Bezug auf die jeweiligen Änderungen auf Teilzielfunktionsebene in skalierten Form bezüglich dieser Ausgangslösung.

5.6.2 Phase 2: Metaheuristische Optimierung

Auf die erste heuristische Optimierungsphase folgt als zweite Optimierungsphase eine auf das Problem zugeschnittene Metaheuristik, welche die vorherige, wissensbasierte Heuristik ergänzt. Bei dem in der vorliegenden Arbeit vorgestellten, methodischen Konzept werden ein problemspezifisch angepasster Simulated Annealing-Algorithmus als geeignetste Einzellösungsmetaheuristik mit der bekanntesten, populationsbasierten Metaheuristik, einem Genetischen Algorithmus, verglichen und gegenübergestellt. Beide Metaheuristiken erhalten dieselben grundsätzlichen Modifikationsmöglichkeiten der Veränderung der Lösung(en). Die metaheuristische Optimierung greift jeweils auf die beste, gefundene Lösung aus der heuristischen Optimierung zurück und bewertet diese initial gemäß des in Kapitel 4 modellierten Zielfunktionssystems. Das übergeordnete Ziel besteht jeweils darin diese Initiallösung weiter zu verbessern. Mit dem stochastischen Suchansatz wird die Initiallösung schnell verbessert, ohne die Hauptmerkmale und Charakteristika der Lösung (zu) stark zu verändern, wobei lokale Optima im Zuge

¹¹Man versteht die Gegenwart derart, sodass diese bereits die letzte Periode von Durchlaufzeit 1 darstellt. Die erste Periode in der Zukunft, z.B. die jeweils nächste Woche, stellt daher die erste Woche von Durchlaufzeit 2 dar.

der Suche vermieden werden. Die Fähigkeit des SA, schnell bessere Lösungen auf der Grundlage einer einzelnen, bereits guten Ausgangslösung (*Plateaulösung*) zu finden, ohne das Risiko, die gewünschten Kerneigenschaften der heuristischen Lösung zu verlieren, wird in diesem Fall mit der Fähigkeit der Genetischen Algorithmen verglichen, potenziell gute Lösungen in großen Suchräumen zu finden. Als komplementärer Ansatz zur heuristischen Optimierungsphase beantwortet diese Arbeit, speziell für den vorliegenden industriellen Anwendungsfall, die Frage, ob ein trajektorienbasiertes Simulated Annealing Verfahren in der zweiten Optimierungsphase besser und zielgerichteter oder weniger gut geeignet ist als ein populationsbasierter, explorativer Genetischer Algorithmus.

5.6.2.1 Vorbereitungen

Beide Metaheuristiken verwenden die in Gleichung (5.1) aufgestellte Zielfunktion zur Bewertung der einzelnen Lösungen. Die angegebenen Gewichtungsfaktoren erlauben es der Planung, zusätzliche Kontrolle über subjektive Lösungspräferenzen vorzunehmen. Die problemspezifische Implementierung der Metaheuristiken durch die Integration eines zielgerichteten Suchmechanismus verbessert die Performance bzw. das Auffinden der Pareto-Front in komplexen Suchräumen i.d.R. erheblich. Die algorithmischen Anpassungen des Suchmechanismus konzentrieren sich in diesem Fall insbesondere auf die Minimierung des Werksgradienten (f_2). Dieser kann von der Heuristik nicht gezielt verbessert werden, ohne dass dabei teilweise z.B. die Artikelgradienten bzw. die Kapazitätsauslastungsgradienten verschlechtert werden. Die Metaheuristik, insbesondere der SA, ist in der Lage, eine Sicherungskopie der jeweils bisherigen besten Lösung zu erstellen¹² und systematisch die bereits zuvor durch die Heuristik versetzten Mengen zu erfassen. Der erneute Versatz von bereits versetzten Produktionsmengen aus der ersten Phase erfordert die Berücksichtigung zweier zusätzlicher Nebenbedingungen (vgl. Kamhuber et al. 2020, S. 7f.):

$$\sum_{t=0}^{t_{max_j}} p_{ijt} = p_{ij} \quad \forall i, j \quad (5.11)$$

$$p_{ijz} = \sum_{t=0}^{t_{max_j}-z} p_{ijt} \quad \forall i, j, z \quad (5.12)$$

Gleichung (5.11) impliziert, dass die Summe aller Produktionsversatzvolumina p_{ijt} , welche also von Artikel j um t Perioden in die Periode i vorversetzt worden sind ($t > 0$) und derjenigen, die nicht vorverschoben wurden ($k = 0$), dem Gesamtproduktionsvolumen p_{ij} des Produktionsartikels j in Periode i entspricht. Dies stellt eine transparente, im Zuge der metaheuristischen Optimierung notwendig gewordene, Erweiterung der Nebenbedingung (5.4) dar.

In Gleichung (5.12) werden die zur Verschiebung als noch möglich verbleibenden und freigegebenen Volumina p_{ijz} pro Periode und Produktionsartikel zuzüglich der jeweils verbleibenden (maximal) zulässigen zeitlichen Verschiebung z definiert. Diese Nebenbedingung identifiziert die im Zuge der Verschiebung um z Perioden noch möglichen Produktionsversatzvolumina p_{ijz} auf Basis der bereits in diese Periode versetzten Produktionsvolumina p_{ijt} . Diese Einschränkungen werden für alle Produktionsartikel und Perioden eingehalten. Die entsprechenden Versatzmengen werden in einer sogenannten *HashMap*¹³ nach erfolgreicher Lösungsänderung jeweils aktualisiert festgehalten. Das Ziel dieser beiden unterstützenden Nebenbedingungen in der metaheuristischen Optimierung besteht darin, mehr Transparenz und Übersicht von Nebenbedingung (5.4) aus Unterabschnitt 5.3.1 zu erhalten. Im Zuge der Metaheuristik werden bereits durch die Heuristik (einmalig) versetzte Mengen ggf. um weitere Perioden versetzt, falls das noch realisierbar ist.

¹²Der GA stellt dieses Verhalten durch den Einsatz von zumindest einem Elite-Individuum in der Population sicher.

¹³Eine HashMap bezeichnet eine Datenstruktur, welche zu einem Schlüsselwert (z.B. Kombination von Artikelnummer und Periode) einen dazugehörigen Wert speichert.

Ein veranschaulichendes Beispiel ($i, j = konst., t_{max_j} = 4$) mit einer Versatzmatrix vm für verschobene Produktionsvolumen (p_{ijt}) und dazugehörig noch versetzbaren Mengen (p_{ijz}) ist in Tabelle 5.4 dargestellt. In diesem Beispiel sind von 12 Einheiten (E abgekürzt) Produktionsmengen (für Artikel j in Periode i) 4 E davon durch die zuvor ausgeführte Heuristik noch nicht versetzt worden ($t = 0$), während bereits 2 E um eine Periode ($t = 1$), 3 E um zwei Perioden ($t = 2$), 2 E um drei Perioden ($t = 3$) und 1 Einheit um vier Perioden ($t = 4$) vorversetzt worden sind. Dies bedeutet, dass gemäß der Einschränkung (5.12) noch 4 E um max. vier Perioden (d.h. $p_{ijz}(z = 4)$), 6 E um max. drei Perioden, 9 E um max. zwei Perioden sowie 11 E um max. eine Periode vorverschoben werden können. Wenn der Wert für z.B. $p_{ijt}(t = 0)$ durch den nächsten Optimierungsschritt nun um 2 E reduziert wird (weil bisher noch nicht versetzte Produktionsmengen vorverschoben werden), reduzieren sich in diesem Fall auch entsprechend die Werte für p_{ijz} . Der konkrete Umfang der Änderungen innerhalb von $p_{ijz}(z = 1, 2, 3, 4)$ hängt davon ab, um wie viele Perioden die noch nicht versetzten Artikelmenen $p_{ijt}(t = 0)$ nun versetzt werden. Im Zuge eines Mengenversatzes von 2 E um drei Perioden ($t = 3$) wird an dieser Stelle festgehalten, dass im Zuge dieser Implementierung zuerst das Mapping für $p_{ijt}(t = 0)$ um 2 E reduziert wird, bevor das Mapping für $p_{ijt}(t = 3)$ verwendet wird. Außerdem müssen sämtliche Mappings von p_{ijz} mit einem z -Wert ≥ 1 , entsprechend reduziert werden. Der Wert für z errechnet sich über die Differenz von t_{max} und dem konkret ausgeführten Versatz t , hier also $z = (t_{max_j} - t) = 1$. In dem konkreten Beispiel ergeben sich abzüglich der 2 E aus den Werten von Tab. 5.4 für $p_{ijz}(z = 1)$ also anstelle von 11 E \rightarrow 9 E, für $p_{ijz}(z = 2)$ anstelle von 9 E \rightarrow 7 E bzw. für $p_{ijz}(z = 3) \rightarrow$ 4 E bzw. für $p_{ijz}(z = 4) \rightarrow$ 2 E.

Tab. 5.4: Mapping (vm) aus Produktionsartikelversatzmengen (p_{ijt}) und dazugehörigen, noch möglichen Produktionsversatzmengen (p_{ijz}) (vgl. Kamhuber et al. 2020, S. 8)

p_{ijt}	$p_{ijt}(t = 0)$	$p_{ijt}(t = 1)$	$p_{ijt}(t = 2)$	$p_{ijt}(t = 3)$	$p_{ijt}(t = 4)$
#E	4 E	2 E	3 E	2 E	1 E
p_{ijz}	$p_{ijz}(z = 0)$	$p_{ijz}(z = 1)$	$p_{ijz}(z = 2)$	$p_{ijz}(z = 3)$	$p_{ijz}(z = 4)$
#E	12 E	11 E	9 E	6 E	4 E

5.6.2.2 Gemeinsamer algorithmischer Ablauf (SA, GA)

Beide Algorithmen (SA, GA) übernehmen die durch die Glättungsheuristik (Phase 1) erhaltene, beste Lösung als Input (vgl. Kamhuber et al. 2020, S. 7-9). Diese besteht aus der Produktionsplanmatrix (ppm), der Pufferbestandsverlaufmatrix (stm) und der Kapazitätsauslastungsvorschau (cum). Dann initialisieren die Algorithmen die Versatzmatrix (vm) über eine HashMap. Algorithmus 5.2 fasst das angepasste stochastische Nachbarschaftssuchverfahren auf der Grundlage des SA, angewendet auf das vorliegende Problem, zusammen. Nach einem Backup der jeweils aktuell besten Lösung (Zeile 4) wählt der Algorithmus drei zufällige Artikel (Zeile 5) und zufällig benachbarte Perioden (Zeile 6) zur Verschiebung aus und bewertet die entsprechenden, dazugehörigen, Werksgradienten. Nun wird die größte, positive Veränderung bzw. Differenz aller Werksgradienten (mit dem größten Potential) innerhalb dieser Tournament ausgewählt und die aktuelle ppm entsprechend modifiziert (Zeilen 7 - 8). Wenn die abgeleitete Lösung alle Randbedingungen (Zeile 9) erfüllt, wird die neue Lösung sofort akzeptiert und auch hinsichtlich der anderen Teilziele bewertet (Zeilen 10 - 11). Zur vollständigen Bewertung von $f(\mathbf{x})$ muss zur Bestimmung von f_3 dabei jedes Mal das integrierte Simulationsmodul ausgeführt werden. Danach wird die bewertete Lösung im Verhältnis zur besten Lösung aus der heuristischen Optimierung skaliert (Zeile 12). Die global erfolgreiche Aktualisierung auf der Grundlage einer Verbesserung des Fitnesswertes (Zeile 14, Zeile 18) beinhaltet eine Aktualisierung der entsprechenden Lösungsmatrizen (ppm, stm, cum, vm). Andernfalls wird die Lösung verworfen und die alte Lösung wiederhergestellt (Zeile 20).

Algorithmus 5.2 Phase 2: Simulated Annealing Optimierung (vgl. Kamhuber et al. 2020, S. 9)

Input: Ausgangslösung $\mathbf{x} = (ppm, stm, cum, vm)$ // Beste Lösung aus Alg. 5.1 erweitert um eine Versatzmatrix vm

Output: Optimierte Lösung $\mathbf{x}' = (ppm', stm', cum', vm')$ lt. vorliegendem Zielfunktionssystem (5.1)

```

1: Initialize  $t \leftarrow 0$ ;  $T \leftarrow T_{min}$ ;  $y \leftarrow$  Evaluate  $f(\mathbf{x})$  // Zielfunktionsbewertung gemäß (5.1)
2: while  $t > T_{min}$ 
3:   for  $i = 1 \rightarrow max_I$  : do //  $max_I$ : Wiederholungen pro Temperatur
4:      $\mathbf{x}_{prev} \leftarrow \mathbf{x}$  // Backup der vorherigen Lösung erstellen u. speichern
5:      $(j_1, j_2, j_3) \leftarrow random(1, m)$  // Selektiere zufällig drei Artikel
6:      $(i_1, i_2, i_3) \leftarrow random(1, n - 1)$  // Selektiere zufällig drei benachbarte Periodenpaare
7:     // Wähle max. pos. Gradienten mit größtem Verbesserungspotential (Mutation) bzgl.  $f_2(\mathbf{x})$ 
8:      $ppm_{ij} \leftarrow max(ppm_{i_1 j_1} - ppm_{(i_1-1)j_1}, ppm_{i_2 j_2} - ppm_{(i_2-1)j_2}, ppm_{i_3 j_3} - ppm_{(i_3-1)j_3})$ 
9:     if (Check((5.2) – (5.4) and (5.11) – (5.12)) and  $\mathbf{x}' \neq \mathbf{x}$ ) then // Prüfung der Nebenbedingungen
10:      Update  $\mathbf{x}' \leftarrow \mathbf{x}$ 
11:       $y' \leftarrow$  Evaluate  $f(\mathbf{x}')$  // Zielfunktionsbewertung gemäß (5.1)
12:      Scale  $y'$  // Skalierung der aktuell bewerteten Lösung
13:      if ( $y' < y$ ) then
14:         $y' \leftarrow y$  // Akzeptiere die aktuell besser bewertete Lösung
15:      else
16:         $z \leftarrow random(0, 1)$ 
17:        if ( $z < e^{\frac{y-y'}{T}}$ ) then // Akzeptanzkriterium in Anlehnung an (3.10) bzw. (3.11)
18:           $y' \leftarrow y$  // Akzeptiere die aktuell bewertete Lösung gemäß Akzeptanzkriterium
19:        else
20:           $\mathbf{x} \leftarrow \mathbf{x}_{prev}$  // Verwerfe die aktuelle Lösung, Wiederherstellung der vorherigen Lösung
21:        end if
22:      end if
23:    end if
24:  end for
25:   $T \leftarrow \alpha \cdot T$ 
26: end while
27: return  $\mathbf{x}' = (ppm', stm', cum', vm')$ 

```

Die Optimierung wird, gleichermaßen bei GA und SA, von einem stochastischen, zielgerichteten und (greedy) turnierbasiertem (Mutations-) Suchmechanismus unterstützt. Dieser garantiert, dass sich zumindest Teilziel f_2 (nahezu) nicht bzw. kaum verschlechtert und sich $f(\mathbf{x})$ auf Grund der Akzeptanzregel beim SA nicht oder zumindest (je nach Abkühlungsplan) kaum verschlechtert. Dabei werden drei zufällige Artikel und jeweils adjazente Periodenpaare ausgewählt und hinsichtlich einer möglichen Veränderung vorab bezüglich f_2 bewertet (Alg. 5.2, Zeilen 5 – 8). Bei einer Verbesserung bzw. einer vergleichsweise geringen Verschlechterung der Lösung bezüglich $f(\mathbf{x})$ wird diese mit hoher Wahrscheinlichkeit angenommen (siehe Alg. 5.2, Zeilen 14 u. 18), andernfalls verworfen. Beim populationsbasierten GA wird das Akzeptanzkriterium (Alg. 5.2, Zeile 17) weggelassen, wodurch ebenfalls (auch größere) Verschlechterungen von $f(\mathbf{x})$ akzeptiert werden. Der Algorithmus beginnt mit der Verschiebung von Volumina, die zuvor nicht versetzt worden sind und geht dann iterativ zur Verschiebung von Volumina über, die bereits verschoben wurden, aber immer noch ein gewisses Verschiebungspotenzial bieten. Sobald keine $p_{ijt}(t = 0)$ -Volumina mehr vorhanden sind, setzt der Algorithmus mit $p_{ijt}(t = 1)$ -Volumina und so weiter fort, bis $t \leq t_{max_j}$ für jedes der p_{ij} -Volumina erreicht ist. Durch die gezielte Optimierung von f_1 im Zuge der heuristischen Optimierungsphase fokussiert sich die metaheuristische Optimierung auf f_2 , weil eine gezielte Optimierung von f_3 auf Grund des dazu notwendigen Simulationsfeedbacks ohne zusätzliches Wissen schwierig umsetzbar ist. Da eine weitere Behandlung bzw. (Auf-) Teilung der Aktionsmengen durch die metaheuristische Optimierung in diesem konkreten Anwendungsfall nicht vorgesehen ist, konzentrieren sich GA und SA auf die Optimierung der Kurantmengen.

5.6.2.3 Metaheuristische Optimierung: Widersprüchliche Teilzielsetzungen

Während des metaheuristischen Optimierungsprozesses gibt es einen nennenswerten Abtausch zwischen der Glättung auf Werksebene (f_2) und auf Articlebene (f_1), siehe Abbildung 5.7, als Beispiel für einen negativen Kompromiss (vgl. Kamhuber et al. 2020, S. 8): Die Metaheuristiken zielen, wie bereits im vorherigen Abschnitt diskutiert, explizit auf die Optimierung des Werksgradienten ab. Fallweise wird dieser nur auf Kosten eines verschlechterten Artikelgradienten verbessert, da letztere bereits in der heuristischen Optimierungsphase verbessert worden sind. Abbildung 5.7 veranschaulicht eine Situation, die (im Falle der gleichen Gewichtung von f_1 und f_2) zu einer global schlechteren Lösung führt. Die Häufigkeit solcher Kompromisse nimmt mit fortschreitender metaheuristischer Optimierung zu, weil f_2 zusehends verbessert wird und das Potential zur weiteren Verbesserung sukzessive reduziert wird, während das Verschlechterungspotential bei f_1 hingegen in Relation zunimmt. Der Suchmechanismus sowie das SA-Akzeptanzkriterium bzw. in Analogie der GA-Auswahloperator stellen deshalb sicher, dass neben verbesserten Lösungen teils auch (wenig) schlechtere Lösungen für die Aktualisierung der SA-Lösung oder die GA-Reproduktion akzeptiert werden, damit die möglichst explorative Suche aufrechterhalten bleibt.

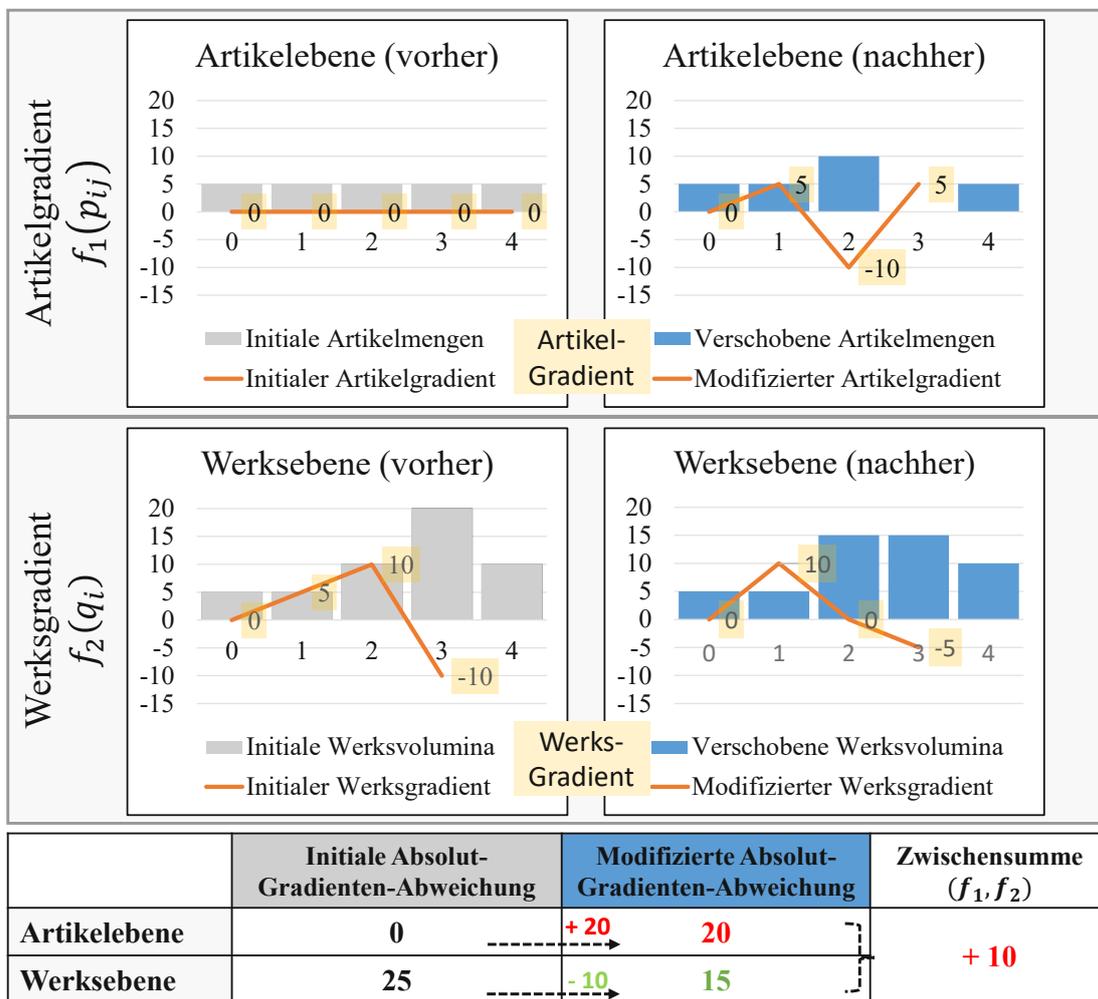


Abb. 5.7: Beispiel für einen schlechten Kompromiss (*trade-off*) (vgl. Kamhuber et al. 2020, S. 8)

Am Ende der metaheuristischen Optimierung werden die am besten bewertete Lösung und die zwei besten (schwach-) Pareto-optimalen Lösungen bezüglich f_1 und f_2 (sowie beim SA zusätzlich die zuletzt gefundene Lösung) erneut innerhalb eines vordefinierten Gewichtungsbereichs (unter Verwendung von 2 zusätzlichen Gewichtungsvektoren) bewertet. Die Lösung, die in diesem vordefinierten Bereich insgesamt die geringste Fitness aufweist, wird als beste Lösung final ausgewählt. Dieses abschließende Verfahren ist eine effiziente kombinierte Anpassung von Chang, Hsieh und Lins Ansatz der graduellen

Prioritätsgewichtung (siehe auch 3.4.5.1) sowie Phase 3 von Karimi-Nasab und Konstantaras Zufalls-Suchverfahren, welches in 3.4.5.3 diskutiert wird (vgl. Chang et al. 2002, S. 171-183; Karimi-Nasab und Konstantaras 2012, S. 479-490). Das Verfahren von Karimi-Nasab wählt die beste Lösung aus einer Liste von Pareto-optimalen Alternativlösungen über die TOPSIS Methode aus. Diese beiden Verfahren werden derart kombiniert, dass nur einige wenige bestimmte (3 – 4 zumindest schwach) Pareto-optimale Lösungen erneut bewertet und gegenübergestellt werden. Diese Herangehensweise stellt somit einen geringen Zeitaufwand dar. Durch den vordefinierten Gewichtungsbereich wird jede dieser Lösungen auch auf ihre Robustheit bewertet, wobei die Robustheit über eine Veränderung der Gewichtungen ermittelt wird. Am Ende wird die robusteste Lösung als Gesamtlösung des Verfahrens herangezogen und ausgegeben.

5.6.2.4 Unterschiede Implementierung SA - GA

Der spezifisch angepasst implementierte GA, der auf dem Standard-GA basiert (siehe Alg. 3.3), imitiert den diskutierten zielgerichteten Suchmechanismus, indem er diesen als Mutationsoperator für eine gezielte Suche verwendet, und wendet die fitnessproportionale Selektion mittels Roulette-Rad Auswahloperator (siehe Abb. 3.6) auf die Individuen der Population an. Dieser Selektionsoperator, der die Individuen für die nächste Population im Verhältnis zu ihrer Fitness auswählt, eignet sich aufgrund der Diversität und des Explorationscharakters des Algorithmus besser für kleine Populationen als z.B. der auf Ranglisten oder Turnieren basierende Selektionsoperator. Ausgenommen von dieser Selektionsmethode ist die jeweils aktuelle Anzahl an Elite-Individuen (je nach Populationsgröße 1 – 2 EI), welche immer in die nächste Generation mitgenommen werden.

Der Standard-GA (SGA) arbeitet prinzipiell mit den drei Kernoperatoren Selektion, Mutation und Rekombination (vgl. Goldberg 1989, S.10-14) und unterscheidet sich durch die populationsbasierte Selektion und Suche bereits wesentlich von einem SA, welcher sich auf die Veränderung einer bestehenden Lösung bzw. Trajektorie verlässt und diese laufend bewertet und akzeptiert oder verwirft. Während der Mutationsoperator bei SA und GA gleichwertig implementiert wird, bietet der SGA noch die Verwendung eines als (problemspezifisch) geeignet empfundenen Rekombinations-Operators an.

Da bereits kleine Änderungen in den Mappings, siehe Tab. 5.3, ggf. große Änderungen mit sich bringen, wird der bereits besprochene Mutations-Operator von einem Rekombinations-Operator, der wenig Genmaterial im Zuge der Kreuzung zerstört, unterstützt. Der in dieser Arbeit entwickelte Rekombinations-Operator vertauscht jeweils eine Gen-Position von zwei zufällig ausgewählten Individuen unter Berücksichtigung einer einstellbaren Verhältnismäßigkeit, wobei konkret ein selbstdefiniertes Verhältnis¹⁴ zwischen 0.67 und 1.50 für eine erfolgreiche Rekombination vorausgesetzt wird. Dieser stochastische, Swap-Exchange basierte Operator ist ggf. eine gute explorative Ergänzung zu dem turnierbasierten Mutationsmechanismus. Der Grund für kleinteilige Änderungen in sowohl Rekombination als auch Mutation liegen darin begründet, dass bereits sehr kleine Änderungen in den Mappings, siehe Tab. 5.3, ggf. große (ZF) Änderungen mit sich bringen. Wenn im Zuge des algorithmischen Ablaufs die eine oder andere Änderung im Zuge eines (zu) großen Mutations- oder Rekombinationsschritts übersprungen wird, ist es nur schwer möglich, diese Änderung wieder (schrittweise) rückgängig zu machen, weil der GA über kein Gedächtnis verfügt.

Abb. 5.8 stellt die Ergebnisse der Anwendung des Mutationsoperators mit der Wahrscheinlichkeit $p_{mut} = 1$ mit der Anwendung des Rekombinationsoperators mit der Wahrscheinlichkeit $p_{rek} = 1$ jeweils auf die gesamte Population bezogen in je 2 Läufen (Datensatz 4, siehe Kapitel 6) gegenüber. Dabei stellt sich das frühe und stark ausgeprägte Konvergenzverhalten des effizienten Mutationsoperators heraus. Die Abbildung verdeutlicht, dass ein GA (12 Individuen, 1600 Generationen) mit dem Mutationsoperator bereits nach ca. 1.000 ZFAen (Zielfunktionsauswertungen) deutlich bessere Ergebnisse hervorbringt als ein GA mit dem besprochenen Rekombinationsoperator nach ca. 18.000 ZFAen.

¹⁴Dieses Verhältnis hängt vom Anwendungsfall ab und ist nicht unmittelbar in dieser Ausprägung auf eine andere Implementation übertragbar.

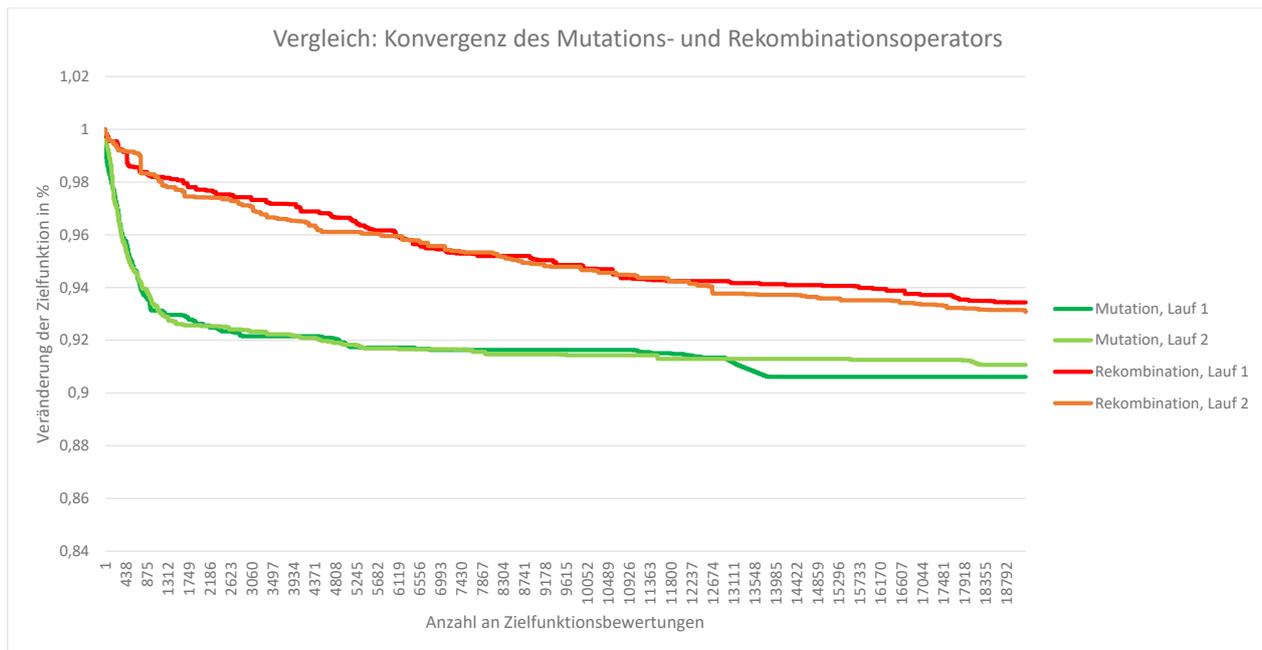


Abb. 5.8: Gegenüberstellung von Mutations- und Rekombinationsoperator (Quelle: eigene Darstellung)

Basierend auf den vergleichenden Optimierungsergebnissen beider Operatoren aus Abb. 5.8 wird der entwickelte Rekombinations-Operator für die Gegenüberstellung der Leistungsfähigkeit von SA und GA auf Grund der langsamen Konvergenz vorläufig deaktiviert. Im Gegensatz zum SGA werden die selektierten Individuen iterativ daher ausschließlich durch (Punkt-) Mutationen, siehe Zeile 8 aus Alg. 5.2, verändert und neu bewertet, wodurch eine gewisse Ähnlichkeit zu einem EA bzw. der Implementation des SA in Alg. 5.2 vorliegt. Die entsprechenden Unterschiede zwischen einem GA und einem EA werden in 3.3.2 diskutiert. Der Rekombinationsoperator wird im Anschluss an die Feststellung des für das vorliegende Problem besser geeigneten Algorithmus ergänzend im Detail für den GA diskutiert und ggf. (mit einem bestimmten Anteil) wieder aktiviert.

5.6.3 Objektorientierte Modellierung

Die Methodik wird in der Programmiersprache Java *objektorientiert* umgesetzt. Dies bedeutet, dass jede Veränderung einer (Zwischen-) Lösung in Form von Objekten modelliert wird. Die Lösung für den Produktionsplan besteht aus einem Objekt der Klasse *Zeilen*, welches selbst wiederum aus vielen Objekten der Klasse *Zeile* besteht. Jede Modifikation einer bestehenden Lösung, z.B. der Ausgangslösung, bedeutet das Hinzufügen einer Zeile zum bestehenden Zeilen-Objekt. Dies ist für sämtliche Modifikationsschritte und Phasen der hybriden Optimierung gültig und umfasst daher sowohl Mengenreduktionen als auch Mengenerhöhungen. In jeder Zeile ist die vollständige Information aufgelöst, wann eine Menge von welchem Artikel mit welchem Versatz in eine bestimmte Periode versetzt worden ist. Das Zeilen-Objekt ist zentraler Kern der Modellierung und wird v.a. an die Zielfunktion bzw. das Simulationsmodell übergeben, um entsprechende Auswertungen zu erhalten. Das Zeilen-Objekt verfügt über eine anfängliche Größe von ca. 1.000 – 2.000 Zeile Objekten, während die finale Lösung je nach Laufzeit ca. 5.000 – 20.000 Zeilen umfassen kann. Darüber hinaus werden diverse, weitere objektorientierte Klassen verwendet, wobei der Fokus auf Grund der geforderten Laufzeit (effizienz) auf der Nutzung möglichst einfacher Datenstrukturen im Mittelpunkt steht: *Arrays*, *ArrayLists* und *HashMaps* (siehe Anhang A.16). Weitere objektorientierte Klassen wurden einerseits für den GA oder SA implementiert, z.B. *Chromosome*, oder für die Simulation zur Raumauslastungsvorschau. Ein Objekt der Klasse *Chromosome* beinhaltet neben dem Zeilen-Objekt weitere Informationen, z.B. betreffend den Zielfunktionswert bzw. eine Mappingübersicht über alle versetzten Volumina.

5.7 Statistische Datenanalyse

Das Optimierungsverfahren wird durch eine rollierend ausgeführte, statistische Datenanalyse ergänzt (vgl. Kamhuber et al. 2019, S. 2114f.). Das Ziel der zur Analyse gehörigen Auswertung besteht darin, Empfehlungen für eine kontinuierliche Anpassung der Stammdaten, insbesondere der Mindest- und Zielreichweiten, sowie aktualisierte ABC-Klassifikationen auf der Grundlage historischer Daten zu liefern. Als Input werden die rollierend gespeicherten Ist-Bestandsverläufe aus jeder einzelnen Periode, die Produktionsvorschauemengen auf Articlebene und die dazugehörigen Produktionsrückmeldungen verwendet. Der zur Auswertung herangezogene Datensatz umfasst insgesamt 23 Planungswochen von KW 30/2018 - KW 52/2018. Der Verteilungsplot bzw. *Boxplot*¹⁵ in Abbildung 5.9 zeigt die Streuung der Lagerbestandsverläufe für jeden Artikel sowie die Mindest-, Durchschnitts- und Ziellagerbestände. Dieses Diagramm bildet die Grundlage für die Bewertung der Lagerbestandsvolatilität. Idealerweise liegt der Ziellagerbestand, durch den grünen Punkt in der Grafik gekennzeichnet, genau auf dem Median. Die Abbildung veranschaulicht, dass z.B. das Bestandsniveau von Artikel 1 leicht über dem Median liegt und sehr stabil ist, während Artikel 7 einen Überschussbestand anzeigt. Die daraus abgeleitete Empfehlung für letzteren ist die Reduktion des entsprechenden Mindest- und Ziellagerbestandsniveaus. Die Prämisse für die Evaluierung besteht darin, dass die Artikel auch nahe am Median liegen, um als stabil bewertet zu werden. Abweichungen vom Median lassen rückschließen, dass dies in der Vergangenheit nicht immer realisierbar war. Das bedeutet, dass bestimmte Artikel schwieriger planbar sind und dementsprechend einen sogenannten Risikozuschlag erhalten. Ebenfalls deuten stabile Bestandsverläufe wie bei Artikel 1 darauf hin, dass für diese Artikel, basierend auf deren Stabilität, bewertet über die Artikelhistorie, eine entsprechende Bestandsreduktion realisierbar ist.

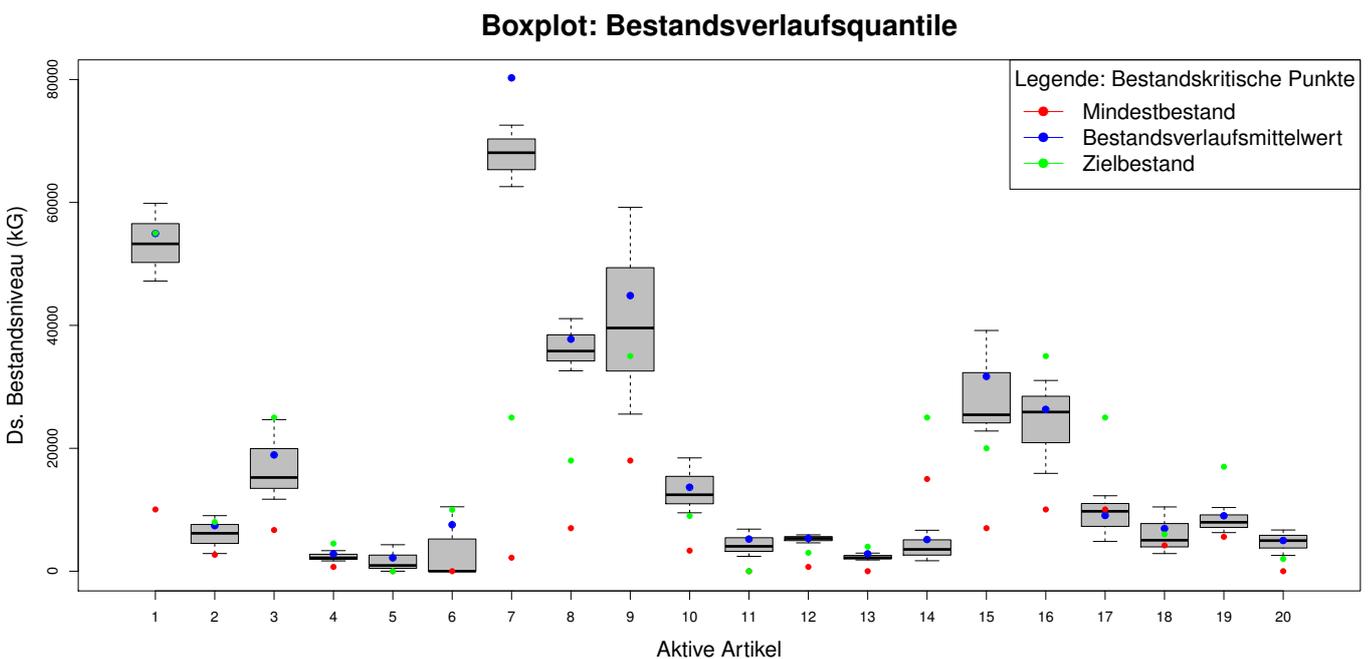


Abb. 5.9: Boxplot zur Bewertung der Bestandsvolatilität in Bezug auf die Produktionsartikel (vgl. Kamhuber et al. 2019, S. 2115)

Als weiteren Input für die Datenanalyse zeigt Abbildung 5.10 einen Vergleich auf Articlebene zwischen der statischen und dynamischen relativen Abweichung zwischen dem vorgeschlagenen und dem aktuellen Produktionsvolumen. Die statischen relativen Produktionsabweichungen entstehen durch die Gegenüberstellung von real beauftragten Produktionsmengen des oben genannten Zeitraums versus einem einzigen erstellten Produktionsvorschlag aus der Vergangenheit für die in diesem Zeitraum liegende Zukunft. Die dynamischen Produktionsabweichungen entstehen durch die Gegenüberstellung

¹⁵siehe zur Erklärung auch den Anhang, Abschnitt A.12

von real beauftragten Produktionsmengen gegenüber den rollierend erstellten Produktionsvorschlägen, wobei der Vorschlag für die jeweils nächste Woche in die Analyse aufgenommen worden ist.

Diese Grafik belegt den Vorteil einer rollierenden Produktionsplanung hinsichtlich geringerer Produktionsabweichungen im Vergleich zur statisch durchgeführten Produktionsplanung. Die zuletzt genannte Variante erstellt einmalig in der Vergangenheit ohne regelmäßige planungsbezogene Informationsaktualisierungen einen Produktionsplan. Die Vorschläge unterscheiden sich in der konkreten Abbildung von den Rückmeldedaten zur Ist-Produktion aufgrund des Hochfahrens des neu gebauten Werks, konkret definierten Produktionsrhythmen sowie diverser, externer Gründe (Anlieferungs- und Beschaffungsstrategien, Preis- und Volumenseffekte, Fehler in der Datenaufbereitung, etc.).

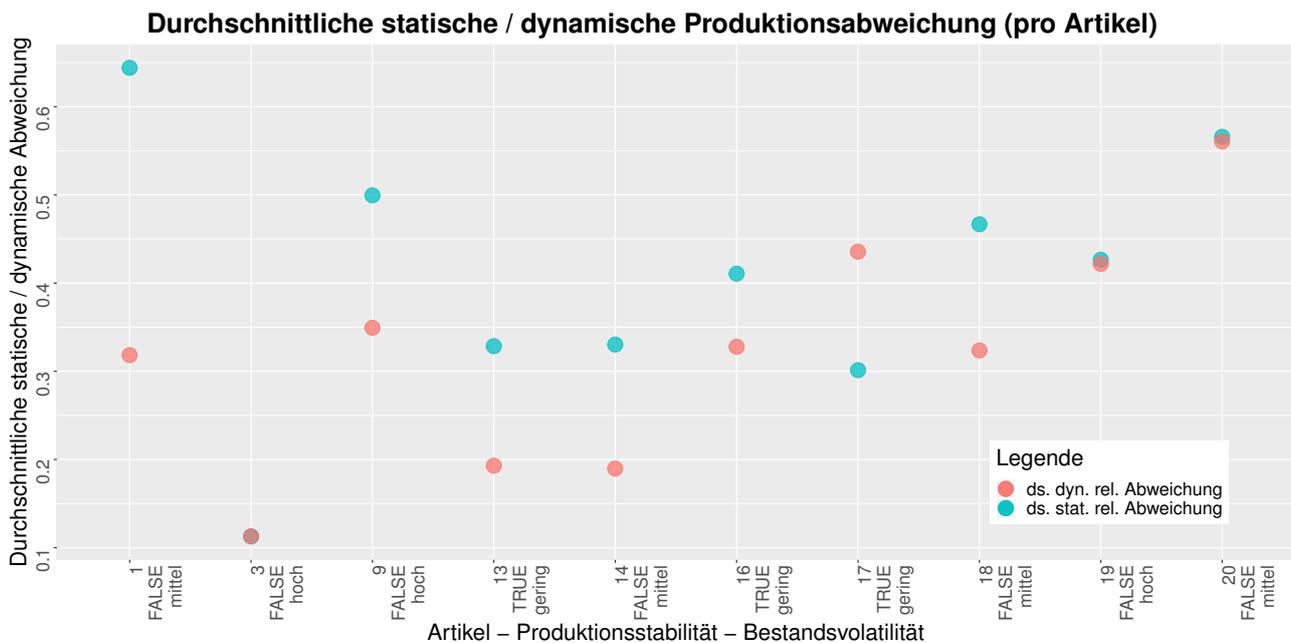


Abb. 5.10: Durchschnittliche statische und dynamische Produktionsabweichungen zwischen Produktionsvorschlag und Produktionsrückmeldung (vgl. Kamhuber et al. 2019, S. 2115)

5.8 Fazit

Dieses Kapitel beschreibt, in Übereinstimmung mit der *Research Rigor* Richtlinie gemäß Design Science bzw. ORMA, den detaillierten Aufbau und die Modellbildung für die zu entwickelnde Produktionsglättungsmethodik sowie die Umsetzung der dazugehörigen hybriden Optimierungsverfahren.

Aus der Anforderungsübersicht sowie den Zielen dieser Arbeit wird das Zielfunktionssystem vollumfänglich formuliert. Die Transparenz, Einstellbarkeit und Begründbarkeit der zu entwickelnden Methodik schaffen die Grundlage für die Erstellung und Verwendung einer gewichteten Zielfunktion mit Fokus auf das Thema Produktionsglättung. Das entsprechende Zielfunktionssystem wurde in (5.1) – (5.12) inklusive der systemimmanenten Nebenbedingungen vollständig formuliert und aufgestellt. Die heuristische Optimierung verfolgt dabei das Hauptziel, den wichtigsten Zielfunktionsbestandteil zuerst zu verbessern, ohne dabei die anderen Zielfunktionsbestandteile deutlich zu verschlechtern. Daher ist eine Relaxation der strengen, lexikographischen Optimierung sinnvoll, weil andernfalls der zur Verfügung stehende Suchraum stark eingeschränkt und eine kontinuierlich exakte Bewertung der anderen Zielfunktionsbestandteile im Zuge der Heuristik notwendig wird. Weil nicht alle Zielfunktionsbestandteile heuristisch gezielt optimierbar sind, werden zwei gleichwertig implementierte Metaheuristiken für das vorliegende Problem untereinander verglichen. Während die integrierte Simulation fundamental notwendig für die Bewertung der Zielfunktion und die Durchführbarkeit unterschiedlichster Planungsszenarien ist, zeigt die Datenanalyse zusätzlich zu hebendes Optimierungspotential als Ergänzung zu der hybriden Optimierung auf.

Die Zielfunktion, siehe Glg. (5.1), beinhaltet einen booleschen Zielfunktionsbestandteil (Teilziel f_5) und ist dadurch bereits nichtlinear. Außerdem wird die Teilzielfunktion f_3 anhand einer dynamischen Simulation zur Kapazitätsauslastungsvorschau ausgewertet. Durch das Mengengerüst sowie die artikel-spezifischen Arbeitspläne und die Aggregation der Artikel zu Artikelgruppen auf Stellplätze ist der Zielfunktionszusammenhang nichtlinear. Die Teilziele f_1 und f_2 ergänzen sich entweder, verhalten sich zueinander neutral, oder sind ggf. im Zuge der globalen Zielfunktionsminimierung auch gegenläufig, siehe Abb. 5.7. Dementsprechend stellt sich die Hypothese aus Abschnitt 2.3, dass ein nichtlineares Optimierungsproblem vorliegt, als korrekt heraus.

In Bezug auf das vorliegende Produktionsplanungsproblem trifft das Prinzip der widersprüchlichen Zielsetzungen zu: Einerseits wird eine gleichmäßige Produktionsausstoßrate für die zu produzierenden Artikel gefordert bzw. durch die Zielfunktion gewährleistet, andererseits liegt z.B. ein je Produktionsartikel individueller Arbeitsplan bzw. individuelles Zielbestandsniveau (i.d.R. mit einem dynamisch besetzten Wert > 0) vor. Die unterschiedlichen, einzelnen Zielsetzungen erfordern die Betrachtung von mehreren Entscheidungsvariablen gleichzeitig, siehe auch Unterabschnitt 5.3.1, wodurch in Kombination mit dem Planungshorizont sowie zumindest einer dynamischen Nebenbedingung und einer begrenzten Anzahl an Zielfunktionsauswertungen ein Blackbox-Problem gemäß 2.3.2 vorliegt.

Kapitel 5 beschreibt die Entwicklung und Umsetzung der hybriden Optimierungsmethodik gemäß DS bzw. ORMA, siehe Abschnitt 1.5. Die lexikographische Optimierung erlaubt eine gezielte, heuristische Optimierungsphase und benötigt dabei weder Zielfunktion noch Simulations-Feedback. Der methodische Kern der heuristischen Optimierung ist in Abb. 5.5 anhand von Phase 1 der PGH dargestellt: Das Konzept der Losteilung wird methodisch derart umgesetzt, sodass es die Zielfunktion bzw. insbesondere die Teilziele f_1 und f_2 implizit gezielt verbessert, ohne dabei die in Abschnitt 5.3 erstellte Zielfunktionsgleichung (5.1) zu verwenden. Die Methode behandelt Kurant- und Aktionsmengen gesondert, d.h. separat voneinander, in jeweils einer eigenen Optimierungsphase und verwendet neben statisch vorgegebenen Kapazitätsgrenzen eine dynamische Kapazitätsvorgabe für die Engpassanlage, welche gemäß Goldratt zuvor in der Problemanalyse identifiziert worden ist (vgl. Lenort et al. 2012, S. 225).

Die zweite auf die Heuristik aufsetzende, metaheuristische Optimierungsphase stellt die beiden vielversprechendsten Metaheuristiken für dieses Problem einander gegenüber und greift dabei auf Zielfunktion und Simulations-Feedback zurück. Die Gewichtungen wurden nach Abstimmung mit der Planung und IT definiert, um deren Präferenzen einfließen zu lassen. Die zielgerichtete Suche fokussiert sich dabei auf f_2 bzw., bedingt durch das Simulationsfeedback, zusätzlich auf f_3 . Diese beiden Teilziele wurden im Zuge der vorangegangenen, heuristischen Optimierung, die insbesondere eine Verbesserung von f_1 und f_4 angestrebt hat, nicht explizit optimiert. Teilziel f_5 ist gemäß Glg. (5.9) eine Abschätzung der zu erwartenden Rüstaufwände pro Periode. Dementsprechend empfiehlt sich eine Mitbetrachtung, jedoch keine explizite Optimierung, im Zuge der Methodik. Der objektorientierte Charakter verleiht der Methodik eine einfache (iterative) Rückverfolgbarkeit im Zuge der vollständigen Lösungsgenerierung. Dies schafft Vertrauen hinsichtlich jeder einzelnen Optimierungsphase innerhalb des iterativen Lösungsfindungsprozesses. Die im Zuge des metaheuristischen Optimierungsprozesses zunehmend widersprüchlichen Zielsetzungen belegen, dass eine rein lexikographisch heuristische Optimierung alleine nicht ausreicht und eine Metaheuristik für den weiteren Suchprozess erfordert. Am Ende dieser Optimierungsphase werden die besten Lösungen in einer vordefinierten (Gewichtungs) Bandbreite erneut bewertet. Dadurch erfolgt eine Verfeinerung des Verfahrens gemäß ORMA im Zuge der Selektion der robustesten Lösung.

Die offline integrierte statistische Datenanalyse und Auswertung aus Abschnitt 5.7 bietet durch die Möglichkeit der Nutzung historischer Datensätze insofern einen strategischen Mehrwert für die Optimierung, als sie die Qualität der Bestandsverlaufs- und Stammdaten rollierend automationsgestützt validiert und über ein entsprechend offline erstelltes Ersatzmodell, siehe 3.4.5.7, Vorschläge zur Anpassung der jeweiligen Parameter (ABC-Werte, aktualisierte Werte für die Bestandsüberwachung) liefert.

6

Kapitel 6

Validierung des entwickelten Optimierungsverfahrens

Die entwickelte Methode wird in Form eines Programms auf Java 4.7.3a (Java 10) implementiert und die Fallstudie auf einem handelsüblichen Notebook-Computer (2,7 Gigahertz Intel Core i7-7500 CPU, 16 GB RAM) ausgeführt. Die Daten und Planungsergebnisse werden über ein verschlüsseltes Oracle-Datenbankprotokoll ausgetauscht. Eine rechnergestützte Studie wird an vier verschiedenen Datensätzen durchgeführt, welche reale Szenarien der Produktion am neuen Werksstandort darstellen:

- Datensatz 1: 78 Planungswochen, 13/2018 - 38/2019 (Hochlaufphase)
- Datensatz 2: 78 Planungswochen, 40/2018 - 13/2020 (stabiles Produktionssystem)
- Datensatz 3: 78 Planungswochen, 10/2020 - 35/2021 (stabiles Produktionssystem, Vergleich GA-SA)
- Datensätze 4 und 4': 78 Planungswochen, 22/2022 - 47/2023 bzw. 23/2022 - 48/2023 (aktuellster Datensatz, Operatoren-Vergleich)

6.1 Voraussetzungen und Rahmenbedingungen für die Validierung

Die Voraussetzungen für die Validierung werden einerseits von entsprechenden Stammdaten auf Datenbanktabellen gewährleistet, welche die Produktions- und Kapazitätsstammdaten des Werksstandorts zur Verfügung stellen. Andererseits wird das veränderliche Mengengerüst auf einer Datenbanktabelle (zeilenweise) zur Verfügung gestellt. Jede Zeile beinhaltet die Angabe des Produktionsartikels, der dazugehörigen Produktionsmenge und Kalenderwoche. Außerdem erfolgt die Angabe, ob es sich um eine bestandswirksame Kurantmenge oder eine nicht bestandswirksame Aktionsmenge handelt. Aktionsmengen beinhalten zusätzlich die Angabe einer Aktionsnummer. Von der Methodik wird anschließend für jede Zeile die Modifikationsstufe ergänzt. Dies stellt eine Information dar, welche Veränderung diese Zeile erfahren hat. Weiters wird für jede Zeile der maximal mögliche Zusatzversatz (in KW) angegeben sowie in Relation dazu der realisierte Zusatzversatz. Ein Datensatz stellt eine Lösung dar und besteht, je nach Artikelumfang, Planungshorizont und Aktionsmengen aus ca. 2.000 Zeilen und wird im Laufe der heuristischen Optimierung entsprechend um ein Vielfaches auf ca. 5.000 - 10.000 Zeilen erweitert, siehe Unterabschnitt 5.6.3. Die metaheuristische Optimierung ergänzt den Datensatz erneut um einen bestimmten Faktor (auf bis zu 20.000 Zeilen). Die Voraussetzung für die Validierung besteht darin, dass sich in den durchgeführten Läufen die Produktionsartikel- und Bestandsstammdaten sowie Kapazitätsangaben pro Datensatz nicht verändert haben.

Es werden in Summe vier zeitlich verschiedene Datensätze für die Validierung ausgewählt, weil die Charakteristik der Methode dadurch sowohl beim Hochfahren des Werks als auch in einem bereits stabilen Produktionssystem bewertet wird. Für den Werksstart existiert auf Grund der Einmaligkeit der Situation nur ein Datensatz (weitere würden sich durch lediglich eine Planungswoche unterscheiden),

während für das stabile Produktionssystem ein Datensatz per Ende 2018 (Datensatz 2), ein Datensatz per Anfang 2020 (Datensatz 3) sowie ein Datensatz per Anfang 2022 (Datensatz 4) existieren. Diese werden so gewählt, weil sie einerseits zeitlich möglichst weit auseinanderliegen und andererseits eine unterschiedliche Saisonalität aufweisen. Die Methode soll für alle vier Datensätze eine in Bezug auf die Charakteristika ähnliche und zudem qualitativ hochwertige geglättete Lösung in einer akzeptablen Laufzeit liefern. Darüber hinaus werden die beiden implementierten Metaheuristiken anhand von Datensatz 3 vergleichend in Bezug auf den realen Einsatz gegenübergestellt. Datensatz 4 dient der Validierung der Ergebnisse aus dem algorithmischen Vergleich in Datensatz 3. Die Datensätze unterscheiden sich abgesehen vom betrachteten Zeitraum v.a. durch die jeweiligen Produktionsausstoßschwankungen der Basislösung und den Anteil der Aktionsmengen (je nach Datensatz zwischen annähernd 0 und max. 25%).

Die Rahmenbedingungen für die Validierung werden für den metaheuristischen Vergleich um die folgenden Bedingungen erweitert:

- Beide Algorithmen verwenden dieselbe Startlösung, die beste von der Heuristik gefundene Lösung.
- Beide Algorithmen verwenden denselben gezielten Suchmechanismus, welcher eine aktuelle Lösung auf eine bestimmte Art und Weise manipuliert.
- Beide Algorithmen werden anhand ihrer jeweils besten ermittelten Parametrisierung verglichen.

Die Optimierungsmethode wird in Abschnitt 6.2 ganzheitlich auf Grund ihrer Güte und Leistungsfähigkeit sowohl hinsichtlich der heuristischen als auch metaheuristischen Optimierung bewertet, sowie hinsichtlich ihrer Laufzeiteffizienz nach Landau theoretisch und praktisch in Abschnitt 6.3 validiert. Zudem werden die in 3.8 diskutierten Grundsätze für einen algorithmischen Vergleich berücksichtigt.

6.2 Auswertung und Interpretation der Ergebnisse

Die Ergebnisse der hybriden Optimierung werden gemäß dem aufgestellten Zielfunktionssystem, siehe auch Abschnitt 5.3, ausgewertet und interpretiert. Während die Heuristik regelbasiert eine bestimmte Anzahl an Optimierungsschritten vornimmt, wird die Anzahl an Zielfunktionsauswertungen bei den Metaheuristiken durch die Parametrisierung gezielt vorgegeben. Ein *Optimierungsschritt* ist eine Änderung einer aktuellen Lösung, ohne dabei eine Zielfunktionsbewertung vorzunehmen. Die Zielfunktionsbewertung wird bei der Heuristik nachträglich durchgeführt, um einerseits eine in Teilzielen bewertete skalierte Ausgangslösung für die Metaheuristiken zu erstellen und andererseits die Qualität der von der Heuristik gefundenen (Zwischen-) Lösung als auch deren einzelnen Optimierungsphasen zu beurteilen. Die Ergebnisse der Heuristik sowie der Metaheuristiken wurden vom Autor dieser Arbeit teilweise veröffentlicht (vgl. Kamhuber et al. 2018, S. 101-116; Kamhuber et al. 2019, S. 2114-2116; Kamhuber et al. 2020, S. 9-12).

6.2.1 Ergebnisse der heuristischen Optimierung

Die in den Abbildungen 6.1 und 6.2 dargestellten, globalen Optimierungsergebnisse werden als Verlaufskurven neben den Trends der einzelnen Teilzielergebnisse gezeigt. Die erste und zweite Phase der heuristischen Optimierung stellen den methodischen Kern der mehrphasigen heuristischen Optimierung dar und beinhalten die Produktionsglättung für Aktions- und Standardvolumen. Diese beiden Phasen sind für 30% - 40% der Verbesserung der gesamten Zielfunktion verantwortlich. Der Gesamtkapazitätsauslastungsverlaufgradient (aggregiert aus fünf Teilgradienten pro Kapazitätsgruppe) wird durch die Optimierung des Artikelgradienten in den Phasen 1 und 2 positiv beeinflusst. Die Optimierung der drei wichtigsten Teilziele $f_1 - f_3$, also der Artikel-, Werks- und Kapazitätsauslastungsverlaufgradienten, in den Phasen 1 und 2 wird auf Kosten eines minimal schlechteren Lagerbestandsverlaufs sowie eines höher bewerteten, aggregierten Rüstkostenaufwands erreicht. Das liegt v.a. an der Logik des Verfahrens begründet. Die Methodik sieht vor, dass in bestimmten Fällen Mengen von Perioden in andere Perioden

versetzt werden, in denen zuvor keine Produktionsmengen dieses Artikels vorgesehen bzw. berücksichtigt waren.

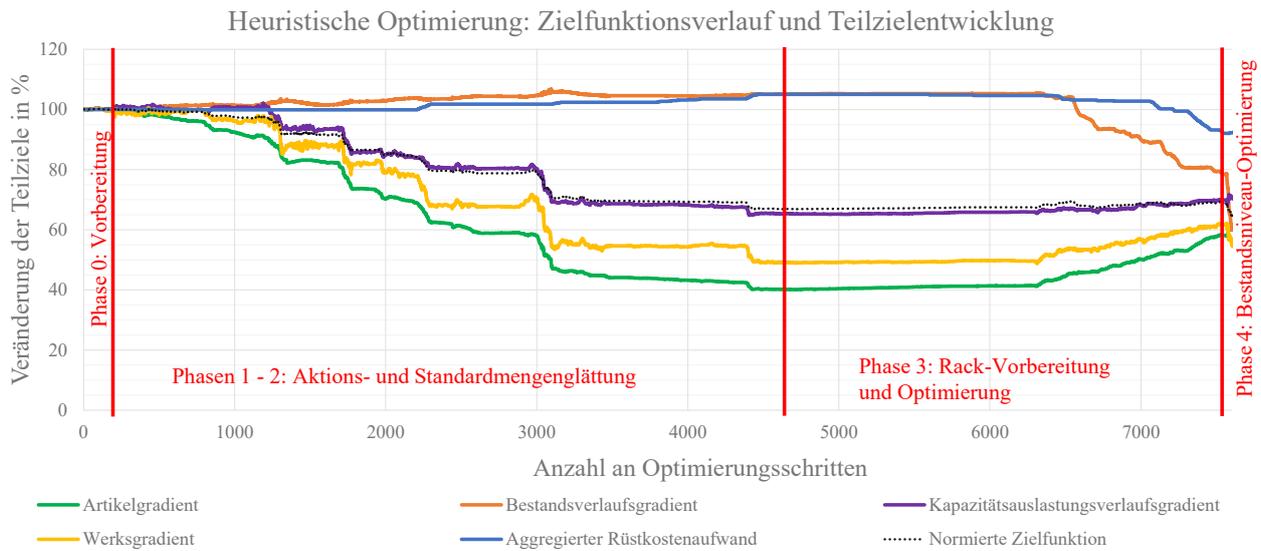


Abb. 6.1: Optimierungsergebnisse der heuristischen Optimierung, Datensatz 1 (vgl. Kamhuber et al. 2020, S. 10)

Diese Ergebnisse werden erreicht, indem je nach individuellem Datensatz nur etwa 10 - 12% der jährlichen Produktionsvolumina um einige Perioden verschoben werden, siehe auch Tabelle 6.1 (vgl. Kamhuber et al. 2020, S. 10) auf Basis von Datensatz 2. Diese Tabelle verdeutlicht, dass rund 90% der versetzten Produktionsvolumina um maximal vier Perioden bzw. Wochen versetzt werden. Die Glättung erfordert insgesamt, bedingt durch die Logik der Optimierungsmethodik in den Phasen 3 und 4, ggf. weniger Produktionslose. Die Verschiebung bzw. Aggregation von (Teil-) Losen in bestehende Lose aus anderen Perioden resultiert insgesamt in weniger Produktionslosen, siehe auch den aggregierten Rüstkostenaufwand in Abb. 6.1. Außerdem bedingt die Optimierung zusätzliche Kapazitäten im Reifelager bzw. eine entsprechend größere Dimensionierung im Zuge der Planung des Reifelagers. Die in den Phasen 1 und 2 erzeugten zusätzliche Lose werden während der gesamten heuristischen Optimierung von Teilziel $f_5(p_{ij})$ bewertet und in den Phasen 3 - 4 (durch taktische Rundungen) minimiert. Diese Bedingungen führen zu zusätzlich erforderlichen Investitionen in die erweiterte Reifelagerkapazität sowie ggf. während der Optimierung (v.a. in den Phasen 1 - 2 bzw. temporär) zu höheren Gesamtrüstkosten, welche im Anwendungsfall auf Grund der Spezifika pro Loswechsel jedoch relativ gering ausfallen.

Tab. 6.1: Heuristische Versatzmengen im Verhältnis zur Gesamtproduktionsmenge (vgl. Kamhuber et al. 2020, S. 10)

Anzahl an Versatzperioden (Wochen)	1	2	3	4	≥ 5
Relativer Anteil an der Gesamtproduktionsmenge	1.9%	2.4%	2.7%	3.6%	< 1%

Der Vergleich von Abb. 6.1 und 6.2 verdeutlicht, dass die Methodik auf beiden Datensätzen ein ähnliches Optimierungspotential entfaltet, dabei jedoch die Länge bzw. Dauer der einzelnen Optimierungsphasen je nach Datensatz stark schwankt. So fallen die ersten beiden Phasen, je nach Aktionsvolumenanteil am Gesamtproduktionsvolumen, länger (Datensatz 1) oder auch kürzer (Datensatz 2) aus. Ein ähnliches Bild ergibt sich beim Vergleich von Phase 3. Die einzelnen Teilziele entwickeln sich jedoch, unabhängig von den Längen der jeweiligen Optimierungsphasen, relativ ähnlich.

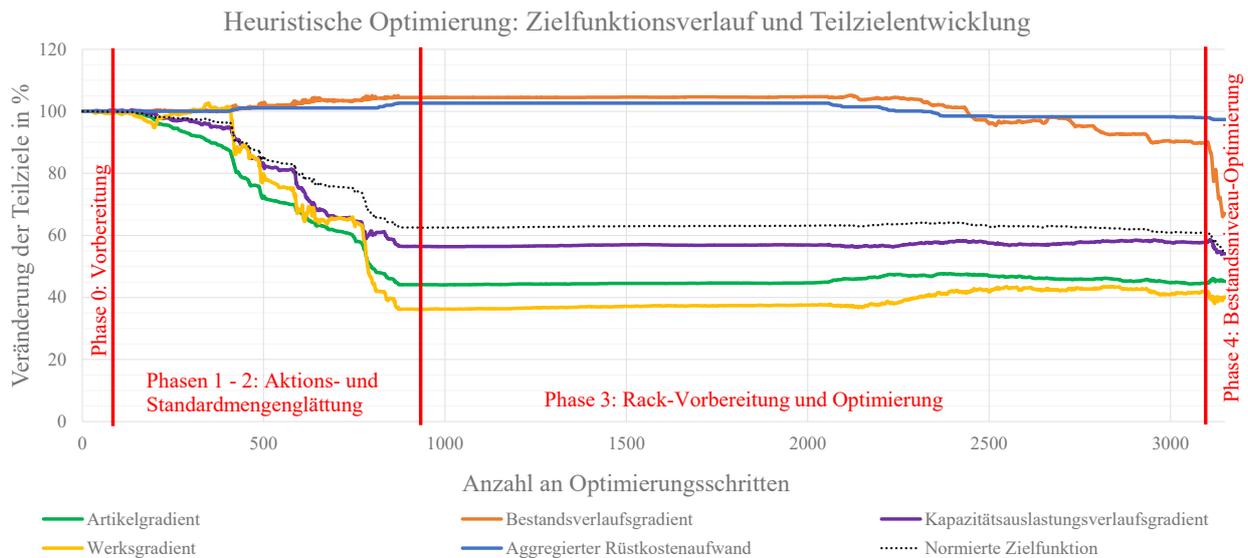


Abb. 6.2: Optimierungsergebnisse der heuristischen Optimierung, Datensatz 2 (vgl. Kamhuber et al. 2020, S. 10)

Die durch den Bau eines größeren Reifelagers zusätzlich anfallenden Investitionskosten fallen, in Anbetracht des konkret vorliegenden industriellen Anwendungsfalles, deutlich geringer aus im Vergleich zu den Einsparungen in zweistelliger Millionenhöhe, die durch den Bau und den harmonisierten bzw. geglätteten Betrieb der Produktionsanlagen mit einer optimierten Kapazitätsauslastung für die Engpassanlage als auch der nachfolgenden Grundproduktionskapazitäten erzielt werden.

In diesem Zusammenhang ist es wichtig, dass die Dimensionierung und der Betrieb der Produktion nicht unabhängig voneinander möglich ist: Wenn die Dimensionierung des Werks entsprechend geringer ausfällt, ist die Produktion als Konsequenz daraus geglättet umzusetzen. Die im Rahmen dieser Arbeit entwickelte Methodik sieht dabei vor, dass ein gewisser Anteil am Produktionsvolumen dabei zeitlich früher begonnen wird und mit einer entsprechend längeren Durchlaufzeit im Reifelager ausgestattet ist. Die Methodik führt diese Produktionsanteile entsprechend gekennzeichnet mit. Die Ergebnisse für beide Datensätze mit der gleichen Gewichtung zeigen im Gesamtbild unterschiedliche Optimierungsverläufe auf Teilzelebene, siehe Tabelle 6.2. Die Länge der Optimierungsphasen variiert in Abhängigkeit von den einzelnen Datensätzen, ihrem spezifischen Standard- und Aktionsvolumen sowie der Qualität der von der Planung und IT für die Heuristik bereitgestellten Ausgangslösung. Bezüglich der Ablaufreihenfolge innerhalb der PGH stellt sich ergebnisseitig heraus, dass die Aktionsmengenglättung ggü. der Standardmengenglättung zu priorisieren ist.

Tabelle 6.2 stellt eine ergebnisorientierte Zusammenfassung der beiden Abbildungen 6.1 und 6.2 auf Teilzielfunktionsebene dar und verdeutlicht das Funktions- und Wirkprinzip der lexikographischen Optimierung (vgl. Kamhuber et al. 2020, S. 11): Dabei geht hervor, dass die Teilziele f_1 und f_2 , die Artikel- und Werksgradienten, am stärksten verbessert worden sind. Teilziel f_3 , der Kapazitätsauslastungsverlaufsgradient, lässt sich auf Grund des notwendigen Simulationsfeedbacks nur implizit, also nicht gezielt, verbessern und ist offensichtlich stärker vom Datensatz abhängig. Das vierte Teilziel f_4 , der Bestandsverlaufsgradient, profitiert v.a. von der gezielten Optimierung in der vierten heuristischen Optimierungsphase. In Bezug auf Teilziel f_5 , die aggregierten Rüstkostenaufwände, wird an dieser Stelle angemerkt, dass sich die Gesamtlosanzahl gegenüber der Initiaillösung kaum reduziert bzw. verändert hat. Der aggregierte Rüstkostenaufwand wird im Laufe der gesamten heuristischen Optimierung i.d.R. um einige Prozentpunkte verbessert. Dies liegt vorwiegend an Phase 3. Dort werden gezielt Paletten für Racks gebildet, sodass einige Produktionsmengen in bestimmten Perioden nicht produziert und stattdessen vorversetzt werden, wodurch sich so (trotz ggf. zuvor in den Phasen 1 – 2 gesteigener Losanzahl) geringere Gesamtrüstkosten ergeben.

Tab. 6.2: Teilzielergebnisse der heuristischen Optimierung (vgl. Kamhuber et al. 2020, S. 11)

Szenario / Teilziel	f_1	f_2	f_3	f_4	f_5
Datensatz 1	-39.6%	-43.6%	-29.7%	-38.6%	-7.7%
Datensatz 2	-54.8%	-59.8%	-45.9%	-33.1%	-2.6%

Abhängig von den Gewichtungen und den spezifischen Datensätzen wird der gewichtete Fitnesswert im Vergleich zur unbehandelten Ausgangslösung um insgesamt etwa 30 - 45% verbessert. Der Schwerpunkt der heuristischen Optimierung liegt auf der Nivellierung der Produktionsmengen auf Artikel- und Werksebene.

Der Optimierungfortschritt aller Teilziele während der vier Phasen ist konsistent und die Dimension der globalen Zielfunktionsverbesserungen in ihrer jeweiligen Amplitude für verschiedene Datensätze vergleichbar. Diese Konsistenz der Optimierungsmerkmale ist eine Hauptanforderung für die wissensbasierte, heuristische Optimierung. Sie stellt sicher, dass die Ergebnisse im praktischen Planungsprozess ohne weitere (menschliche) Aufsicht oder Anpassung bereits in dieser Form verwendbar und umsetzbar sind.

6.2.2 Ergebnisse der vergleichenden metaheuristischen Optimierung

Im Zuge der metaheuristischen Optimierung werden die Präferenzen durch den Gewichtungsvektor $W = [1.25, 1.25, 1, 1, 0.5]$ dargestellt. Dies entspricht den primären Zielen einer geglätteten Produktion (in Bezug auf f_1 bzw. f_2) und den strategischen Zielen von Planung und IT ($f_1 - f_4$) als relativ ähnlich gewichtete Teilziele. Die folgenden Teile der Arbeit beinhalten die Bestimmung der optimalen Parameter für das Simulated Annealing-Verfahren bzw. den Genetischen Algorithmus mittels eines rechnergestützten Verfahrens, gefolgt von einem Vergleich der beiden (optimal) parametrisierten Algorithmen zur Auswahl des insgesamt robusteren Verfahrens für die hybride Optimierung.

6.2.2.1 Parametrisierung des Simulated Annealing-Verfahrens

Im Rahmen der rechnergestützten Studie werden die für das Simulated Annealing-Verfahren relevanten Parameter systematisch modifiziert, um möglichst optimale Parameter zu ermitteln und das Optimierungsverhalten hinsichtlich Robustheit und Konsistenz zu bewerten. Tabelle 6.3 gibt einen Überblick über die Parametersätze für die getesteten Szenarien basierend auf dem Standardszenario 1. In jedem dieser Szenarien wird systematisch immer ein Parameter (**fett** gekennzeichnet) im Vergleich zum vorherigen Szenario variiert. Zuerst wird dabei in Bezug auf den initial gefundenen Abkühlungsplan die Anfangstemperatur T_{init} modifiziert, gefolgt von einer modifizierten Gefriertemperatur T_{min} und einer verlangsamteten Auskühlgeschwindigkeit α . Schlussendlich werden die Iterationen I_T pro Temperatur verändert. Die Reihenfolge der Modifikation der Parameter beruht dabei auf der Logik, dass zuerst die Anfangs- und Endtemperatur für den Abkühlungsplan gefunden wird. Auf Basis dieser Rahmenbedingungen wird die Abkühlgeschwindigkeit bestimmt. Das Feintuning wird durch die Modifikation der Anzahl an Bewertungen pro Temperatur abgeschlossen. Die Studie wird auf Basis von Datensatz 2 durchgeführt, weil dieser für die Bewertung des stationären Produktionssystems die größte Relevanz aufweist (vgl. Kamhuber et al. 2020, S. 11f.).

Tab. 6.3: Simulated Annealing: Parametersätze und Szenarien (vgl. Kamhuber et al. 2020, S. 11)

Szenario / Parameter	I_T	T_{init}	T_{min}	α
Szenario 1	100	0.001	0.00001	0.50
Szenario 2	100	0.01	0.00001	0.50
Szenario 3	100	0.005	0.00001	0.50
Szenario 4	100	0.0025	0.00001	0.50
Szenario 5	100	0.0005	0.00001	0.50
Szenario 6	100	0.0005	0.000005	0.50
Szenario 7	100	0.0005	0.000005	0.75
Szenario 8	100	0.0005	0.000005	0.90
Szenario 9	200	0.0005	0.000005	0.90
Szenario 10	300	0.0005	0.000005	0.90

Basierend auf der skalierten Zielfunktion und dem Umstand, dass nur eine Produktionsänderung pro Optimierungsschritt durchgeführt wird, ist die Gesamtzielfunktionsdifferenz pro Schritt relativ gering und variiert zwischen 10^{-4} und 10^{-2} Fitnesspunkten. Dementsprechend muss die Anfangstemperatur bereits auf einem sehr niedrigen Niveau eingestellt werden. Tabelle 6.4 zeigt die Verbesserungen der Zielfunktion (ZF) der in Tabelle 6.3 definierten Szenarien bzw. Parametersätze gegenüber der Basislösung von Datensatz 2. Die auf Basis der Optimierungsverläufe erstellte Abb. 6.3 zeigt die Gesamtzielfunktionsverläufe in Bezug auf den ausgewählten Gewichtungsvektor in dieser Fallstudie (vgl. Kamhuber et al. 2020, S. 11f.).

Tab. 6.4: SA Optimierungsergebnisse pro Szenario (vgl. Kamhuber et al. 2020, S. 11)

Szenario	1	2	3	4	5	6	7	8	9	10
ZF-Verbesserung	-8.3%	+1.8%	-3.1%	-7.5%	-9.4%	-9.4%	-9.7%	-10.0%	-10.4%	-9.9%

Die in Tab. 6.3 definierten Szenarien, die in Abbildung 6.3 untereinander verglichen werden, zeigen deutlich, dass die Szenarien 8 und 9 die anderen übertreffen, siehe Tab. 6.4.

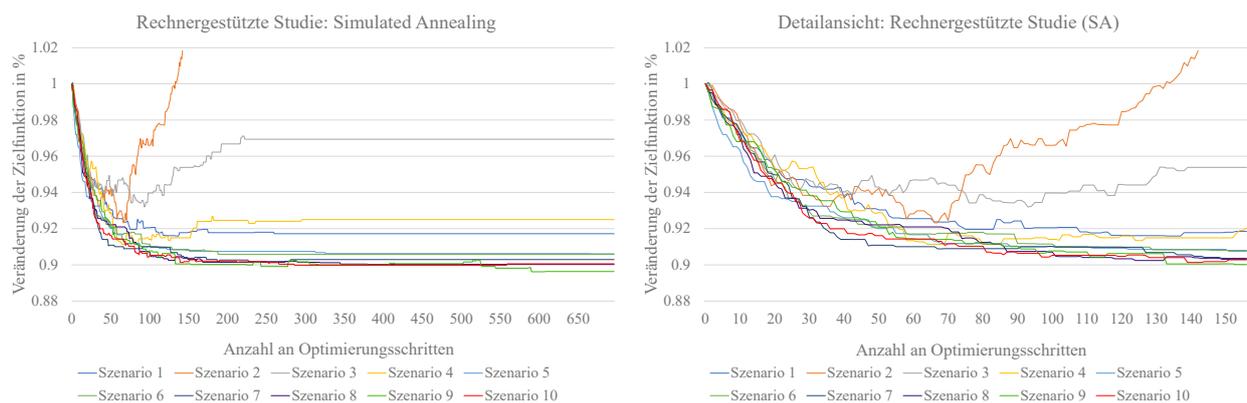


Abb. 6.3: SA Optimierungsergebnisse der rechnergestützten Studie (vgl. Kamhuber et al. 2020, S. 12)

Die daraus resultierende Empfehlung für das vorliegende spezifische Problem, die auf der rechnergestützten Studie basiert und eine möglichst gute Annäherung an die beiden besten Szenarien liefert, ist in Tabelle 6.5 (vgl. Kamhuber et al. 2020, S. 12) dargestellt und wird in weiterer Folge ausgewertet für den algorithmischen Vergleich bezüglich Laufzeit und Lösungsqualität mit dem GA herangezogen.

Tab. 6.5: Empfehlung für die SA-Optimierungsparameter (vgl. Kamhuber et al. 2020, S. 12)

Iterationen pro Temperatur I_T	Anfangstemperatur T_{init}	Gefriertemperatur T_{min}	α
100 – 200	0.0005	0.000005	0.90

6.2.2.2 Optimierungsergebnisse des Simulated Annealing-Verfahrens

Eine detaillierte Betrachtung bezüglich des Fortschritts der SA-Optimierung anhand von Abb. 6.4 zeigt, dass sich der Mechanismus der gezielten lokalen Suche als sehr hilfreich erweist, um den Werksgradienten innerhalb weniger hundert Optimierungsschritte um weitere ca. 35% zu reduzieren, während sich der Artikelgradient dabei nur um weniger als 5% verschlechtert. Als vorteilhafter Nebeneffekt wird auch der Bestandsverlaufgradient um ca. 10% verbessert. Die gesamte Zielfunktionsverbesserung durch Simulated Annealing beträgt, mittels der besten Parametrisierung (Szenario 9) auf Datensatz 2, etwa 10% im Vergleich zur besten Lösung aus der heuristischen Optimierung.

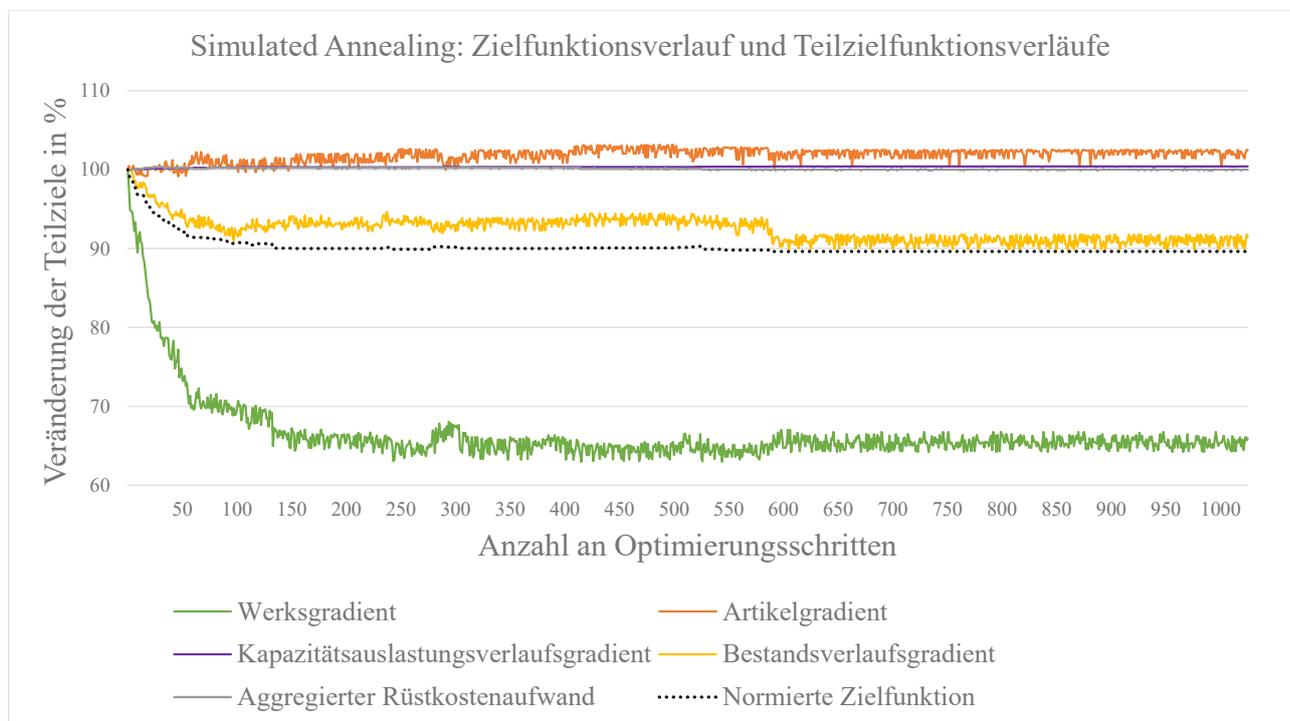


Abb. 6.4: Detailansicht: SA-Optimierung (Quelle: eigene Darstellung)

Abb. 6.5 zeigt die Ergebnisse der hybriden Optimierung (Datensatz 2, Szenario 9), also der sequenziellen Ausführung von Heuristik und Simulated Annealing. Sowohl Abb. 6.4. als auch Abb. 6.5 verdeutlichen, dass der SA Algorithmus bereits nach relativ wenigen (rund 1000) Verbesserungen bzw. Optimierungsschritten keine weitere signifikant bessere Lösung mehr findet. Gleichzeitig ist in Abb. 6.4 ersichtlich, dass der SA Algorithmus nach ca. 600 Optimierungsschritten noch einen Abtausch von zwei Teilzielen durchführt (Verbesserung des Bestandsverlaufgradienten gegenüber einer Verschlechterung des Werksgradienten). Der algorithmische Vergleich mit dem GA wird zeigen, ob der GA die gleiche Charakteristik aufweist und dem SA unterlegen oder überlegen ist. Die Teilzielfunktionsergebnisse von Abb. 6.5 sind kompakt in Unterabschnitt 6.2.3 in Tab. 6.9 ersichtlich.

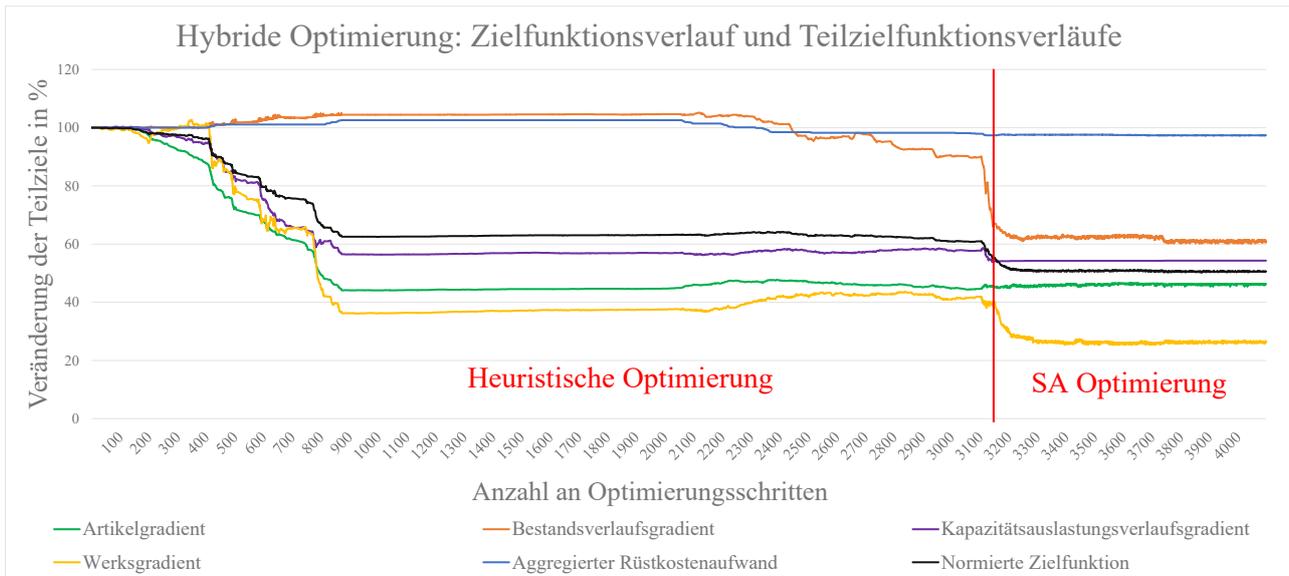


Abb. 6.5: Ergebnisse der kombiniert heuristischen und SA Optimierung, Datensatz 2 (vgl. Kamhuber et al. 2019, S. 2114)

6.2.2.3 Implementierung und Optimierungsergebnisse des Genetischen Algorithmus

Die Parametrisierung des GA (Variante 1 bzw. GA-V1) umfasst, neben der Verwendung von kleinen Populationsgrößen (max. 36 Individuen (Ind.) pro Generation), ein Elite-Individuum (EI). GA-V1 verwendet 2 Operatoren (Selektion u. Mutation) und deaktiviert (vorläufig) den in Abschnitt 5.6 entwickelten Crossover-Operator für den GA.

Abbildung 6.6 vergleicht die Nutzung verschiedener Populationsgrößen für kurze (bis zu 4.000 Zielfunktionsauswertungen (ZFAen)) und mittlere Läufe (bis zu 18.000 ZFAen) und zeigt, dass ein GA mit einer kleinen Population (3 - 6 Ind.) für kürzere Optimierungsläufe (max. 12.000 ZFAen) sehr effizient ist, während er die gleiche Lösungsqualität (16% Optimierungspotenzial) wie größere Populationen bietet. Bei längeren Optimierungsläufen (rund 70.000 ZFA) ist ein GA mit Populationen von der Größe eines Rudels (12 - 24 Ind.) etwas effizienter als ein GA mit einer kleineren Population. Im Zuge der Auswertung werden pro Populationsgröße jeweils 2 Optimierungsläufe auf Datensatz 3 durchgeführt (vgl. Kamhuber et al. 2020, S. 12).

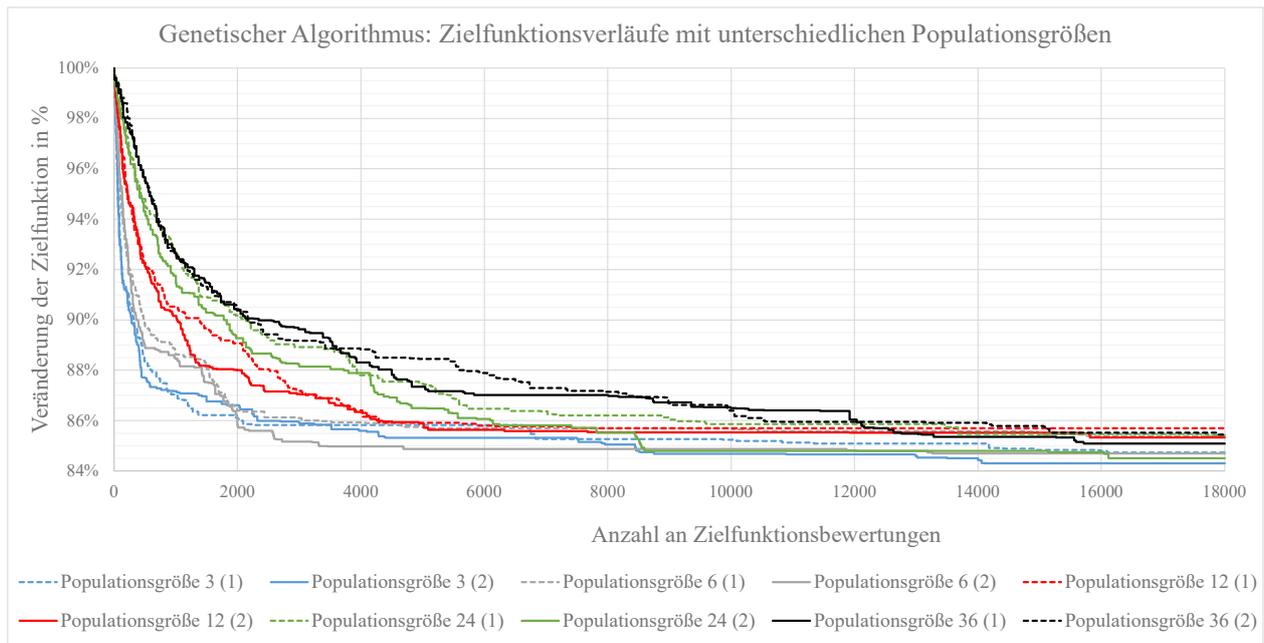


Abb. 6.6: GA Optimierungsergebnisse für verschiedene Populationsgrößen, Datensatz 3 (vgl. Kamhuber et al. 2020, S. 12)

Tabelle 6.6 fasst die GA-Empfehlungen (Variante 1, abgekürzt GA-V1) auf Grundlage des auf Datensatz 3 gelösten Problems zusammen. Auf Grund des Vorliegens einer bereits relativ guten heuristischen Lösung lässt sich erklären, dass die GA-Parametrisierungen mit durchwegs wenigen Individuen auskommen. Für die Ausgangspopulation werden in jeder Parametrisierung Kopien der besten Lösung aus der Heuristik verwendet. Die Optimierung stagniert nach ca. 50.000 Auswertungen und erreicht dabei ein Optimierungspotential von maximal 17% (vgl. Kamhuber et al. 2020, S. 11f.).

Tab. 6.6: GA Empfehlungen (GA-V1) für verschiedene Laufzeiten (vgl. Kamhuber et al. 2020, S. 14)

GA Laufzeit (Anzahl an ZFA)	≤ 5.000 ZFA	≤ 18.000 ZFA	18.000 – 72.000 ZFA
Empfohlene GA Parametrisierung	3 – 6 Ind., 1 EI	3 – 12 Ind., 1 EI	12 – 36 Ind., 1 – 2 EI
Optimierungspotential des GA	max. 15%	max. 16%	max. 17%

6.2.2.4 Vergleich der beiden Algorithmen: SA versus GA

Der Vergleich zwischen dem SA und dem GA erlaubt zwei zentrale Schlussfolgerungen:

Erstens übertrifft der GA den besten SA-Lauf in den im Zuge des Vergleichs durchgeführten Läufen immer und in jeder Parametrisierung, außer bei sehr kurzen Läufen mit max. 1.500 Zielfunktionsauswertungen, abhängig von der gewählten Populationsgröße. Jenseits dieser Optimierungslaufzeit stagniert der SA und bleibt in der bisher besten, gefundenen Lösung *stecken* oder fällt zurück, während die GA-Optimierung immer noch in der Lage ist, iterativ, unter den gleichen Suchbedingungen und gleichwertig umgesetzten, algorithmischen Such- und Modifikationsprozeduren, weiter bessere Lösungen zu finden. Die Differenz zwischen dem SA und dem GA beträgt je nach Parametrisierung nach 4.000 Auswertungen etwa 2 – 6% und wird mit zunehmender Dauer der GA-Optimierung größer - bis zu 8%. Während der beste SA-Lauf ca. 9% Optimierungspotential auf Datensatz 3 aufweist, siehe dazu auch Abb. 6.7 und nach relativ kurzer Zeit stagniert, schafft der GA in längeren Optimierungsläufen bis zu 17%, siehe Tab. 6.6. Abb. 6.7 verdeutlicht die Stagnation des Zielfunktionsverlaufs beim SA nach bereits ca. 1500 Zielfunktionsauswertungen, während sowohl der schlechteste als auch beste GA-Lauf, deren Parametrisierungen sich durch die Populationsgröße und Anzahl an Elite-Individuen unterscheiden, das Optimierungspotential des SA bereits nach ca. 1500 Bewertungen übertreffen.

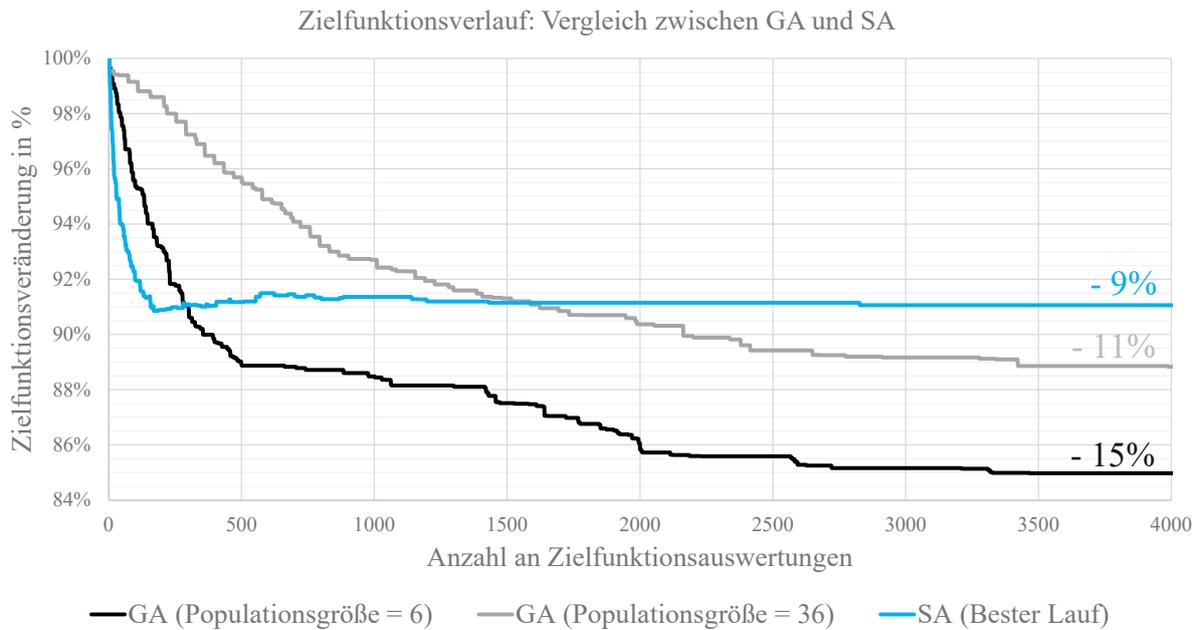


Abb. 6.7: Vergleich zwischen GA und SA, Datensatz 3 (vgl. Kamhuber et al. 2020, S. 13)

Zweitens ist die beste Parametrisierung des SA, die aus einer rechnergestützten Studie auf der Grundlage von Datensatz 2 abgeleitet wurde, nicht gleichzeitig auch die beste Parametrisierung für Datensatz 3. Optimierungsläufe auf Datensatz 3 mit der Parametrisierung von Szenario 9 aus Tabelle 6.3 zeigen ein starkes *Fallback*¹ der besten Lösung während der SA-Optimierung, im Gegensatz zu den Optimierungsverläufen auf Datensatz 2. Der GA ist durch seinen populationsbasierten Charakter weniger anfällig für ein derartiges Verhaltensmuster auf unterschiedlichen Datensätzen und wird deshalb für die hybride Optimierung empfohlen. Das zweite Kriterium ist insbesondere für die Methodik entscheidend, weil sich die Struktur der Datensätze im Detail (Mengen pro Produktionsartikel, Auslastungsschwankungen pro Artikel und Periode, etc.) im Zuge der rollierenden Planung laufend ändert. In Bezug auf die GA-Läufe von Abb. 6.7 mit 6 und 36 Individuen lässt sich in Analogie zu Abb. 6.6 außerdem festhalten, dass der GA mit der größeren Population mit zunehmendem Optimierungsverlauf sukzessive aufholt und die Zielfunktion nach 18.000 ZFAen um ca. 14.6% verbessert hat, während der GA mit der kleineren Population stagniert und die ZF auf 15.3% kaum weiter verbessert (vgl. Kamhuber et al. 2020, S. 11f.).

6.2.2.5 Aktivierung der Rekombination (GA)

In Unterabschnitt 5.6.2 wurde vorläufig auf die Aktivierung des vorgestellten Rekombinations-Operators verzichtet. Nachdem der GA bereits ohne Rekombination und nur mit einem gleichwertig implementierten Mutations-Operator den SA übertrifft, stellt sich dieser Abschnitt der Arbeit die Frage, ob und unter welchen Umständen bzw. in welchem Anteil sich der vorgestellte Rekombinations-Operator ggf. als sinnvolle Ergänzung zur bisherigen hybriden Optimierungsmethode erweist. Im Zuge der Aktivierung des Rekombinations-Operators wird ein Individuum im in dieser Arbeit entwickelten GA nach der Selektion also entweder mutiert oder gekreuzt. Dadurch wird vorgebeugt, dass ein einzelnes Individuum mehrere Änderungen auf einmal erfährt, ohne zwischenzeitlich eine Aktualisierung des dazugehörigen Zielfunktionswerts zu erhalten. Abb. 6.8 basiert auf Abb. 5.8 und erweitert diese nun um jeweils 2 Läufe mit unterschiedlichen Rekombinationsraten zwischen 20% und 80% bei einem GA mit 12 Individuen. Dabei werden mit Bezug auf Abb. 6.8 insgesamt 3 Sachverhalte festgestellt:

- In kurzen Läufen mit weniger als ca. 4.000 ZFAen (~ 15 Min. Laufzeit) ergibt die Integration des explorativen Rekombinations-Operators unabhängig vom Rekombinations-Anteil aus Sicht der Optimierung (auf diesem Datensatz) keinen Mehrwert.

¹ *Fallback* bedeutet in diesem Zusammenhang, dass der Algorithmus mit der gewählten Parametrisierung (zu) häufig dazu neigt, schlechtere Lösungen zu akzeptieren. Dies ist ein Indiz dafür, dass die gewählte Parametrisierung suboptimal ist.

- In mittleren (ca. 20.000 ZFAen ~ 75 Min. Laufzeit) und längeren Läufen (mit ~ 30.000 ZFAen) stellt sich ein Anteil zwischen 20% und 60% als optimal heraus, wobei eine Rekombinationsrate mit 40% empfohlen wird. Die Integration der Rekombination mit diesen Raten übertrifft den nur auf Mutationen basierendem GA um 1 – 2.5% in Abhängigkeit des konkreten Laufs, der dazugehörigen Parametrisierung, und dem jeweiligen Datensatz.
- Rekombinationsraten mit 80% oder höher werden nicht empfohlen, weil der GA durch den explorativen Charakter des Rekombinations-Operators zunehmend vom Optimierungsziel ablenkt wird und dementsprechend teilweise langsamer konvergiert als ein GA ohne Rekombination.

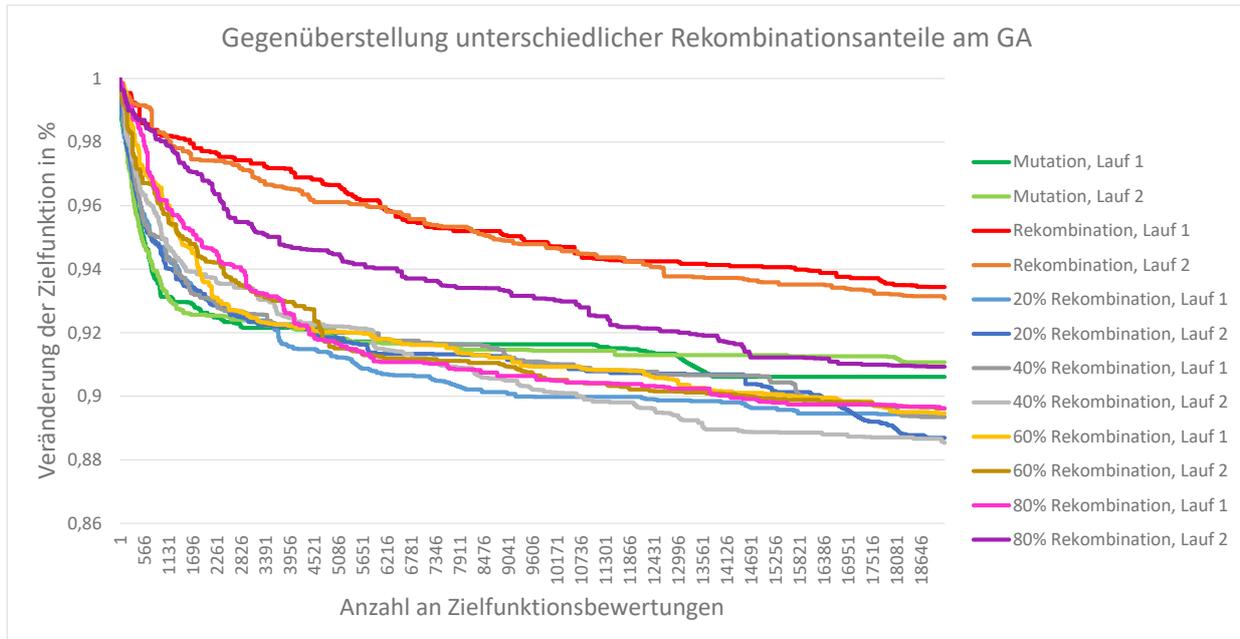


Abb. 6.8: Vergleich unterschiedlicher Rekombinationsraten, Datensatz 4 (Quelle: eigene Darstellung)

Im Zuge der Validierung der Ergebnisse von Abb. 6.8 werden diese auf Grund der Bedeutung der Thematik für die Methodik auf einem ähnlichen Datensatz 4', welcher Datensatz 4 in Bezug auf den Zeitraum ähnelt, wiederholt durchgeführt. Dabei stellt sich auf Abb. 6.9 ein prinzipiell ähnliches Bild dar, wobei der Rekombinations-Operator anteilig mit 40 – 60% als beste Kombination aus der Optimierung hervorgeht. Die Integration des Rekombinations-Operators gibt bereits nach bereits ~ 1.000 ZFAen einen bestimmten Mehrwert, welcher nach 25.000 ZFAen je nach Anteil zwischen 1.5 und 4% an der gesamten Optimierung beträgt. Längere Optimierungsläufe mit bis zu 80.000 ZFAen (12 Individuen, 40% Rekombinationsanteil) haben kaum bzw. wenig zusätzliche Verbesserung (max. 1%) gegenüber mittleren Läufen mit 20.000 – 30.000 ZFAen gebracht. Deshalb sind die Abbildungen auf den wesentlichen Ausschnitt aus dem Optimierungsverlauf (mittlere Läufe) begrenzt.

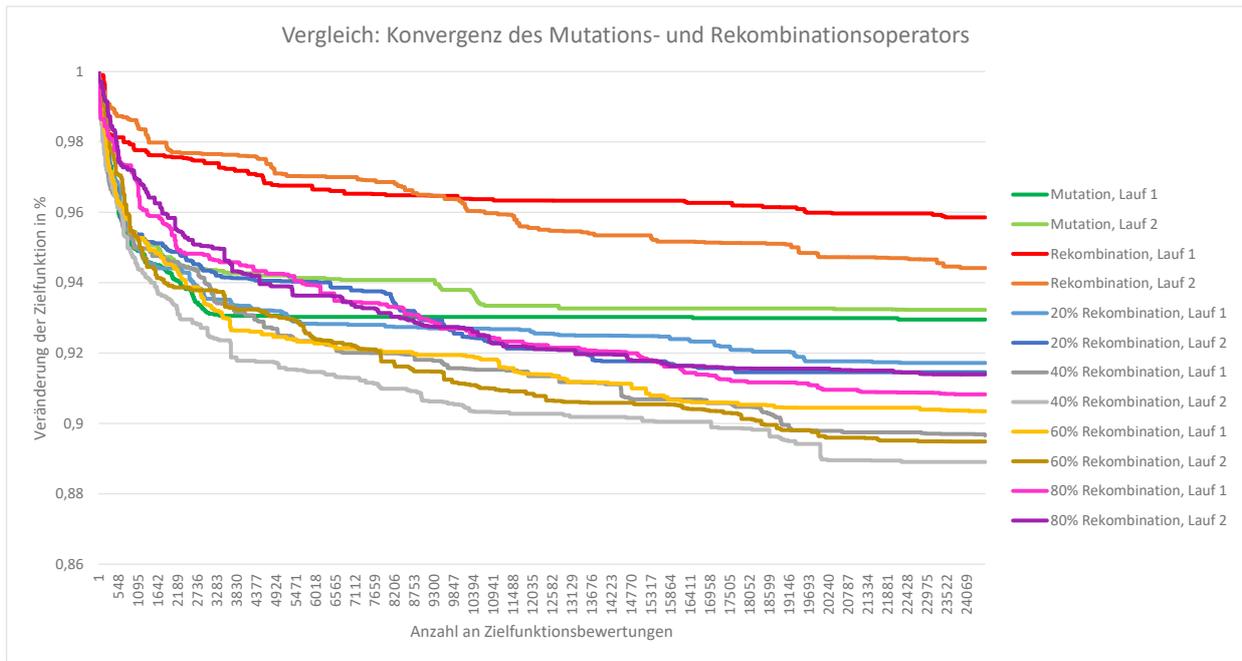


Abb. 6.9: Vergleich unterschiedlicher Rekombinationsraten, Datensatz 4' (Quelle: eigene Darstellung)

Zur Validierung der ermittelten Parametrisierung des GA (mit anteilig 40% Rekombinationsanteil) werden Läufe mit unterschiedlichen Populationsgrößen zwischen 6 und 24 Individuen durchgeführt, siehe Abb. 6.10. Dabei hat sich eine Populationsgröße von 12 Individuen als beste Konfiguration bezüglich Laufzeit und Konvergenzverhalten sowie Robustheit ergeben. Vergleichend mit Abb. 6.6 lässt sich an dieser Stelle festhalten, dass der Einsatz des Rekombinations-Operators auf Grund seines explorativen Charakters grundsätzlich ein wenig größere Populationen und Parametrisierungen mit mittleren bzw. längeren Optimierungsläufen mit ≥ 20.000 ZFAen präferiert, während in kürzeren Läufen (≤ 5.000 ZFAen) kleinere Populationen mit z.B. 6 Individuen rascher konvergieren, allerdings teilweise anfälliger für eine anschließende Stagnation sind, siehe auch Lauf 1 in Abb. 6.10.

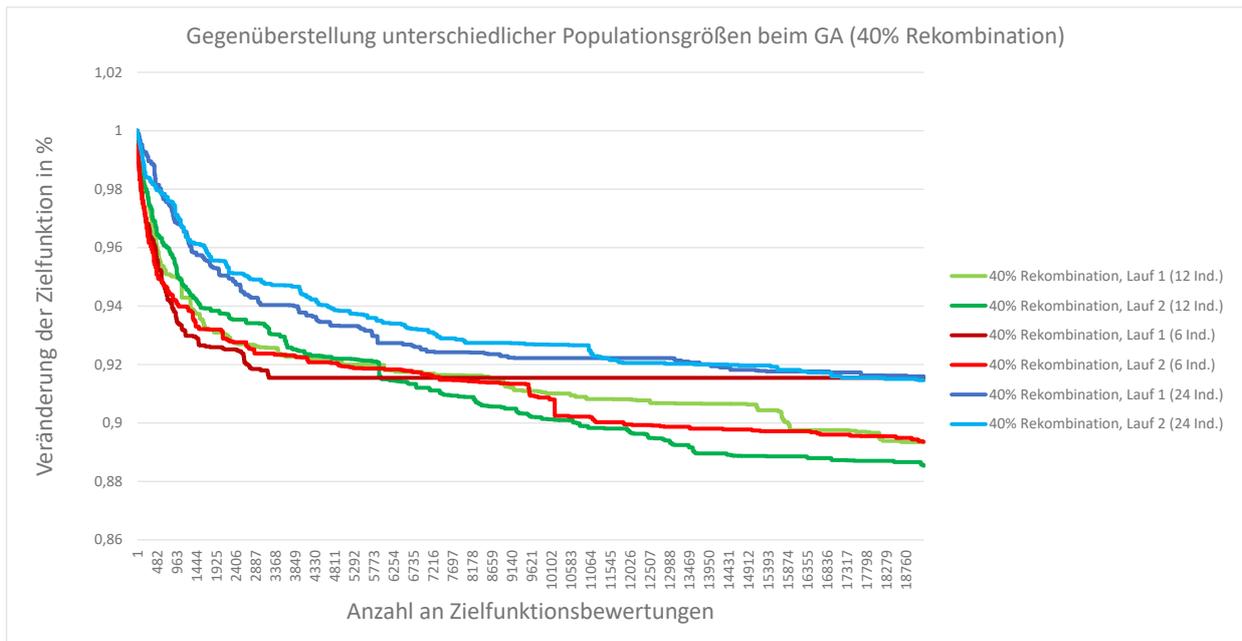


Abb. 6.10: GA mit Rekombination: Vergleich unterschiedlicher Populationsgrößen, Datensatz 4 (Quelle: eigene Darstellung)

Gleichzeitige Veränderung von Individuen mittels Mutation und Rekombination: Der folgende Paragraph widmet sich der Frage, ob die bisher gewählte GA mit Variante 2 (GA-V2), bestehend aus einer Population mit z.B. 12 Individuen, welche anteilig zu 40% rekombiniert ($p_{rek} = 0.4$) bzw. alternativ zu 60% mutiert ($p_{mut} = 0.6$) werden, eine bessere Konvergenz im Zuge der Optimierung mit sich bringt, als die ggf. gleichzeitige Mutation und Rekombination von Individuen im Zuge von Variante 3 (GA-V3). Tab. 6.7 stellt die verschiedenen GA-Varianten exemplarisch anhand einer Population mit 5 Individuen gegenüber. Variante 3 integriert pro Individuum teilweise mehrere Veränderungen simultan. In Tab. 6.7 steht S für Selektion, M für Mutation, und R für Rekombination.

Tab. 6.7: Umsetzungsvarianten des GA (Quelle: eigene Darstellung)

Umsetzungsvarianten des GA	Ind. 1	Ind. 2	Ind. 3	Ind. 4	Ind. 5
GA-V1: S + ($p_{mut} = 1.0$)	S + M	S + M	S + M	S + M	S + M
GA-V2: S + ($p_{mut} = 0.6 \parallel p_{rek} = 0.4$)	S + M	S + R	S + M	S + R	S + M
GA-V3: S + ($p_{mut} = 1.0 \wedge p_{rek} = 0.4$)	S + M	S + M + R	S + M	S + M + R	S + M

Abb. 6.11 stellt die beiden Varianten des GA mit Rekombination, den GA mit Variante 2 (GA-V2) und den GA mit Variante 3 (GA-V3), vergleichend unter der sonst identen Parametrisierung (12 Individuen, 40% Rekombinationsanteil) auf Datensatz 4' gegenüber. Dabei stellt sich heraus, dass der GA-V2 in beiden Läufen dem GA-V3 nach ca. 15.000 ZFAen zunehmend überlegen ist und diesen am Ende der Optimierung im Mittel um ca. 1.5% übertrifft. Während der GA-V3 zunehmend stagniert, stellt sich der explorativere Charakter des GA-V2 im Laufe der Optimierung als vorteilhaft heraus.

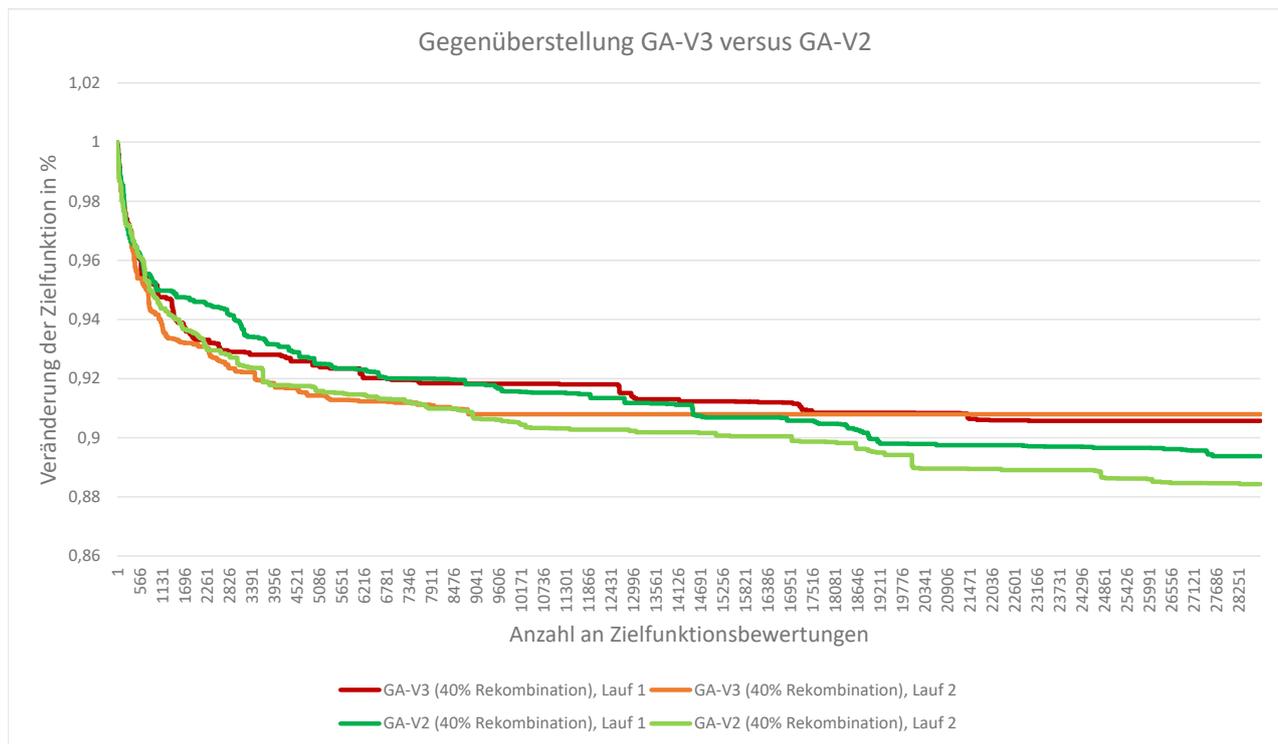


Abb. 6.11: Gegenüberstellung von GA-V2 und GA-V3, Datensatz 4' (Quelle: eigene Darstellung)

Tab. 6.8 stellt eine Empfehlung für den GA-V2 und den GA-V3 in Bezug auf verschiedene Laufzeiten auf den Datensätzen 4 und 4' dar. Aus dem Vergleich von Tab. 6.6 und Tab. 6.8 geht hervor, dass das Optimierungspotential auch erheblich vom jeweiligen, zur Verfügung stehenden, Datensatz abhängt. Auf diese Weise gelingt es einem GA-V1 auf Datensatz 3, die heuristische Lösung um bis zu 17% zu verbessern, während auf den Datensätzen 4 als auch 4' kein entsprechend umfangreicher Optimierungslauf die heuristische Lösung eines GA-V2 bzw. GA-V3 um mehr als 13% verbessert hat.

Tab. 6.8: GA Empfehlungen (V2 und V3) für verschiedene Laufzeiten (Quelle: eigene Darstellung)

GA Laufzeit (Anzahl an ZFA)	≤ 5.000 ZFAen	≤ 20.000 ZFAen	18.000 – 72.000 ZFAen
Empfohlene GA Parametrisierung	3 – 6 Ind., 1 EI	6 – 12 Ind., 1 EI	12 – 36 Ind., 1 – 2 EI
Optimierungspotential des GA	max. 9%	max. 12%	max. 13%

6.2.3 Ergebnisse der hybriden Optimierung

Die Ergebnisse der hybriden Optimierung werden durch Datensatz 2 (bester SA-Lauf, siehe Abb. 6.5), Datensatz 3 (bester Lauf von GA-V1 ohne Rekombination) und Datensatz 4' (bester Lauf von GA-V2 mit 40% Rekombination) zusammenfassend repräsentiert.

Die durch den Genetischen Algorithmus erreichte Gesamtzielfunktionsverbesserung beträgt (je nach Datensatz, Lauf und Parametrisierung) etwa 15% im Vergleich zur besten Lösung aus der heuristischen Optimierung und etwa 40 – 50% im Vergleich zur anfänglichen, manuellen Lösung, die von der Planung bereitgestellt wurde. Abbildung 6.12 (bester Lauf mit dem GA ohne Rekombination aus Abb. 6.6) zeigt die Zielfunktionsverläufe der hybriden Optimierung (Elite-Individuum) inklusive den einzelnen Teilzieltrends der beiden Optimierungsphasen. Darüber hinaus zeigt diese Abbildung beispielhafte Kompromisslösungen, sogenannte *Trade-Offs*, die während der Optimierung mittels des GA durchgeführt werden, z.B. die Verbesserung des Werksgradienten auf Kosten eines schlechteren Artikelgradienten bzw. Bestandsverlaufs (vgl. Kamhuber et al. 2020, S. 12; Kamhuber et al. 2019, S. 2113f.). Abb. 6.12 zeigt außerdem, dass die Teilziele insgesamt weitgehend gemäß ihrer (lexikographischen) Hierarchie bzw. Priorität verbessert worden sind. Das bedeutet $f_2 < f_1 < f_3 < f_4 < f_5$, wobei f_3 nicht gezielt optimierbar ist, siehe Abschnitt 5.4. Der Hauptgrund für $f_2 < f_1$ liegt in der Beschaffenheit des Datensatzes. Ab einem bestimmten Punkt in der Optimierung lässt sich der Artikelgradient nur mehr schwierig bzw. auf Kosten von anderen Teilzielen verbessern, siehe auch Abb. 5.7 diesbezüglich.

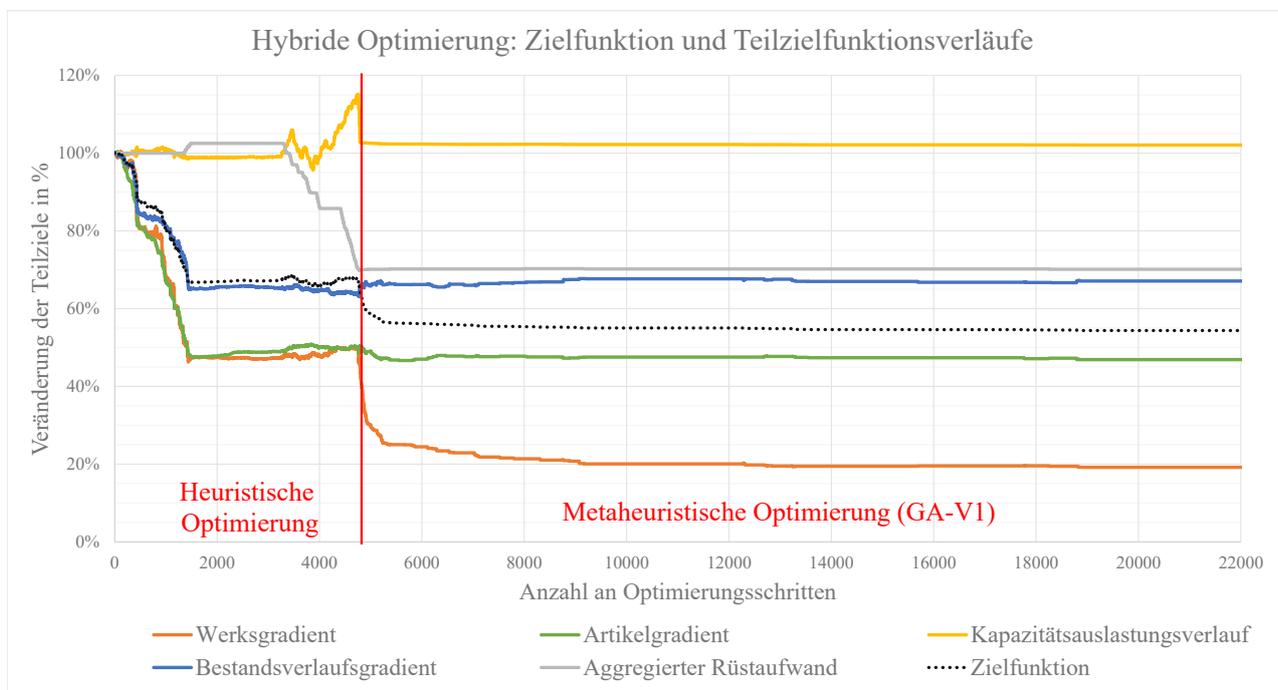


Abb. 6.12: Hybride Optimierung: PGH und GA, Datensatz 3 (vgl. Kamhuber et al. 2020, S. 13)

Tabelle 6.9 fasst die Ergebnisse der hybriden Optimierung mittels GA-V1 aus Abb. 6.12 kompakt zusammen. Die Dominanz von f_1 und f_2 kommt bereits aus der heuristischen Optimierung und bestätigt sich in den eingeschwungenen Datensätzen 2 und 3, welche durch die Abbildungen 6.2, 6.4 und 6.5 dargestellt sind. Dies ist eine Bestätigung für den methodischen Erfolg der lexikographischen,

heuristischen Optimierung. Die signifikante, weitere Verbesserung des Werksgradienten (f_2) im Zuge der gezielten metaheuristischen Optimierung bestätigt den Erfolg durch den Einsatz des gezielten Suchmechanismus, der durch den Mutations-Operator ausgedrückt wird. Die im Zuge der Metaheuristik zusätzlich entstehende Versatzmenge beläuft sich auf max. 3%. Dies liegt an den insgesamt relativ geringen Versatzmengen sowie daran, dass durch die Heuristik vorverschobene Mengen bereits als Versatzmengen gezählt werden, wenngleich sie durch die Metaheuristik weiter vorversetzt werden. Die gesamte, durch die Heuristik und Metaheuristik entstandene, Versatzproduktionsmenge beläuft sich dadurch, je nach Datensatz und Aktionsmengenanteil, in Einklang mit Tabelle 6.1, auf max. 15%. Durch die Versatzmengen werden Produktionsmengen teilweise in Perioden hineinversetzt, wo vorher keine Mengen gestanden sind. Trotzdem ergibt sich bei Teilziel f_5 eine Verbesserung von bis zu 30%.

Abb. 6.13 stellt die Ergebnisse der hybriden Optimierung auf Datensatz 4' vom GA in der Variante 2 dar. Wie auf Datensatz 3 wird der Artikelgradient im Zuge der metaheuristischen Optimierung nicht mehr signifikant verbessert. Die beiden verändernden (Mutations- und Rekombinations-) Operatoren des GA verbessern v.a. den Werksgradienten weiter. Der beste Lauf des GA-V2 übertrifft die Ergebnisse des entsprechenden GA-V1 um bis zu 4%, siehe Abb. 6.9.

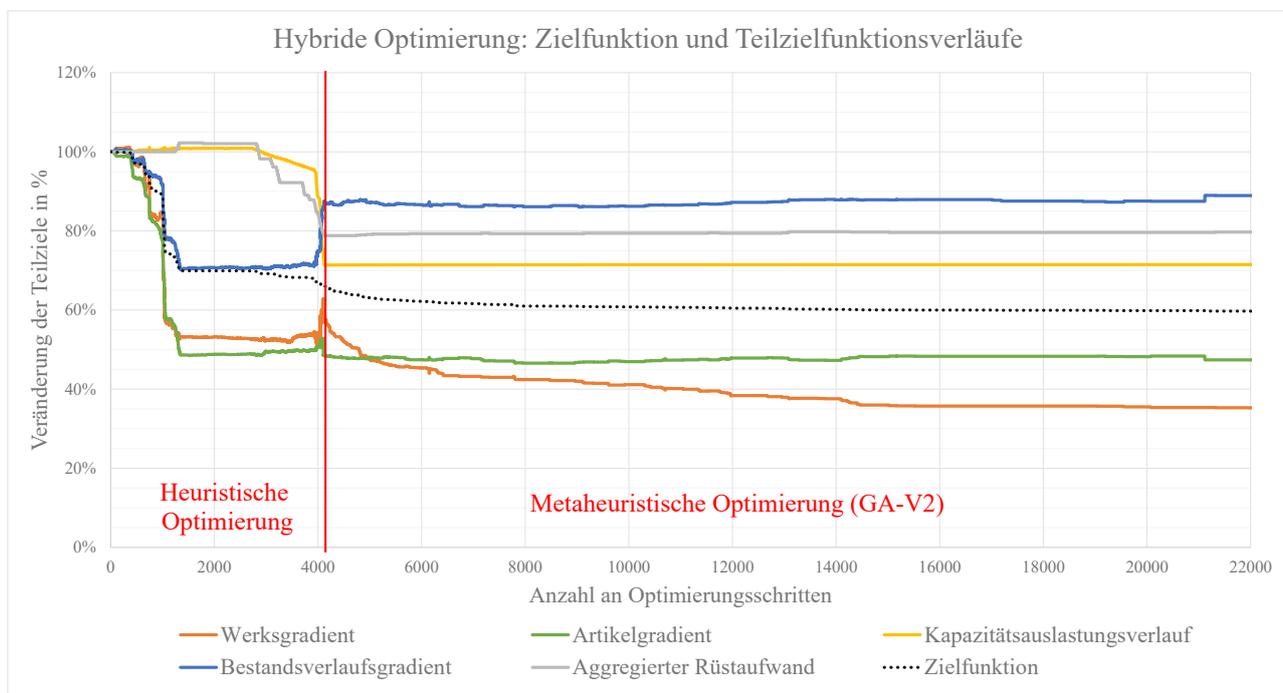


Abb. 6.13: Hybride Optimierung: PGH und GA, Datensatz 4' (Quelle: eigene Darstellung)

Tab. 6.9 fasst die hybriden Optimierungsergebnisse mittels SA (Datensatz 2), GA-V1 (Datensatz 3) und GA-V2 (Datensatz 4'), untereinander (nach 22.000 ZFAen) verglichen, kompakt zusammen. Es ist klar ersichtlich, dass der Fokus der hybriden Optimierung auf f_1 und f_2 liegt. Diese Teilziele werden auf Datensatz 4' mittels GA-V2 um 53% bzw. 65% reduziert, während die anderen Teilziele um 11 – 29% optimiert werden. Der Gesamtzielfunktionswert wird je nach Datensatz zwischen 40% und 49% verbessert. Die Ergebnisse beider GA-Implementationen ergeben auf 2 verschiedenen Datensätzen (3 bzw. 4') ein relativ einheitliches und ähnliches Bild mit einem klaren Schwerpunkt auf die Optimierung der Teilzielfunktionen f_1 und f_2 . Während sich auf den Datensätzen 2 und 4' das Teilziel f_3 prinzipiell bereits durch die Heuristik besser minimieren lässt, ist andererseits Teilziel f_4 auf den Datensätzen 2 und 3 stärker besserbar, während sich Teilziel f_5 auf den Datensätzen 3 und 4' stärker verbessern lässt. Am Ende der heuristischen Optimierung findet auf Datensatz 4' ein Trade-Off zwischen dem Bestandsverlauf und dem Kapazitäts- und Rüstaufwandsverlauf statt: Im Zuge der gezielten Bestandsreduktion wird der Kapazitäts- und Rüstaufwandsverlauf verbessert, während der Bestand anschließend (an anderen Stellen) in Ermangelung freier Raumkapazitäten nicht wieder vollständig aufgepuffert wird und sich

auf diese Weise verschlechtert.

Im Zuge dieser abschließenden Betrachtungen wird auf Grund der Überlegenheit des GA ggü. dem SA, siehe Abb. 6.7, mit Ausnahme kurzer Läufe zur unmittelbaren Rekonfiguration des Produktionssystems mittels GA-V1, der GA-V2, gemäß Tab. 6.8 parametrisiert, zur Anwendung empfohlen.

Tab. 6.9: Veränderung von f sowie der einzelnen Teilzielfunktionen (Quelle: eigene Darstellung)

GA/SA	Datensatz	Referenz	f	f_1	f_2	f_3	f_4	f_5
SA	2	Abb. 6.5	-49%	-54%	-74%	-46%	-39%	-3%
GA-V1	3	Abb. 6.12	-46%	-53%	-81%	+2%	-33%	-30%
GA-V2	4'	Abb. 6.13	-40%	-53%	-65%	-29%	-11%	-20%

6.3 Theoretische und praktische Bewertung der Laufzeitkomplexität

Die Auswertung der Laufzeitkomplexität deckt den gesamten hybriden Optimierungsansatz ab und bewertet, wie sich der Rechenaufwand sowohl der Heuristik als auch der Metaheuristik generalisiert mit zunehmender Problemkomplexität entwickelt. Die ersten beiden Phasen der Heuristik erfordern, dass der Algorithmus, einmal für die Aktions- bzw. Kurantmengen, über den gesamten Planungshorizont n und alle Artikel m jeweils einmal iteriert, was zu einer Zeitkomplexität von $\mathcal{O}(2 \cdot m \cdot n)$ führt. Die dritte Phase erfordert eine ähnliche Iteration, was zu $\mathcal{O}(m \cdot n)$ führt, während die vierte Phase nur über einen kleinen Teil (je nach Parametrisierung z.B. eine Durchlaufzeit) für alle Artikel iteriert. Dies resultiert in $\mathcal{O}(m)$. Insgesamt ergibt sich eine gesamte (theoretische) Laufzeitkomplexität von $\mathcal{O}(3 \cdot m \cdot n + m)$ für die heuristische Optimierung (vgl. Kamhuber et al. 2020, S. 12ff.).

Abbildung 6.14 bewertet die praktischen Laufzeiten im Vergleich zu ihren jeweiligen theoretischen, idealen, linearen Laufzeiten im Hinblick auf die folgenden berechneten Kernelemente: die heuristische Optimierungsphase für die Standard- und Aktionsmengen sowie die Zielfunktionsauswertung und die integrierte Simulation zur Kapazitätsauslastungsentwicklung am Beispiele des SA. Als Referenzpunkt für die Erstellung der Kurven wird der real vorliegende Planungshorizont (78 Wochen) angenommen. Die Grafik zeigt, dass die Kurven-Paare für die jeweils korrespondierende theoretische (ideale) und praktische (tatsächliche) Ist-Laufzeit eng aneinander liegen. Das bedeutet, dass die Laufzeit grundsätzlich *annähernd linear* (in Bezug auf einen zunehmenden Planungshorizont) wächst.

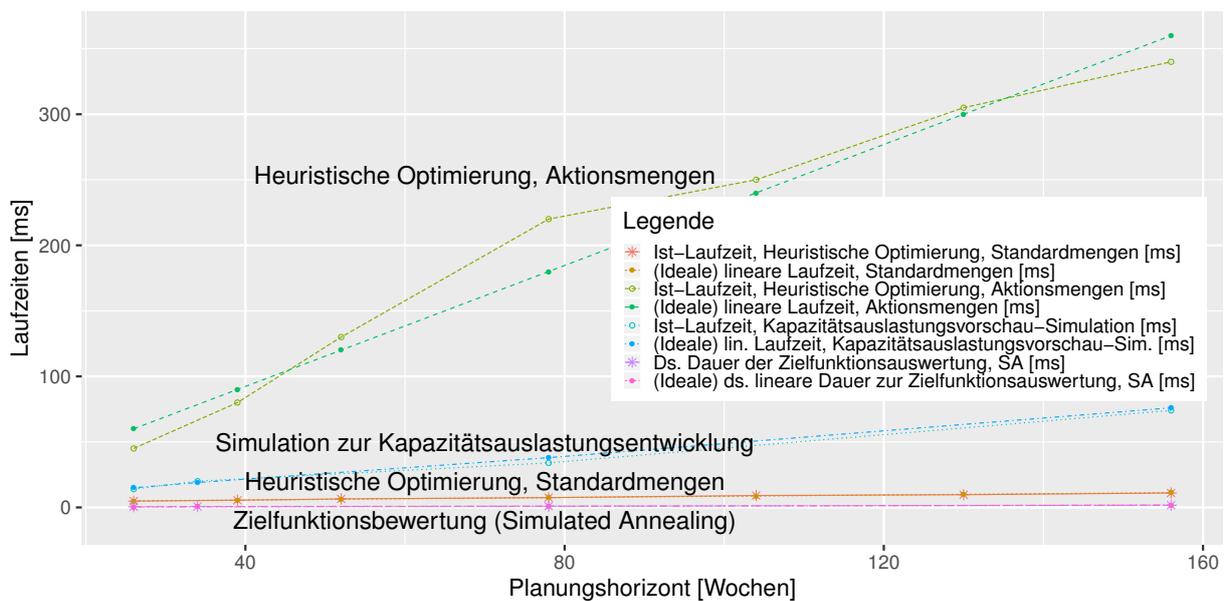


Abb. 6.14: Validierung der Laufzeitkomplexität (vgl. Kamhuber et al. 2020, S. 14)

Das SA-Verfahren als auch der GA hängen v.a. von der Umsetzung ihrer jeweiligen Operatoren und Parameter ab und weder von n noch m , mit Ausnahme der Zielfunktion. Die Herausforderung besteht auch darin, dass die Auswertung der Zielfunktion, siehe (5.1), sehr rechenintensiv ist, da hierbei die Matrizen aller Teilziele periodisch aktualisiert werden. Das ergibt $\mathcal{O}(5 \cdot m \cdot n)$. Außerdem wird vorab eine Simulationsausführung benötigt. Dementsprechend sollte die gesamte Laufzeitkomplexität *linear* ansteigen, was sich bei praktischen Tests zur Validierung der theoretischen Laufzeitkomplexität mit zunehmendem Planungshorizont, siehe Abb. 6.14, als korrekt erweist.

Laufzeit-Bewertung für Heuristik und Metaheuristik: Die resultierende Laufzeit für die gesamte Heuristik (einschließlich der Datenaufbereitung, der Überprüfung der Nebenbedingungen, der kontinuierlichen Aktualisierung aller Matrizen und der strategischen Bestandsoptimierungsphase) beträgt unter Berücksichtigung der gegenwärtig berücksichtigten Artikelsumme² ($m \sim 25$) und des aktuellen Planungshorizonts von $n = 78$ Wochen etwa 480 ms, wobei für die Berechnungen in den ersten beiden Phasen 240 ms und für die Aufpufferung 140 ms benötigt werden. Außerdem hängt sie von der Basisgröße und den Modifikationsschritten am Zeilen-Objekt ab und dauert im umfangreichsten Fall bis zu wenigen Sekunden.

Bei der metaheuristischen Optimierung wird die integrierte Simulation nach jedem erfolgreichen Optimierungsschritt einmal ausgeführt. Dies ergibt, unter den gleichen Annahmen für Artikelsumme bzw. Planungshorizont wie für die Heuristik, ca. 40 – 50 ms für jeden Simulationsdurchlauf, während die Zielfunktionsauswertung (inkl. Skalierung) ca. 2 – 3 ms beansprucht. Die Ausführung der Simulation ist dabei eine notwendige Bedingung im Rahmen der anschließenden Zielfunktionsauswertung. Der GA ist im Gegensatz zum SA je nach Einsatz seiner (bis zu drei) Operatoren in der Laufzeit intensiver, weil zusätzlich (zur Mutation) noch Selektion und (ggf.) Rekombination angewendet werden. Für die Selektion werden pro Individuum zusätzlich 70 – 80 ms aufgewendet³. Die Rekombination benötigt beim GA 15 – 60 ms in Abhängigkeit der Such- und Umsetzungsdauer für eine erfolgreiche Rekombination. Die Umsetzung der Mutation erfordert für beide Algorithmen jeweils 10 – 40 ms in Abhängigkeit der Dauer der Suche für eine geeignete Mutationsstelle bzw. eine erfolgreiche Mutation. Die Metaheuristiken erfordern also für jeden Optimierungsschritt durchschnittlich jeweils ca. 100 – 240 ms Laufzeit in Abhängigkeit vom verwendeten Algorithmus, dem Fortschritt der Optimierung, sowie dem Umfang des konkreten Datensatzes. 1000 Zielfunktionsauswertungen benötigen dementsprechend zwischen 100 und 240 Sekunden Laufzeit (Erwartungswert: ca. 3 Minuten), d.h. 5000 Zielfunktionsauswertungen ergeben eine zu erwartende Laufzeit von ca. 15 Minuten.

Die Laufzeit der heuristischen Phase repräsentiert also nur einen Bruchteil der Laufzeit der metaheuristischen Phase. Die Laufzeitengpässe in der metaheuristischen (GA, SA) Phase sind die iterativ ausgeführte Simulation, die Modifikationen an der Lösung (bestehend aus verschiedenen Matrizen, siehe Tab. 5.3) durch den Veränderungsmechanismus des SA bzw. die bis zu drei verwendeten Operatoren des GA, das Update des Versatzmappings nach jeder Modifikation, und die dazugehörige Zielfunktionsauswertung.

Effizienz-Vorteile der Heuristik: Die heuristischen Optimierungsschritte werden deutlich schneller ausgeführt als die metaheuristischen, weil die laufende Zielfunktionsauswertung entfällt, für die Durchführung der heuristischen Optimierungsschritte weniger Zeit benötigt als für die komplexeren Schritte der Metaheuristik wird und die Aktualisierung der Kapazitätsgradienten in der Metaheuristik nicht notwendigerweise⁴ zur Laufzeit der Heuristik angewendet wird. Im Zuge der Heuristik werden die Nebenbedingungen, die Ungleichungen (5.2) – (5.4) (sofern aktiv), stets vor Mengenverschiebungen

²Je nach Saison kann die Zahl der Produktionsartikel leicht variieren.

³Die Laufzeitintensität der Selektion begründet sich dabei v.a. über die Anlage von sogenannten *deep copy* Objekten im Zuge des Designs der Methode. Dabei werden die beiden Objekte voneinander disjunkt, d.h. die Änderungen am ausgewählten Individuum der neuen Population durch z.B. Mutation bzw. Rekombination werden nicht auf das entsprechende Individuum der vorherigen Population übertragen.

⁴Eine Ausnahme bildet die Aktualisierung der Gradienten zum Zwecke der nachträglichen Erstellung von Grafiken für Zielfunktionsverläufe.

überprüft. Im Unterschied zur Metaheuristik erfolgt dies konstruktiv, sodass nur die jeweils letzte Änderung mit der bisher übergebenen Raumauslastungsvorschau (5.3) bzw. der Kapazität der Engpassanlage (5.2) aktualisiert in Bezug auf die Einhaltung der entsprechenden Nebenbedingung bewertet wird. Im Zuge der Metaheuristik wird die Kapazitätsauslastungsvorschau stets neu erstellt, weil die aktuell veränderte Lösung lt. Zielfunktionssystem, siehe Gleichung (5.1), vollständig ausgewertet wird.

Während die Metaheuristik jede einzelne Modifikation an einer Zwischenlösung bewertet, ist diese Zielfunktionsauswertung für die Heuristik nicht erforderlich, weil die Zielfunktion in dieser Phase noch nicht existiert. Diese multikriterielle und daher relativ aufwändige Bewertung umfasst neben der Ermittlung einer realisierbaren bzw. gültigen Kapazitätsauslastung mittels Simulation auch die kontinuierliche Evaluierung der einzelnen Teilkapazitätsgradienten, die den Gesamtkapazitätsgradienten ergeben. Jede Produktionsänderung bedeutet, dass alle fünf Teilgradienten aus der ersten betroffenen Periode für den gesamten, verbleibenden Planungshorizont aktualisiert werden. Die Heuristik hat einen weiteren Leistungsvorteil gegenüber der Metaheuristik: Sie ist in einigen Fällen in der Lage, innerhalb einer Modifikation der Lösung mehrere Optimierungsschritte gleichzeitig durchzuführen (siehe auch Abb. 5.5). Je länger die metaheuristische Optimierung läuft, desto mehr Versuche sind notwendig, bis eine Verbesserung gefunden wird.

Conclusio: Zusammenfassend lässt sich sagen, dass die Heuristik in der Lage ist, deutlich schneller und effizienter eine gute Lösung zu erreichen, welche die operativen (f_1, f_2) und strategischen Ziele ($f_3 - f_5$) der Planung erfüllt und als Ausgangslösung für die nachfolgende, *langsamere*, metaheuristische Optimierung dient. Der GA hat eine globalere Sicht auf das Problem und ist in der Lage, zusätzliches ungenutztes Potenzial durch eine gewichtete Zielfunktionsbewertung auszuschöpfen bzw. die Pareto-Front zu erreichen. Der Nutzen und die Robustheit der hybriden Optimierungsmethode wachsen mit zunehmender Problemkomplexität, wie der interne Benchmark und die Ergebnisse der Laufzeitkomplexität zeigen. Die Methode wird mit zunehmenden Belastungsspitzen *effektiver*, während die Laufzeit weiterhin annähernd linear in Abhängigkeit des Planungshorizonts bzw. der Produktionsartikelanzahl wächst. Durch die Kombination von Heuristik und GA lässt sich die linear zunehmende Laufzeit flexibel, z.B. über ein entsprechendes Abbruchkriterium (Vorgabe der Populationsgröße bzw. Anzahl an Iterationen, Abbruch bei Stagnation, etc.) entsprechend steuern.

6.4 Ergebnisse der statistischen Datenanalyse

Die rollierende Planungsmethodik inkludiert eine statistische Datenanalyse, die in Abschnitt 5.7 diskutiert wird. Als Endergebnis dieser statistischen Analyse, ergänzt durch eine Risikobewertung in Bezug auf die Volatilität des Lagerbestandsverlaufs, wird auf der Grundlage von Schwellwerten eine Gesamtempfehlung für die Anpassung der Mindest- und Zielwerte der jeweiligen, artikelspezifischen Bestandsparameter abgeleitet, siehe Tabelle 6.10. Diese Gesamtempfehlung pro Artikel setzt sich dabei gleich gewichtet aus einer Empfehlung auf Basis des jeweiligen Bestandsniveaus bzw. der dazugehörigen Produktionsstabilität zusammen. Beispielsweise ergibt die Gesamtempfehlung für Artikel 17 aus Basis der Empfehlungen $R20 + L10 = R10$. Die Methode sieht vor, dass bei relativ bestandssicheren Artikeln, analysiert auf Basis von Vergangenheitsdaten, Bestand abgebaut bzw. reduziert wird und vice versa. Dieselbe Logik wird auch für das Kriterium Produktionsstabilität angewendet: Relativ produktionsstabile Artikel werden mit geringeren Ziellagerbestandsniveaus beplant, während weniger produktionsstabile Artikel mit einem Risikoaufschlag beplant werden. Bei beiden Kategorien wird das jeweilige Risiko in die Gruppen $Risiko = \{Gering, Mittel, Hoch\}$ eingeteilt. Die Zuordnung zu einer dieser Gruppen erfolgt über vordefinierte Schwellwerte. Außerdem werden rollierend die jährlichen Produktionsvorschauungen aktualisiert und in diesem Zusammenhang rollierend die artikelspezifische ABC- Zuordnung. Diese Empfehlungen werden mittels R (vgl. The R Foundation 2021) automatisiert erstellt und von der IT und Planungsabteilung im Zuge von Expert*innenbefragungen vor Ort validiert und anschließend manuell umgesetzt. Das Modul wird im Zuge der Entwicklung dieser Arbeit offline angebunden, eine Online-Integration (in die Java-Umgebung) wurde getestet und ist möglich (vgl. Kamhuber et al. 2019, S. 2114-2116).

Tab. 6.10: Empfehlungen auf Basis der statistischen Datenanalyse (vgl. Kamhuber et al. 2019, S. 2116)

Artikel	Smin	Smax	ABC	Produktions-Vorschau	ABC-Empfehlung	Bestandsrisiko-Evaluierung	Empfehlung (Bestandsniveau)	Empfehlung (Produktionsstabilität)	Gesamt-Empfehlung
1	10050	55000	A	447566	A	OK	K	K	K
2	2680	8000	A	67453	A	OK	K	K	K
3	6700	25000	A	341840	A	OK	K	R10	R10
4	700	4500	B	51425	B	OK	K	K	K
5	0	0	B	24617	B	Hohes Bestandsniveau, hohes Risiko	L20	K	L20
6	0	10000	A	331457	A	OK	K	K	K
7	2200	25000	A	440377	A	Hohes Bestandsniveau, hohes Risiko	L20	K	L20
8	7000	18000	A	103656	A	Hohes Bestandsniveau, hohes Risiko	L20	K	L20
9	18000	35000	A	340918	A	OK	K	R10	R10
10	3350	9000	A	75455	A	Hohes Bestandsniveau, mittleres Risiko	L10	K	L10
11	0	0	A	21363	C	Hohes Bestandsniveau, hohes Risiko	L20	K	L20
12	700	3000	C	21304	C	Hohes Bestandsniveau, mittleres Risiko	L10	K	L10
13	0	4000	A	41586	B	OK	K	L10	L10
14	15000	25000	A	130838	A	Geringes Bestandsniveau, hohes Risiko	R20	K	R20
15	7000	20000	A	47114	B	Hohes Bestandsniveau, mittleres Risiko	L10	K	L10
16	10050	35000	A	329122	A	OK	K	L10	L10
17	10050	25000	A	176292	A	Geringes Bestandsniveau, hohes Risiko	R20	L10	R10
18	4200	6000	A	151732	A	OK	K	K	K
19	5600	17000	A	269876	A	OK	K	R10	R10
20	0	2000	A	28800	B	Hohes Bestandsniveau, hohes Risiko	L20	K	L20

Legende: [K = Bestandsniveau halten] [L10/L20 = Bestandsniveau reduzieren um 10%/20%] [R10/R20 = Bestandsniveau erhöhen um 10%/20%]

6.5 Nutzen der Optimierungsmethode

Der Nutzen gestaltet sich, in Einklang mit der *Communication of Research* Richtlinie gemäß Design Science, multidimensional: Mit Hilfe des anfänglich statischen Planungstools wurde das neu zu errichtende Werk, begleitet von Simulation, erfolgreich auf Basis der neuen Produktionsablauflogik kostenminimierend geplant, dimensioniert und ausgelegt. Im Zuge der Inbetriebnahme des neuen Werksstandorts wurde das Planungswerkzeug dynamisch, sodass es rollierend (d.h. im konkreten Anwendungsfall wöchentlich) und automationsgestützt unter der Benützung von möglichst aktuellen Datenbankdaten hochwertige Vorschläge für die Planung generiert, welche im Anschluss manuell durch spezifische Informationen ergänzt und umgesetzt werden.

6.5.1 Bewertung der Kostenrelevanz der Optimierungsergebnisse:

Der Produktionsglättungseffekt auf Werksebene (Teilziel f_2) unter der Gegenüberstellung von GA (V1) und SA sowie der Basislösung ist dabei in Abb. 6.15 (Datensatz 3) zu sehen. Die Schwankungen auf Werksebene sind dabei in den Perioden mit den Belastungsspitzen um bis zu 35% in einzelnen Perioden reduziert worden, siehe z.B. die Perioden KW 31 – 33. Der GA übertrifft die beste Parametrisierung des SA und ist dabei auf Grund seines populationsbasierten Charakters robuster. Im Endausbau verfügt der GA-V2 zusätzlich noch über einen explorativen, problemspezifisch entwickelten Rekombinationsoperator. Der GA erweist sich, auf Grund der Ausnutzung von Holland's Schema-Theorem, als beste Wahl, um einerseits die Grundstruktur der bereits erreichten, heuristischen Lösung nicht zu stark zu verändern und dennoch lokale Minima unabhängig von der konkreten Parametrisierung vorzubeugen.

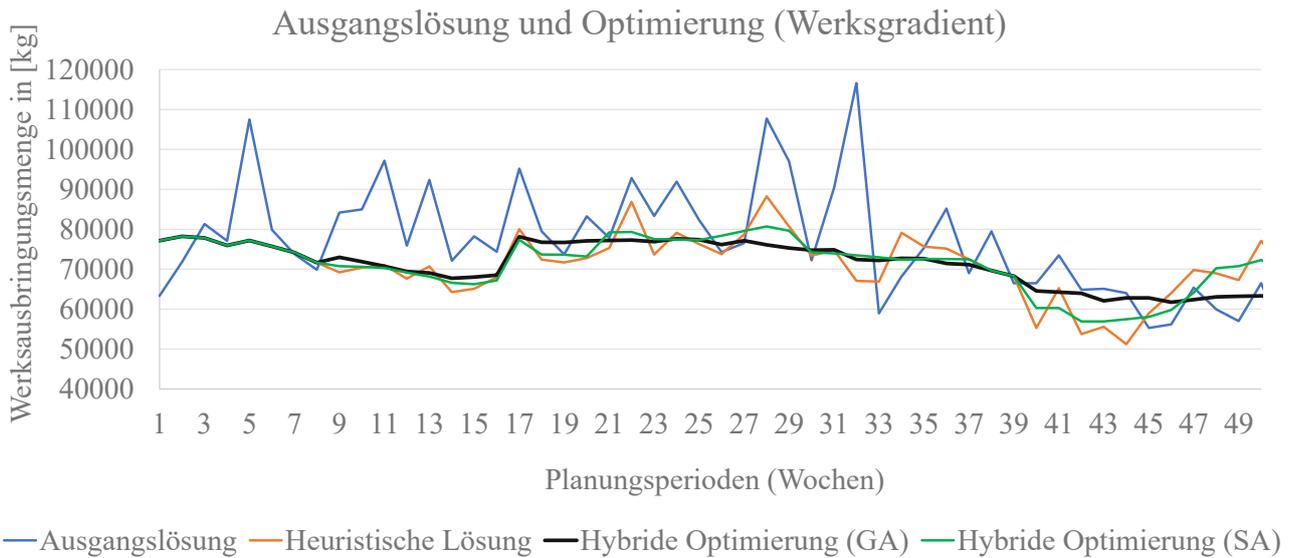


Abb. 6.15: Effekt der Produktionsglättung auf Werksebene (f_2) (vgl. Kamhuber et al. 2020, S. 15)

Die einzelnen Zielfunktionsbestandteile stellen mit Ausnahme von f_5 grundsätzlich Schwankungen (der Produktionsausbringungsrates, der Raumauslastung und im Bestandsverlauf) dar und spiegeln, in den jeweiligen Dimensionen einer möglichst gleichmäßigen Produktionsartikel- bzw. Werksdurchsatzmenge, Kapazitätsauslastung und Bestandsverlauf, ein robustes Produktionssystem mit vergleichmäßiger Produktionsausstoßrate. Die Kosteneinsparungen lassen sich im direkten Betrieb indirekt über die Reduktion von Überstundenschichten, der besseren Planbarkeit und Auslastung von Mensch und Maschine, sowie der höheren Leistungsfähigkeit des Produktionssystems im Verbund und der längeren Nutzungsdauer, bis die nächste Ausbaustufe notwendig wird, bewerten. Somit ergeben sich einmalige Einsparungen bei der Bauphase des Werksstandorts im zweistelligen Millionen-Bereich⁵ durch die Realisierung eines geglätteten Produktionssystems. Außerdem wird durch die Glättung eine längere Nutzungsdauer bis zum Erreichen der nächsten Ausbaustufe sichergestellt. Die Kosteneinsparungen in der Bauphase sind v.a. auf eine gleichmäßigere Nutzung und Auslastung der einzelnen Aggregate in der Grundproduktion (f_3) zurückzuführen, sodass die Kosteneinsparungen in diesem Bereich in Relation mit der Reduktion der Belastungsspitzen einhergehen.

Die größten Kostenreduktionspotentiale liefern im operativen Betrieb f_1 sowie f_2 . Die Minimierung von f_1 führt zu einem entsprechend ruhigeren Produktionsprozess auf Produktionsartikelebene sowie zu einer besseren Versorgungssicherheit durch die jeweiligen Lieferant*innen, welche die ursprünglichen Belastungsspitzen nun vergleichmäßigt anliefern. Dies führt auch zu einer entsprechenden Beruhigung im Wareneingangslager. Die Minimierung von f_1 leistet einerseits wesentliche Vorarbeit für die Minimierung von f_2 . Andererseits fokussiert sich die Minimierung von f_2 auf die Reduktion der Belastungsspitzen an der Engpassanlage im Werk und ist damit primär für die Reduktion von Überstundenschichten sowie die möglichst ökonomisch sinnvolle Nutzung von Perioden mit (ursprünglich) weniger Last verantwortlich. Die Reduktion von f_2 um bis zu ca. 80% (siehe Abb. 6.12 bzw. Tab. 6.9) resultiert in einer Reduktion der maximalen Belastungsspitzen (siehe Abb. 6.15) bzw. einer damit entsprechend einhergehenden Reduktion der Investitionskosten um ca. 35%. Das führt zu einer entsprechend anteiligen Reduktion der laufenden Betriebskosten im Vergleich mit der Ausgangslösung für den jeweils betrachteten Zeitraum. Dies begründet sich durch eine bessere Nutzung von Personal in Leerlaufzeiten bzw. eine Reduktion von entsprechenden Personalkosten (Mehrschichten, Überstunden) in den geringer ausfallenden Belastungsspitzen der optimalen Lösung.

⁵Unter Rücksichtnahme auf den*die industrielle*n Anwendungspartner*in wird auf eine konkrete Zahl an dieser Stelle verzichtet.

6.5.2 Simulation von Szenarien

Darüber hinaus werden mit dem Planungstool zukünftige (v.a. Mengen-) Szenarien in Form von Konfigurationsvarianten abgebildet: Dies umfasst beispielsweise das Durchspielen von Mengenszenarien auf Basis bestimmter Aktionen oder Leistungsveränderungen von Artikeln. Möglich sind auch veränderte Stammdaten wie z.B. modifizierte Bestandszielwerte bzw. veränderte Durchlaufzeiten für einen oder mehrere Prozesse im Arbeitsplan. Die Planungsabteilung ist durch die Methodik in der Lage, Szenarien mit veränderten Parametern für diverse Nebenbedingungen, v.a. die maximale Schichtkapazität bzw. das maximal erlaubte Verschiebefenster (auf Artikelebene) zu ändern und sich darauf basierend einen neuen Plan generieren zu lassen.

6.5.3 Anpassungsfähigkeit

Die Planungsabteilung ist durch die Empfehlungen der Datenanalyse befähigt, bestimmte Stammdaten jederzeit rasch zu ändern, wobei sich die Methodik unmittelbar an die geänderten Stammdaten anpasst. Die Änderung von Stammdaten umfasst z.B. die Änderung der maximal zulässigen Werksproduktionsmengen, Mindestbestand bzw. Sollbestand auf den Ebenen Artikel (statisch) und Produktionsperiode (statisch und dynamisch pro Periode bzw. Kalenderwoche änderbar), die Änderung der A/B/C-Zuordnung, die Änderung von Plandurchlaufzeiten bzw. die Änderung der zulässigen Verschiebefenster für Kurant- und Aktionsmengen pro Produktionsartikel. Die Methodik verarbeitet diese Änderungen selbständig und kommt dabei ohne Anpassung der Software aus.

Das unterstützende Datenmodul ist in der Lage bestimmte Parameter (A/B/C-Werte, Bestandsstammdaten) autonom zu überwachen und unterstützt die laufende Anpassung von Einflussgrößen für die Optimierung. Dieser Umstand ist methodisch insbesondere bei einer großen Artikelvielfalt wertvoll, die für die Planer*innen andernfalls nicht mehr überschaubar ist und schafft eine gewisse automatisierte Absicherung, wenn sie diese Veränderungen nicht selbst vorgenommen oder berücksichtigt haben. Dieses Lernen aus historischen Produktionsdaten führt in erster Instanz dazu, dass der Algorithmus adaptiv wird und ggf. eine systematische Aktualisierung von Stammdaten vorgenommen bzw. anschließend wieder in die Optimierung zurückgespielt wird.

6.6 Abgleich mit den Anforderungen an das Optimierungsverfahren

Im Zuge dieses Abschnitts werden die Anforderungen an das Verfahren aus Abschnitt 4.3 mit der entwickelten Methodik kritisch verglichen.

Die Kernanforderung aus Abschnitt 4.3 betrifft die entsprechende Modellierung des Zielfunktionssystems mit einem klaren Fokus auf eine Minimierung der Produktionsschwankungen. Die Modellierung der Zielfunktion mittels Gradienten, welche die schwankende Produktionsausstoßrate auf Artikel- und Werksebene als auch im Zuge der Kapazitätsauslastung sowie im Bestandsverlauf abbildet, erfüllt diese Anforderung akkurat und bestraft sämtliche Absolutabweichungen für alle betrachteten Dimensionen (Artikel- und Werksebene, Kapazitätsauslastungs- und Bestandsverlauf) im gesamten Planungshorizont.

Die Aktionsmengenglättung wird als erstes Modul vor der Kurantmengenglättung ausgeführt, da Experimente gezeigt haben, dass die Glättung der Aktionsmengen einen signifikant größeren Einfluss als die zeitliche Verteilung und Optimierung der Kurantmengen hat. Deshalb wird die Reihenfolge in der Heuristik entsprechend Alg. 5.1 ausgeführt. Das Verfahren verteilt die anfallenden Aktions- und Kurantmengen gleichermaßen systematisch gemäß Abb. 5.5 aus Abschnitt 5.6. Im Zuge der Glättung werden die ursprünglichen, wöchentlichen Produktionsauftragsmengen zeitlich vorverschoben als ggf. auch gemäß einer Heuristik aufgeteilt und entsprechend günstig in Bezug auf die Produktionsausstoßrate auf die einzelnen Perioden verteilt. Dabei verlängert sich bei jeder vorversetzten Produktionsmenge entsprechend die jeweilige Prozessdurchlaufzeit im Reifelager.

Die Methodik verfügt über eine gezielte Heuristik zur Glättung des mittelfristigen Bestandsniveaus auf Produktionsartikelebene und rundet dabei gezielt auf ganze Produktionsgebäude auf oder ab.

Die Methodik integriert eine dynamische Simulation zur akkuraten Entwicklung der Kapazitätsauslastung für alle notwendigen Aggregate und Produktionsbereiche innerhalb des betrachteten Produktionssystems, siehe Abbildung 5.2 aus Abschnitt 5.4.

In Bezug auf die metaheuristische Optimierung wurde sowohl ein SA als auch GA entsprechend *gleichberechtigt*⁶ implementiert und in Bezug auf Effizienz und erreichte Lösungsqualität verglichen. Der Vergleich der beiden Algorithmen, siehe auch Unterabschnitt 6.2.2, hat zu zwei zentralen Schlussfolgerungen geführt, welche insbesondere für rollierende Planungsverfahren bedeutend sind. Abschließend wurde das bessere Verfahren, der GA-V1, durch den ergänzenden Einsatz der Rekombination im GA-V2, weiter verbessert.

Die Besonderheit des Verfahrens liegt, neben der Berücksichtigung von statischen Nebenbedingungen, v.a. an der Berücksichtigung einer dynamischen Kapazitätsgrenze an der Engpassanlage im Zuge der Optimierung, siehe NB (5.2) aus Abschnitt 5.3 bzw. die nachfolgende Abbildung 6.16. Das Hauptziel der Produktionsglättung besteht darin, ein nahezu optimales Produktionsniveau möglichst ganzjährig zu erreichen, das sowohl die statischen als auch die dynamischen Kapazitätsbeschränkungen erfüllt, siehe Abb. 6.16: Der grüne Korridor um das ideale Produktionsniveau stellt die gewünschte Produktionsauslastung dar, die das Produktionssystem effizient verarbeiten kann, wobei sich die Flexibilitätskosten außerhalb dieses Korridors stark erhöhen. Statische Kapazitätsbeschränkungen in diesem Bereich stellen in der Regel die (langfristige) maximale Kapazität der Produktionsanlagen bzw. Produktionsräumlichkeiten dar, während typische dynamische Kapazitätsbeschränkung (un-) geplante Schwankungen der Personalkapazität darstellen. Das Kapazitätsplateau um die Wochen 35 – 41 in der Abbildung wird hauptsächlich durch die Vorbereitung auf die Weihnachtssaison induziert, dies stellt ein gemeinsames Merkmal der Lebensmittelproduktion in Europa dar.

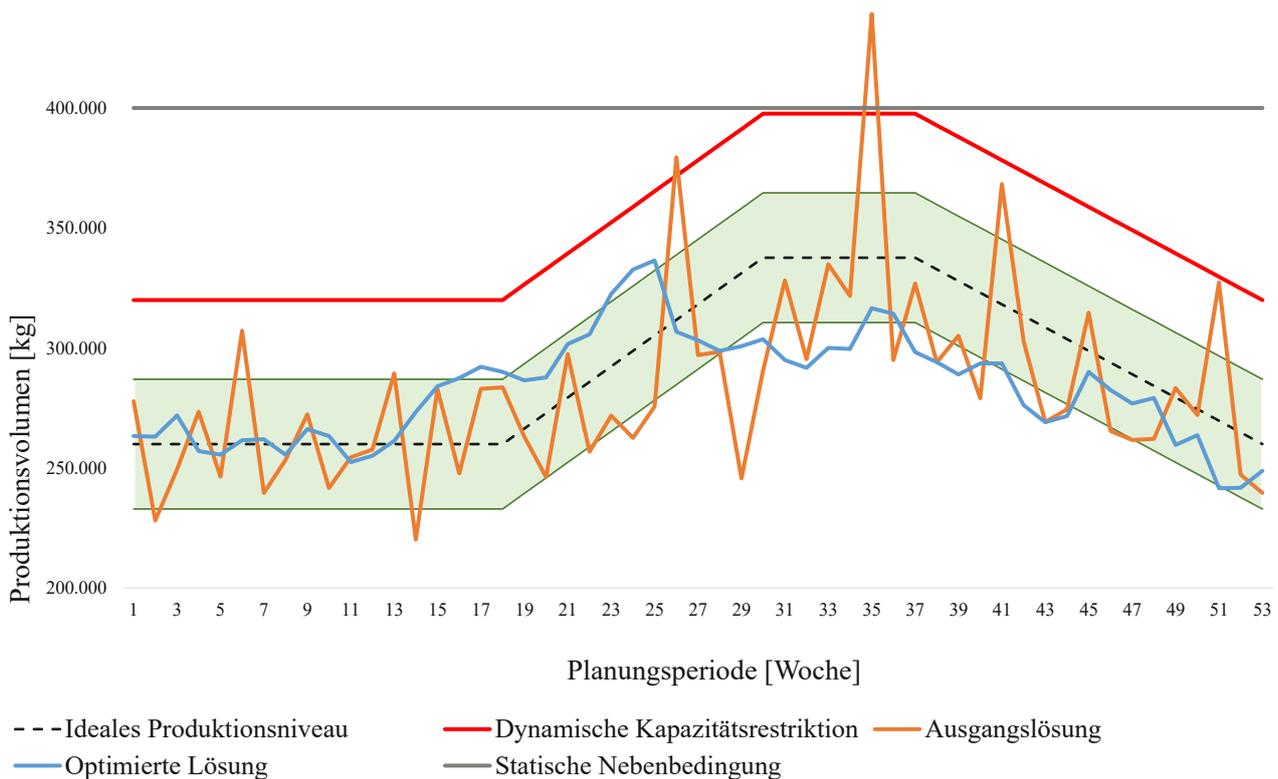


Abb. 6.16: Konzeptionelles Ergebnis: Optimierte Lösung versus Basislösung (vgl. Kamhuber et al. 2020, S. 2)

Eine weitere Besonderheit liegt in der theoretischen und praktischen Laufzeiteffizienz begründet, siehe Abschnitt 6.3. Die Planung vor Ort kann die einzelnen Nebenbedingungen, an die jeweilige Situation angepasst, (de)aktivieren. Dadurch bleibt die Planungshoheit in Bezug auf einen potentiellen Verstoß

⁶im Sinne der Operatoren und Funktionen

einer Nebenbedingung in einer Lösung erhalten bzw. wird durch einen neuen Lauf mit (manuell) modifizierten Rahmenbedingungen gelöst.

Die Optimierungsstudien zeigen die Laufzeiteffizienz auf Basis des Einsatzes von kleinen Populationsgrößen insbesondere für kurze und mittlere Laufzeiten, siehe Tab. 6.6. Diese Feststellung hängt auch mit der für den GA zur Verfügung gestellten Plateaulösung zusammen. Im Vergleich mit dem SA bietet der GA zwei markante Vorteile: Erstens verfügt der GA über den robusteren Lösungsfindungsprozess, der sich nicht nur auf eine Trajektorie verlässt. Zweitens sind die GA-Parameter (Populationsgröße, Iterationen etc.) und der verhältnismäßige Einsatz seiner Operatoren (Selektion, Mutation, Rekombination), siehe auch Tab. 6.7, im Zusammenhang mit rollierender Produktionsplanung wesentlich robuster und universeller einsetzbar in Bezug auf periodisch neue Datensätze, als dies beim SA mit optimierten Parametern der Fall ist. Der im Zuge der rechnergestützten Studie erstellte Abkühlungsplan bleibt, verglichen mit GA Parametern, sensibel in Bezug auf die Charakteristika eines spezifischen Datensatzes und ist daher in logischer Konsequenz für jeden Datensatz erneut anzupassen. Dies ist insbesondere für eine industrielle Anwendung ein entscheidender Nachteil, welche den Umgang mit sich stetig ändernden Datensätzen erfordert. Darüber hinaus bringt der GA grundsätzlich drei Operatoren mit sich, welche jeweils an das vorliegende Problem spezifisch angepasst werden können. Der implementierte GA-V2 verwendet alle drei zur Verfügung stehenden Operatoren in problemspezifischer Form.

Die Heuristik ermöglicht die Anwendung der Methodik auf große Planungshorizonte mit entsprechend großer Produktvielfalt, da der Rechenaufwand für die Optimierung annähernd *linear* und nicht, wie bei vielen Optimierungsverfahren, exponentiell ansteigt. Der lexikographische Ansatz im heuristischen Teil der Methode vermeidet die Nachteile von Zielfunktionsgewichtungen und erkennt, dass jedes Teilziel einen anderen Qualitätsaspekt innerhalb der Lösung und des zugehörigen Optimierungsprozesses bewertet (vgl. Freitas 2004, S. 78f.; Miettinen Kaisa 2001, S. 1-20). Dementsprechend wichtig ist es daher, bei der Übersetzung bzw. Transformation der besten in der Heuristik gefundenen Lösung die Gewichtungen mit IT und Planung kollaborativ zu definieren bzw. im Programm flexibel, also jederzeit veränderbar, zur Verfügung zu stellen.

6.7 Fazit

Die entwickelte Produktionsglättungsmethodik liefert zuverlässig und schnell hochwertige Produktionsvorschläge, welche der Planung bei ihrer Tätigkeit unterstützen. In Übereinstimmung mit der Design as a Search Process Richtlinie gemäß Design Science handelt es sich um einen iterative (meta-) heuristischen Verbesserungsprozess, welcher rasch eine satisfizierende Lösung findet. Die Metaheuristik hat sich als sinnvolle Ergänzung zur lexikographischen Heuristik erwiesen, weil erst durch die Metaheuristik alle Teilziele vollständig gegeneinander abgewogen werden, um eine global bessere Lösung zu finden. Die Heuristik selbst ist dabei effizient im Aufteilen und Splitten von Produktionsmengen, unterscheidend Aktions- und Kurantmengen, aus ungünstigen Perioden in günstigere Perioden, siehe Abschnitt 6.3, und stellt dabei bereits eine sehr gute *Plateaulösung* für die Metaheuristik zur Verfügung. Im Rahmen der Metaheuristik hat sich der gezielte Suchmechanismus als wirkungsvolles Instrument zur gezielten Verbesserung von einem von der Heuristik explizit vernachlässigten Teilziel erwiesen. Der Vergleich der beiden Metaheuristiken in Abschnitt 6.2 spricht ein klares Votum für den, verglichen mit einem SA, robusten GA aus. Der GA-V2 mit Rekombination übertrifft i.d.R. den GA-V1 ohne Rekombination. Dementsprechend empfiehlt sich die anteilmäßige Aktivierung dieses Operators gemäß Tab. 6.7.

Die Methodik wird mit zunehmender Systemlast *effektiver* und verfügt über ein annähernd lineares Laufzeitwachstum, in Abhängigkeit des Planungshorizonts bzw. der Produktionsartikelanzahl. Sämtliche an das Optimierungsverfahren gestellte Anforderungen werden, primär durch die Variabilisierung der Prozessdurchlaufzeit, erfüllt bzw. teilweise, durch die Integration einer dynamischen Kapazitätsnebenbedingung bzw. einer statistischen Datenanalyse, übererfüllt.

7

Kapitel 7

Conclusio und Ausblick

Die Ergebnisse der hybriden Optimierung des Zielfunktionssystems zeigen ein Produktionsglättungspotenzial in Bezug auf das betrachtete Produktionssystem von etwa 35 – 50% im Vergleich zum ursprünglich bereitgestellten Produktionsplan, siehe auch Tab. 6.9. In Bezug auf die direkten Kosten sind die Teilziele f_1 , f_2 sowie f_3 und f_5 direkt kostenrelevant. Das Gesamtkosteneinsparungspotenzial aus den Produktions- und Kapazitätsgradienten variiert je nach Datensatz, wobei sich im Einklang mit Unterabschnitt 6.5.1 konkret ein Kostenreduktionspotential von 35% ergibt. Dies führt zu entsprechend geringeren Investitionen für neue Produktionsanlagen, d.h. weniger Aggregate in den einzelnen Kapazitätsbereichen der Grundproduktion, was wiederum zu geringeren Investitions- sowie laufenden Betriebskosten für eine neue Fabrik sowie zu einer verbesserten Ressourcen- und Energieeffizienz führt. Die Gesamtreduktionskosten für die Errichtung des neuen Werksstandorts liegen im 2-stelligen Millionenbereich. Die laufenden Betriebskosten werden einerseits durch ausgeglichene, besser planbare Schichtzeiten und die verbesserte Nutzung von Leerlaufzeiten der Mitarbeiter*innen gesenkt. Andererseits resultiert eine bessere Auslastung der Aggregate insgesamt in einer entsprechenden Reduktion der laufenden Energiekosten (Belüftung, Klimatisierung) (vgl. Kamhuber et al. 2020, S. 14).

Diese Einsparungen werden durch vergleichsweise geringe Zusatzkosten für (die Investition und den Betrieb betreffend) den verlängerten Reifelagerdauerprozess beim Glätten erreicht. Die Zusatzkosten fallen entsprechend gering aus, weil nur ca. 10 – 15% der jährlichen Gesamtproduktionsmengen durch die Glättung eine verlängerte Reifelagerprozessdauer erhalten. Die höheren, einmaligen Investitions- und Betriebskosten für das Reifelager werden im Zuge der Zielfunktion für die rollierende Planung nicht explizit bewertet. Durchgeführte Simulationsstudien für das Produktionsartikelspektrum des Werks zeigen, dass das im Bau und Betrieb relativ kostengünstige Reifelager, in Einklang mit der gesamten Versatzproduktionsmenge (max. 15%, siehe Abschnitt 6.2 bzw. Tab. 6.1), um ca. 30% größer ausgelegt wird¹. Die Gleichmäßigkeit der Auslastung des Reifelagers wird dabei im Zuge der Optimierung innerhalb von f_3 mitbetrachtet. Einerseits bringt es im Zuge der Optimierung für die Zielsetzung keinen Nutzen, weil diese in diesem Bereich höheren Investitionskosten bereits angefallen sind. Außerdem gleichen sich die derart entstehenden Zusatzkosten durch eine gleichmäßigere Nutzung und Auslastung des Reifelagers wieder aus.

In den folgenden Abschnitten werden zunächst die Ergebnisse aus methodischer und konzeptioneller Sicht kritisch betrachtet. Anschließend werden die Forschungsergebnisse hinsichtlich der Beantwortung der Forschungsfragen diskutiert. Abschließend wird der weitere Forschungsbedarf in dieser Domäne festgehalten sowie ein finaler Ausblick gegeben.

¹Dieser Wert hängt konkret davon ab, mit welchen Artikeln und (Aktions-) Mengen bzw. maximal zulässigen Versatzperioden simuliert wird. Prinzipiell steigt dieser Wert überproportional zur maximal eingesetzten Versatzperiode bzw. den dazugehörigen (Aktions-) Versatzmengen.

7.1 Diskussion

Dieser Abschnitt diskutiert die entwickelte Methode anhand ihrer Ergebnisse sowie Vor- und Nachteile vergleichend mit dem Stand der Technik. Anschließend werden die Forschungsfragen beantwortet.

7.1.1 Kritische Betrachtung der Ergebnisse und Vergleich mit dem Stand der Technik

Der vorgestellte, hybride Optimierungsansatz integriert eine dynamische Simulation für eine möglichst realistische Berücksichtigung der Kapazitätsauslastungsentwicklung. Die problemspezifische Heuristik, die im Gegensatz zu bestehenden, rollierenden Planungsansätzen (vgl. Sampaio et al. 2017, S. 32) mit dynamischen Randbedingungen und einem variabilisierbaren Produktionsprozess saisonalen Effekten *gezielt* entgegenwirkt und den gesamten Planungshorizont ausnutzt, realisiert außerdem eine hohe Laufzeiteffizienz, die eine rollierende Planungsoptimierung begünstigt. Metaheuristische Auswertungen sind deutlich teurer (in Bezug auf die benötigte Laufzeit) im Vergleich zur lexikographisch umgesetzten Heuristik, die in der Lage ist, mehrere Optimierungsschritte gleichzeitig durchzuführen und dabei nur die Nebenbedingungen zu überprüfen, ohne den Fitnesswert stetig neu zu berechnen. Durch den Entfall der laufenden Fitnesswert-Berechnung entfällt die Ausführung des Simulationsmoduls in der Heuristik. Die Heuristik baut sich die Kapazitätsauslastungsentwicklung schrittweise bzw. sukzessive auf, indem die jeweils letzte Produktionsveränderung in die laufende Kapazitätsberechnung als Update integriert wird. Andererseits ist die Simulation dafür vorbereitet, mehrere Produktionsveränderungen auf einmal (d.h. während der Modifikation eines Individuums z.B. durch den GA) zu verarbeiten bzw. zu bewerten. Die Simulation verfügt über keinen logischen Bezug zur Kapazitätsberechnung der vorangegangenen Lösung, weil sich diese Berechnung für jedes Individuum vollständig von Grund auf neu berechnet. Die Integration der Ergebnisse der statistischen Datenanalyse ermöglicht die kontinuierliche Anpassung der Stammdaten in Bezug auf die mittelfristigen Lagerbestandsniveaus der Vergangenheit.

Das Verfahren übertrifft die im Stand der Technik (Kapitel 3) dargestellten und insbesondere in den Abschnitten 3.4 bis 3.6 sowie 3.9 näher diskutierten Verfahren durch die ermöglichte Variabilisierung zumindest eines Produktionsprozesses. Dadurch ist die Optimierung in der Lage, variable Produktionsdurchlaufzeiten im Zusammenspiel mit verschiedenen Arbeitsplänen auszunützen und Produkte entsprechend vorzuzuschieben, ohne dabei eine frühere Fertigstellung (bzw. entsprechende Lagerkosten von Fertigwarenbeständen) als Teil der Lösung zu generieren. Einige der in Kapitel 3 beschriebenen, überwiegend nicht-rollierenden, Verfahren verwenden das Konzept der Loseilung, eine Hybridisierung mit einer integrierten Simulation oder Machine Learning Methoden. Allerdings gibt es mindestens immer ein Merkmal, das von den vom aktuellen Stand der Technik verfügbaren Methoden nicht entsprechend adressiert bzw. berücksichtigt wird. Diese Arbeit verfolgt hingegen die Entwicklung eines hybriden adaptiven Optimierungsverfahrens für dynamische multikriterielle Produktionsglättung anhand eines **Multi-Objective Multi-Product Multi-Period (MOMPMP) capacitated production planning problem**,

- mit einer entsprechend ausformulierten und fokussierten, multikriteriellen Zielfunktion mit Fokus auf das Thema Produktionsglättung,
- das gleichzeitig die Problematik der Losgrößenplanung, siehe Abschnitt 3.6, auf Periodenebene berücksichtigt und integriert,
- und im Zuge der Heuristik die Produktionsschwankungen bzw. Losgrößen *gezielt* sowie *laufzeiteffizient* optimiert und anschließend die beste heuristische Lösung im Zuge einer globalen GA-Suche weiter verbessert,
- die Variabilisierung in Bezug auf die Durchlaufzeit des Produktionsprozesses im Zuge der gesamten Optimierung *gezielt* ausnutzt,
- die Integration eines Simulationsmodells zur Abbildung einer akkuraten Kapazitätsauslastungsvorschau auf Basis der jeweiligen Arbeitspläne berücksichtigt,

- und als rollierender Planungsansatz in Form eines *Digital Twin* umgesetzt sowie einer industriellen Anwendung implementiert wird, wobei die kontinuierlich gesammelten (Stamm-) Daten laufend überwacht und angepasst bzw. verbessert werden.

Gemäß der Kategorisierung von Kritzinger entspricht die Implementierung der in dieser Arbeit entwickelten Methodik der konzeptuellen Umsetzung eines *Digital Twins*: Dabei werden die Datenflüsse zwischen dem physischen Objekt (z.B. einer Fabrik) und dem digitalen Modell in beide Richtungen vollständig integriert und automatisiert. Im vorliegenden industriellen Anwendungsfall erfolgt einerseits die Datenübertragung über die eingerichtete Datenbankschnittstelle *automatisiert* in das Programm, welches die Optimierung inklusive Simulation und Datenauswertung anstoßt. Im Anschluss an einen rollierenden Planungslauf werden die Optimierungsergebnisse *automatisiert* über die Datenbankschnittstelle auf definierten Tabellen zurückgespielt. Der Produktionsplanung werden diese Ergebnisse aggregiert über eine interne GUI-Schnittstelle auf seinem Planungsprogramm *automatisch* angezeigt. Kritzinger hält in Bezug auf die Integrationsebenen fest, dass in der Literatur häufig von einem Digital Twin gesprochen wird, allerdings nur etwa 18% der Publikationen auch einen bidirektionalen Datenaustausch realisieren. Zumeist, d.h. in > 80% der analysierten Literatur, wird lediglich ein *Digital Model* mit bidirektional manuellem Datenaustausch oder ein *Digital Shadow* mit einem automatisierten Datenaustausch in lediglich einer Richtung realisiert, siehe auch Abb. 7.1 (vgl. Kritzinger et al. 2018, S. 1016-1022).

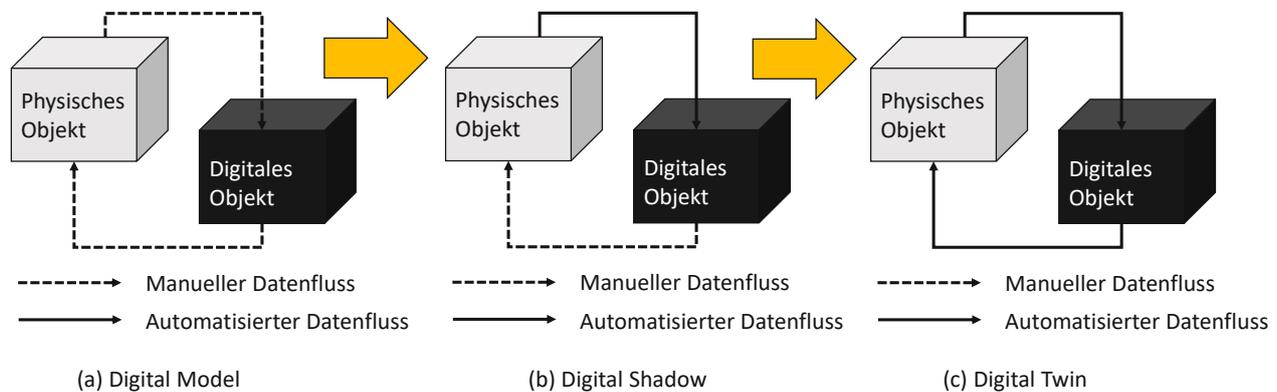


Abb. 7.1: Integrationsebenen vom Digital Model bis zum Digital Twin (Quelle: eigene Darstellung, in Anlehnung an: Kritzinger et al. 2018, S. 1017)

Als nachteilig wird dem in dieser Arbeit entwickelten Digital Twin angehaftet, dass der vollständig automatisierte Datenaustausch zu Misstrauen bei der Planungsabteilung führt, wenn z.B. fehlerhafte Datensätze (redundante Planungsdatensätze, fehlende Aktionsmengenobjekte, fehlerhafte Aktualisierung von Stammdaten) eingespielt werden. Der Digital Twin prüft die Daten hinsichtlich deren Konsistenz (richtiger Datentyp, Planungshorizont, etc.), ist aber mangels Wissens nicht in der Lage, zu überprüfen, ob bestimmte Planungsmengen in einer Woche bei einem definierten Artikel fehlen oder bewusst redundant auftreten. Eine entsprechende Fehlersuche gestaltet sich trotz vollständiger Transparenz im Programm mühsam, weil Daten ggf. fehlerhaft übergeben werden oder die Lösung in der Planungsmaske fehlerhaft aufbereitet bzw. angezeigt wird. Dies resultiert in einem immanenten (Planungs-) Risiko, welches mit der Verwendung der Methodik einhergeht.

Diese Arbeit zeigt, dass sich die Kopplung von Simulation und (hybrider) Optimierung methodisch sinnvoll und effizient gestalten lässt, wenn sich die Simulation auf einen bestimmten (relevanten) Ausschnitt des Produktionssystems beschränkt. Außerdem reduziert die in dieser Arbeit entwickelte Optimierungsmethode systematisch die Produktionsausstoßrate unter der Berücksichtigung von dynamischen Kapazitätsnebenbedingungen sowie der Vorgabe von dynamisch-saisonalen Zielbestandsniveaus.

Ein konzeptionelles Alleinstellungsmerkmal des entwickelten Verfahrens ist die Variabilisierung der Durchlaufzeit von einem der betrachteten Grundproduktionsprozesse. Diese Prozessvariabilisierung ermöglicht, im Zusammenspiel mit dem methodischen Kern der Optimierungsheuristik (dargestellt in Abb.

5.5), die Freisetzung eines entsprechend großen Optimierungspotentials im Rahmen der entwickelten Methodik und entlastet dadurch das gesamte Produktionssystem. Als Nachteil ist im Zusammenhang mit der Methodenentwicklung auch der Aufwand (Kosten, Zeitspanne) für die Modellierung und Entwicklung sowie die Implementierung und laufende Pflege zu bewerten.

Letztlich entscheidet nicht das konkrete Optimierungsergebnis, sondern die Nutzung einer Methodik über den Erfolg der jeweiligen Anwendung. Im Zuge der Lösungsfindung für ein definiertes Problem ist für das Management daher nicht nur die Qualität in Bezug auf das jeweilige Ergebnis entscheidend, sondern v.a. auch die Transparenz des Verfahrens bzw. die Gewährleistung der Rückverfolgbarkeit in Bezug auf die Zielerreichung. Die simulationsgestützte Optimierungsmethodik ist *objektorientiert* und *konstruktiv* aufgebaut, sodass sich jede Änderung im Lösungsobjekt zur Laufzeit transparent beobachten und verfolgen und auch im Nachhinein über die Dokumentation sämtlicher Änderungen programmatisch rückverfolgen lässt. Das ist einer der großen Unterschiede, den eine spezifisch zugeschnittene Methode in Form einer spezifisch entwickelten Anwendung liefert im Vergleich mit einem (generischen) Solver, welcher ggf. nicht flexibel an die real vorliegenden Spezifika anpassbar ist und keine Integration von z.B. Simulation oder statistischer Datenanalyse in die bestehende Methodik erlaubt.

7.1.2 Beantwortung der Forschungsfragen

Im Zuge dieses Abschnitts werden die in Abschnitt 1.5 behandelten Forschungsfragen dieser Arbeit diskutiert und beantwortet.

Forschungsfrage 1: „Wie gestaltet sich eine effizient implementierte, dynamische, multikriterielle Produktionsglättungsmethode, welche die Schwankungen der Produktionsausstoßrate effektiv minimiert?“ besteht aus sechs Unterfragen ($a - f$), die in diesem Abschnitt beantwortet werden:

Ad 1a: Die optimierende Zielfunktion unterscheidet sich in ihrer Ausprägung einerseits in Form von *Absolutdifferenzen* der jeweiligen Variablen deutlich von üblichen Zielfunktionen (vgl. Sihn et al. 2018, S. 450; Karimi-Nasab und Konstantaras 2012, S. 481), siehe Abschnitt 5.3 bzw. Glg. (5.1). Während eine Absolutdifferenz keine direkte Kostenfunktion darstellt, bedeuten diese Absolutabweichungen Schwankungen von zeitlich benachbarten Produktionsartikelmengen (desselben Artikels), welche in der vom Zielfunktionssystem aufgespannten Gesamtzielfunktion minimiert werden. In dieser Arbeit werden diese Absolutdifferenzen gemäß den Teilzielfunktionsgleichungen (5.5) - (5.8) für vier Teilzielfunktionen verwendet, um die entsprechenden (Produktions-, Kapazitätsverlaufs-, und Bestands-) Schwankungen zu reduzieren.

Auf Grund der Logik der Optimierung fehlen bestimmte Zielfunktionsbestandteile bzw. für eine Optimierung typische Kriterien wie Liefertermintreue oder Durchlaufzeit. Die Bewertung der Liefertermintreue in einer mittel- bis langfristigen Produktionsplanung ist einerseits aus Gründen der Genauigkeit fragwürdig. Andererseits verbessert die Methodik prinzipiell die Liefertermintreue, wenngleich dieses Kriterium im Rahmen dieser Arbeit sinnvollerweise nicht bewertet wird: Durch die Vorverschiebung von Produktionsmengen in frühere Perioden verbessert sich die Liefertermintreue zwangsläufig, wobei davon ausgegangen wird, dass die Liefertermintreue bereits bei der Initiallösung erfüllt ist. Die Bewertung der Durchlaufzeit ist grundsätzlich möglich, allerdings nicht förderlich für die (meta-) heuristische Suche, weil längere Durchlaufzeiten auf Grund der Logik der Methodik teilweise² gewünscht sind. Durch die Fristigkeit der Methodik werden Rüstkosten nicht exakt ermittelt, sondern lediglich abgeschätzt. Der Erfolg des GA lässt sich v.a. durch den zielgerichteten Mutations-Operator begründen, der von einem stochastisch-explorativeren Rekombinationsoperator (im richtigen Verhältnis, siehe Abb. 6.9) im Zuge der metaheuristischen Suche begleitet wird.

Ad 1b: Eine dynamische Methode berücksichtigt, wie in dieser Arbeit entwickelt, einerseits im Laufe der Zeit veränderliche Randbedingungen (dynamisch beschränkte Kapazitäten, dynamisch vorgegebene Zielbestandsniveaus) als auch eine entsprechend integrierte Ablaufsimulation betreffend die Kapazitätsauslastungsvorschau. Die dynamischen Kapazitätsvorgaben bzw. Zielbestandsniveaus sind für die rollierende Ausführung der Methodik essentiell. Eine statische Methode verzichtet auf diese

²und zwar genau dann, wenn in einem bestimmten Umfang erforderlich

dynamischen Bestandteile. Die entwickelte Methodik in dieser Arbeit ist auch in der Lage, die Stammdaten dynamisch, also im Zeitverlauf kontinuierlich, anzupassen. Grundsätzlich unterscheidet sich eine dynamische von einer statischen Methodik dadurch, dass sich Parameter bzw. Variablen im Laufe der Optimierung ändern. Das ist bereits während einem einzigen Lauf als auch über mehrere Läufe bzw. Perioden hinweg möglich.

Ad 1c: Der multidimensionale Nutzen der Methodik wird ausführlich in Abschnitt 6.5 diskutiert. Die Schwankungen der Produktionsausstoßrate auf Werksebene wurden in den Perioden mit den Belastungsspitzen um bis zu 35% reduziert. Dadurch haben sich entsprechende Kosteneinsparungen im 2-stelligen Millionenbereich beim Bau der zusätzlichen Fabrik ergeben. Außerdem ermöglicht die Methodik eine bessere Planbarkeit und Auslastung der zur Verfügung stehenden Personal- und Kapazitätsressourcen. Durch die bessere Ressourcenauslastung ergibt sich auch eine nachhaltigere Nutzung der zur Verfügung stehenden Werkskapazitäten. Weitere Nutzendimensionen stellen die (Entscheidungs-) Unterstützung der rollierenden Planung im laufenden operativen Betrieb sowie die Möglichkeit der Zurverfügungstellung und Bewertung bzw. Optimierung von verschiedenen (Absatz- und Produktions-) Szenarien und das laufende Monitoring von Bestandsstammdaten dar.

Ad 1d: Die Landau-Notation, siehe 2.3.1.4, erlaubt grundsätzlich die theoretische Bewertung der Laufzeitkomplexität des Verfahrens. In Abschnitt 6.3 validiert die tatsächliche die theoretisch ermittelte Laufzeitkomplexität. Die Ergebnisse decken sich in den einzelnen Bestandteilen der gesamten Laufzeit mit einer Abweichung von wenigen Prozentpunkten über einen Planungshorizont von (bis zu) 156 Wochen. Die durchgeführte Validierung bestätigt, dass die Laufzeit der entwickelten Methodik mit annähernd linearer Laufzeit (mit Ausnahme der zunehmenden Laufzeit für eine erfolgreiche Mutation bzw. Rekombination) zunimmt. Dies ist insofern günstig, weil die Methodik rollierend ausgeführt wird und damit auf einen längeren Planungshorizont bzw. mehr Produktionsartikel (z.B. für ein anderes Produktionssystem) *generalisiert* übertragbar und anwendbar ist.

Ad 1e: Die Ergebnisse der statischen Methodik (vgl. Sobottka et al., 2017, S. 3495-3506) unterscheiden sich von den Ergebnissen der rollierenden Methodik (vgl. Kamhuber et al., S. 2108-2118; Kamhuber et al. 2020, S.1-16) sowohl in der Verwendung als auch in den Charakteristika der Planung. Die Verwendung der statischen Methode stellt aus konzeptioneller Sicht eine *einmalige* Planungsgrundlage für die Dimensionierung und Auslegung der notwendigen Produktionsanlagen (siehe Abb. 4.1) dar. Die rollierende Planungsmethodik berücksichtigt nach dem Bau zusätzlich den jeweils aktuellen Absatzplan sowie die pro Artikel dazugehörigen aktuellen Stammdaten, Bestandsinformationen in der Grundproduktion als auch im Fertigwarenlager sowie die jeweiligen Zielbestandsniveaus und hat eine andere Aufgabe: Diese besteht primär in der *mittelfristig rollierenden* Bereitstellung von entsprechenden Wochenproduktionsplänen für die (Verarbeitung in der) Grundproduktion bzw. den Frischfleisch-Einkauf, siehe Abb. 5.3.

Ad 1f: Die Hybridisierung mit Machine Learning- und Data Mining-Methoden wird in der Taxonomie von Talbi in Unterabschnitt 3.4.4 andiskutiert (vgl. Talbi, S. 171-215). Der Einsatz der jeweiligen Methode hängt einerseits von der Aufgabe als auch von den zur Verfügung stehenden Daten ab. Im Falle der vorliegenden Arbeit hat sich statistisches Lernen als die sinnvollste Ergänzung herausgestellt, da die zur Verfügung stehenden Daten v.a. aus historischen Bestandsdaten, historisch realisierten Produktionsmengen sowie den jeweiligen Produktionsvorschaumengen bestehen. Im Falle des Vorliegens von z.B. Daten betreffend die automatisierte Fahrzeugsteuerung oder Raumklimatisierung ist es grundsätzlich auch möglich, die Fahrzeug-Routen in Abhängigkeit der Auslastung oder Artikelgruppen bzw. die Produktqualität in Abhängigkeit der jeweiligen Raumtemperatur zu überwachen bzw. zu *lernen*.

Forschungsfrage 2: „*Welche Anforderungen und Voraussetzungen sind für den erfolgreichen Einsatz der in dieser Dissertation entwickelten Methodik für multikriterielle Produktionsglättung in der fleischverarbeitenden Industrie erforderlich? Welche Eigenschaften und Merkmale sind für eine industrielle Anwendung in generalisierter Form zu erfüllen, damit diese Methodik sinnvoll anwendbar ist?*“

Ad 2: Die Variabilisierbarkeit bezüglich der Durchlaufzeit von zumindest einem betrachteten (Teil-)

Produktionsprozess mit relevanten Produktionsmengenanteilen³ muss innerhalb des Produktionssystems grundsätzlich gegeben sein und sich als ökonomisch sinnvoll erweisen. Im vorliegenden industriellen Anwendungsfall genügt es, wenn max. 15% der jährlichen Gesamtproduktionsmenge im Mittel um ca. 5–20% in Bezug auf die ursprüngliche Durchlaufzeit vorversetzt und damit in der Produktion verlängert werden. Die konkreten Prozentsätze hängen jedoch vom Anwendungsfall sowie damit verbunden, den absoluten und relativen Schwankungen des Produktionssystems pro Periode (Stunde, Tag, Woche, Monat, etc.), den damit verbundenen Zusatzkosten, saisonalen Effekten sowie dem Marketing- und Vertriebsgeschäft ab.

Prinzipiell hat die Methodik, verglichen z.B. mit einem Konzern, einen größeren Effekt auf ein KMU, weil die absoluten und relativen Schwankungen dort auf Werksebene größer ausfallen. Dies liegt u.a. darin begründet, dass ein Lieferwerk aus einem Konzern i.d.R. mehrere Absatzmärkte bedient, während ein KMU häufig nur einen Markt bzw. wenige Absatzmärkte beliefert.

In Bezug auf die Granularität, also der Auflösung des Zeithorizonts (Stunde, Tag, Woche, etc.) im Rahmen der Methodik, gilt die Anforderung, dass die Stammdaten in der vom Anwendungsfall geforderten Granularität vollständig zur Verfügung stehen, um in Abhängigkeit des betrachteten Teilziels eine realistische Aussage über die (zu erwartende) Kapazitätsauslastung zu treffen. Für eine entsprechende Glättungsfunktionalität im kurzfristigen Planungshorizont, also in der Feinplanung, ist z.B. die Entwicklung einer entsprechend detaillierten Simulation unter Einbezug der Anliefer- und Einlagerungszeitfenster (auf Stunden- und Tagesbasis) sowie der Produktions- und Schichtpläne inkl. der Berücksichtigung von Ausfallszeiten und Rüstzeiten sowie diversen weiteren detaillierten Informationen aus dem Produktionssystem (Ausschuss etc.) erforderlich. Außerdem stehen all diese Informationen in entsprechender Genauigkeit für die Methodik aktuell abgreifbar, z.B. auf einer Datenbank, bereits zur Verfügung, damit diese instantan für die weitere Verarbeitung bereit sind.

Diese Arbeit leistet einen entsprechenden Beitrag für den Anwendungsbereich der industriellen Produktionsglättung durch die Entwicklung einer über die Implementierung hinausgehende Hyper-Heuristik (siehe auch Alg. 5.1), die mit relativ wenig Aufwand an ein neues Problem anpassbar ist (vgl. Blum et al. 2011, S. 4137).

7.2 Ausblick

Der Ausblick lässt sich konkret anhand von möglichen, teilweise bereits geplanten, Erweiterungen darstellen sowie zudem in den weiteren grundsätzlichen Forschungsbedarf gliedern.

7.2.1 Erweiterungen der Methodik

Der Ausblick auf mögliche Erweiterungen beinhaltet die Optimierung der nachgelagerten Verpackungsprozesse, um auch den Kommissionier- und Verpackungsbereich im betrachteten Werk methodisch entsprechend zu glätten. Darüber hinaus wird die Methodik aktuell auf die konkrete Verwendung in einem weiteren Werk vorbereitet. Es sind bereits Vorbereitungen⁴ getätigt, sodass auch weitere Werke an die Methode anschließbar sind.

Außerdem ist es angedacht, im Zuge der Kapazitätsauslastung auch eine intelligente Logik für die automatische Befüllung und Entleerung der einzelnen Raumkapazitäten zu integrieren. Dies stellt ein unabhängig zu lösendes Teilproblem in Form einer umfangreichen Ergänzung dar und besteht darin, dass die einzelnen Gänge in den Räumen tagesaktuell derart mit Stellplätzen befüllt werden, sodass bei einer Auslagerung möglichst wenig Paletten umgeschichtet und idealerweise die zuletzt befüllten Plätze in den Gängen wieder freigegeben werden.

Um die Genauigkeit und Voraussetzung der Methodik für den realen Betrieb weiter zu erhöhen, bietet es sich an, in Zukunft ein (webbasiertes) Lieferantenportal in die Methode zu integrieren.

³d.h. für einen relevanten Anteil an der jährlichen Gesamtproduktionsmenge

⁴Beispielsweise wurde auf Datenbanktabellen bereits die Spalte Lieferwerk eingefügt.

Dieses soll jeweils aktuelle Lieferant*inneninformationen bereitstellen und diese vorausschauend in der Planung berücksichtigen. Gegenwärtig erhält die Einkaufsabteilung die als Input für die Optimierung benötigten Informationen manuell und in längeren Intervallen, sodass ein Teil des Inputs für die Optimierung (z.B. betreffend des Mengengerüsts) ggf. Änderungen unterworfen ist, die dem Planungs- und Produktionssystem derzeit noch nicht bekannt sind und somit die Planungsgenauigkeit potenziell beeinträchtigen.

Eine mögliche Verbesserung der Zielfunktion berücksichtigt beispielsweise, dass weiter in der Zukunft liegende Perioden weniger Einfluss auf die gesamte Fitness als näher zu berücksichtigende Perioden haben. Dazu benötigt man jedoch ein entsprechendes Einverständnis von Planung und IT sowie eine Definition der einzelnen Planungsabschnitte (Kurzfristplanung, Mittel- und Langfristplanung) im Zeitverlauf. Das Ziel kann z.B. in einer stärkeren Gewichtung von nahe in der Zukunft liegenden Planungsperioden liegen, weil diese bereits unmittelbar relevant für die anstehende Produktion sind. Außerdem ist eine Erweiterung der Zielfunktion bzw. eine Veränderung der Gewichtungen denkbar und möglich.

Aktuell werden im Zuge der Methodik bereits mehrere Planungsszenarien übergeben und automatisiert optimiert sowie zurückgegeben. Dabei erfolgt von der Methodik keine Präferenz für oder gegen eine bestimmte Planungsvariante. Diese erfolgt z.B. im Zuge einer vergleichenden Potentialermittlung, sodass die jeweils besten gefundenen Lösungen gespeichert und anschließend manuell verglichen werden.

In Bezug auf die Laufzeit lässt sich v.a. für umfangreiche Optimierungsläufe eine Logik betreffend die Zielfunktion ergänzen, welche jedes Lösungsobjekt (bzw. die entsprechenden Lösungsmatrizen) und die dazugehörige Zielfunktionsauswertung global abspeichert und vor jeder Bewertung überprüft, ob für diese Lösung bereits eine Bewertung vorliegt. Die Ersparnis in Bezug auf die Laufzeit bedeutet für den vorliegenden Anwendungsfall den Entfall der Simulation sowie der Zielfunktionsauswertung, siehe auch Abschnitt 6.3, und beträgt ca. 25 – 30% für jedes modifizierte Individuum unter der Bedingung, dass bereits eine Bewertung für exakt diese Lösung vorliegt. Eine weitere Möglichkeit zur Reduktion der Laufzeit besteht auch in der Parallelisierung von bestimmten Teilen, z.B. Operatoren, des GA.

Im Rahmen der Weiterentwicklung des GA-V2 besteht die Möglichkeit, diverse, weitere Ausprägungen der drei GA-Operatoren umzusetzen. So bietet sich ggf. eine Erweiterung des Rekombinations-Operators (der Austausch von größeren Gen-Sequenzen, z.B. mehreren, nebeneinanderliegenden Perioden) oder auch ein anderer Selektions-Operator (z.B. rangbasierte Selektion, Turnierselektion, etc.) an. Andererseits ist der in GA-V2 implementierte Rekombinationsoperator bereits in der Lage, dieselben Lösungen, welche sich auch mit einem größeren Operator für die Rekombination umsetzen lassen, herzustellen. Dabei wird aktuell jede Veränderung am Individuum bewertet, ohne größere Veränderungen auf einmal vorzunehmen und eine Zwischenlösung zu überspringen. Dies hängt auch von der Problemstellung sowie dem Einsatz des GA, z.B. als alleiniges oder ergänzendes Suchverfahren ab. Des Weiteren ist es ggf. sinnvoll, den Einsatz von Mutation und Rekombination oder die Populationsgröße im Optimierungsverlauf variabel bzw. dynamisch zu gestalten.

Prinzipiell sind sowohl GA als auch SA für die Optimierung von Kurant- und Aktionsmengen geeignet, wobei diese im konkret betrachteten Anwendungsfall nur als ergänzende Verfahren für die weitere Optimierung der laufend anfallenden Kurantmengen vorgesehen sind. Die eher punktuell anfallenden Aktionsmengen werden also nur durch die regelbasierte Produktionsglättungsheuristik entsprechend der in Unterabschnitt 5.6.1 beschriebenen Logik, siehe auch Abb. 5.5, vorversetzt. Der Nutzen einer weiteren Optimierung der Aktionsmengen (mittels GA bzw. SA) ist dementsprechend im konkreten Anwendungsfall zu hinterfragen und mündet wohl in einem Abtausch von f_1 gegenüber f_2 . Im Zuge einer Vereinbarung mit dem*der industriellen Anwendungspartner*in besteht die Möglichkeit, diesen Freiheitsgrad für die Optimierung zukünftig freizuschalten.

Es besteht prinzipiell die Möglichkeit, die Auflösung der in dieser Arbeit anhand des industriellen Anwendungsfalles umgesetzten Produktionsglättungsmethodik von einer Wochenbetrachtung auf eine Tagesbetrachtung zu verfeinern. Dafür werden v.a. tägliche Rückmeldedaten von Produktionsanlagen und entsprechende Anliefermengen im Tages- und Zeitverlauf benötigt. Mit einer entsprechend detaillierten Simulation ist es in Kombination mit einer tagesgerechten Auflösung der Wochenproduktionsmengen und Arbeitsplandaten möglich, die Methodik entsprechend zu verfeinern. Dies entspricht

nicht der Zielsetzung dieser Arbeit. Der Beitrag von Ricondo stellt ein entsprechendes Framework für einen derart zu konzipierenden digitalen Zwilling zur Verfügung (vgl. Ricondo et al. 2021, S. 762-767).

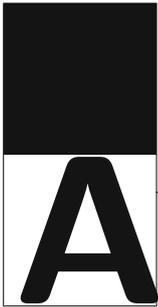
7.2.2 Weiterer Forschungsbedarf

Der Beitrag von Sihm zeigt, dass es möglich ist, für ein komplettes Werk im Bereich der Lebensmittelproduktion eine hybride Simulation zu entwickeln und in die Abläufe methodisch zu integrieren. Als Ergänzung zu der vorgestellten Feinplanungsmethodik ergänzt sich als Kombination eine optimierte Mittel- und Langfristplanung, wie sie im Zuge dieser Arbeit entwickelt worden ist. Dabei besteht jedoch die Herausforderung, dass die beiden Planungshorizonte, wie in 7.2.1 festgestellt, unterschiedliche Planungsziele haben und sich in der kombinierten Ausführungsreihenfolge entsprechend austauschen müssen (vgl. Sihm et al. 2018, S. 447ff.).

Weiterer Forschungsbedarf besteht darüber hinaus auch in der werksübergreifenden Glättung von mehreren Produktionssystemen, z.B. in Bezug auf einen kompletten Werksverbund. In diesem Rahmen werden zu den werkspezifischen Kostensätzen (für Anlagen und Personal) auch die jeweiligen Distributionskosten zwischen den Werken (als ggf. auch zu den Kund*innen) mitbetrachtet. Damit erweitert sich das bisherige Problem um ein Problem aus der Graphentheorie mit entsprechenden Kosten für die einzelnen Knoten und Kanten.

Erweiterungen in Bezug auf den Optimierungsalgorithmus können, je nach Anwendungsfall, sowohl methodisch in der Form erfolgen, dass bestimmte Teilziele exakt gelöst werden: Caumond löst z.B. ein kombiniertes Reihenfolge- und Routingproblem, wobei die Reihenfolge mittels Branch-and-Bound (für kleine und mittelgroße Instanzen) exakt gelöst wird (vgl. Caumond et al. 2009, S. 706-722). Eine Möglichkeit zur Beschleunigung des Optimierungsprozesses zeigt z.B. Sobottka am Beispiele des vollständigen Ersatzes der Optimierungslogik durch künstliche neuronale Netze auf und stellt fest, dass die erzielte Genauigkeit dieser Ersatzmodelle für kleine Simulationsszenarien hinreichend genau ist, allerdings mit zunehmender Auftragskomplexität sowie Produktionssystemgröße signifikant abnimmt. Dabei gestaltet sich die Erzeugung von genügend verlässlichen Trainingsdaten als Herausforderung. Schließlich hält er fest, dass es sich für Ersatzmodelle, welche ausschließlich durch Simulationsfeedback antrainiert werden, aktuell herausfordernd darstellt, das komplette Verhalten eines (komplexen) Produktionssystems möglichst vollständig abzubilden (vgl. Sobottka et al. 2019, S. 1844-1853).

Obwohl die hybride Optimierungsmethode auf der Grundlage einer konkreten, industriellen Fallstudie aus der Lebensmittelproduktion entwickelt worden ist, lässt sich die Methode auf ein breites Spektrum von Planungsanwendungen in der Industrie generalisieren. Das Problem der schwankenden Nachfrage, das zu Ineffizienzen in der Produktion führt, ist bei fast jedem Hersteller vorhanden, ebenso wie die Existenz teurer und knapper (Engpass-) Produktionsanlagen innerhalb der Prozesskette. Die Hauptvoraussetzung für die Anwendung der entwickelten Methodik besteht darin, einen Prozess hinsichtlich seiner Prozessdurchlaufzeit derart variabel zu gestalten, sodass dieser für die Optimierung ökonomisch effizient ausgenutzt wird. Dies bedeutet ggf. ein größeres Zwischenlager, veränderte Prozesszeiten für (alternative) kostengünstigere Produktionsanlagen oder bestimmte Entscheidungen in der Logistik und der Lieferkette betreffen. Im konkreten Anwendungsfall stellt dies ein verhältnismäßig großer dimensioniertes Reifelager am Ende der betrachteten Prozesskette dar. Weiterer Forschungsbedarf besteht damit v.a. in der Identifikation von Möglichkeiten zur Generalisierung der Methodik bzw. entsprechenden Maßnahmen zur Prozessflexibilisierung, z.B. in Form von wandlungsfähigen Produktionszellen, damit die im Rahmen dieser Arbeit entwickelte Methodik entsprechend angepasst einsetzbar ist.



Anhang A

Anhang

A.1 Grundlegende Begriffe aus der Vektoranalysis

Dieser Abschnitt definiert die Begriffe und Symbole des Skalar- und Vektorfeldes sowie des Gradienten- und Nabla-Operators (vgl. Breunig 1995, S. 2):

Skalar- und Vektorfeld: Unter einem Feld wird ein Raum verstanden, in dem jedem Punkt $P(x; y; z)$ eine physikalische Größe zugeordnet ist. Im Falle einer ungerichteten Größe (*Skalar*) spricht man von einem Skalarfeld. Beispiele dafür liefern Temperatur, Dichte und Druck. Im Falle einer gerichteten Größe (*Vektor*) spricht man von einem Vektorfeld. Beispiele hierfür liefern die Gravitation, eine Strömung oder ein elektrisches Feld.

Nabla-Operator: Der Nabla-Operator ist ein formaler Vektor, dessen Komponenten aus den partiellen Differential-Operatoren bestehen.

$$\text{Man definiert: } \nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}$$

Gradient: Auf Basis eines Skalarfeldes $f(x, y, z)$ ist der Gradient von f der Vektor, der in die Richtung der größten Änderung von f im Punkt $P(x; y; z)$ zeigt und dessen Betrag gleich dieser größten Änderung entspricht.

$$\text{Man definiert: } \text{grad } f = \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix} = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}$$

In einem Potentialfeld zeigt der Gradient immer in die Richtung, in welche sich ein Partikel bewegen würde, wenn es an der Stelle P in das Feld gebracht wird.

A.2 Numerische gradientenbasierte Verfahren

Der Anhang erläutert im Zuge der Diskussion der numerischen Verfahren einerseits das Gradientenverfahren und andererseits das Newton-Verfahren.

A.2.1 Das Gradientenverfahren

Das Gradientenverfahren wird auch als *Verfahren des steilsten Abstiegs* bezeichnet. Grundsätzlich betrachtet man hier ein unrestringiertes Minimierungsproblem mit einer stetig differenzierbaren

Zielfunktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Die zentrale Idee ist die Verwendung der steilsten Abstiegsrichtung, wobei der Gradient

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \dots \\ \frac{\partial}{\partial x_N} f(\mathbf{x}) \end{pmatrix}$$

der Funktion $f(\mathbf{x})$ mit $\mathbf{x} = (x_1, \dots, x_N)$ die Richtung des steilsten Anstiegs von f an der Stelle \mathbf{x} angibt (vgl. Dietmar Hömberg 2005, S. 46; Kux 2011, S. 17). Beim Gradienten-Verfahren wird somit ausgenutzt, dass die zugrundeliegende Zielfunktion in Richtung ihres negativen Gradienten am stärksten abfällt. Man bestimmt also in jedem i -ten Iterationsschritt den negativen Gradienten und sucht einen skalaren, positiven Faktor, sodass die Funktion in Richtung des stärksten negativen Gradienten minimiert wird (vgl. Meywerk 2007, S. 282f.).

Das Gradienten(abstiegs)verfahren beruht auf den gemäß Alg. A.1 folgenden Schritten:

Algorithmus A.1 Das Gradientenverfahren (vgl. Ulbrich und Ulbrich 2012, S.18):

Man wähle einen Startpunkt $\mathbf{x}^0 \in \mathbb{R}^n$ und vollzieht für $k = 0, 1, 2, \dots$:

1. Prüfung in Bezug auf Abbruch (STOP, falls \mathbf{x}^k stationär ist).
 2. Berechnung der Richtung $\mathbf{s}^k \in \mathbb{R}^n$ des steilsten Abstiegs, d.h. eine Richtung mit $\nabla f(\mathbf{x}^k)^T \mathbf{s}^k < 0$.
 3. Bestimmung einer Schrittweite $\sigma_k > 0$, sodass $f(\mathbf{x}^k + \sigma_k \mathbf{s}^k) < f(\mathbf{x}^k)$ gilt und die Abnahme der Zielfunktion, $f(\mathbf{x}^k) - f(\mathbf{x}^k + \sigma_k \mathbf{s}^k)$ hinreichend groß ist.
 4. Setze $\mathbf{x}^{k+1} = \mathbf{x}^k + \sigma_k \mathbf{s}^k$. Setze $k = k + 1$ und gehe zu Schritt 1.
-

Der Vektor $\mathbf{s} \in \mathbb{R}^n \setminus \{0\}$ zeigt die Abstiegsrichtung der stetig differenzierbaren Funktion f im Punkt \mathbf{x} an, wenn auf diesem die Steigung von f in Richtung \mathbf{s} negativ ist, d.h. unter der Bedingung, dass $\nabla f(\mathbf{x})^T \mathbf{s} < 0$ gilt (vgl. Ulbrich und Ulbrich 2012, S. 19f.). Die Schrittweite wird oft nach der *Armijo-Regel* bestimmt (vgl. Dietmar Hömberg 2005, S. 42). Gradientenverfahren konvergieren auf Grund ihres *Hill-Climbing* Charakters gegen ein lokales Minimum, welches nicht unbedingt ein globales Minimum darstellt (vgl. Meywerk 2007, S. 283). Zusammenfassend hält Homberg fest, dass das Gradientenverfahren einfach implementierbar ist und gegen Ende des Verfahrens langsam konvergiert (vgl. Dietmar Hömberg, 2005, S. 46).

Bestimmung der Schrittweiten: Bezüglich des allgemeinen Problems der Wahl der Schrittweite existieren weitere Verfahren, beispielsweise das Intervallschachtelungsverfahren (Goldener Schnitt), Quadratische Interpolation oder auch die heuristische Wahl der Schrittlänge (vgl. Kugi 2011, S. 28ff.). Das Intervallschachtelungsverfahren basiert auf einer konvergierenden Folge von Intervallschachtelungen, um das Minimum einzugrenzen. Es ist einfach und robust, benötigt aber vergleichsweise viele Iterationen zur Bestimmung der Schrittlänge. Ein relativ effizientes Verfahren stellt z.B. die quadratische Interpolation dar, welche darauf basiert, mittels drei Punkten eine quadratische Interpolationsfunktion zu legen. Unter der Annahme, dass die voneinander paarweise verschiedenen Punkte $\alpha_{k,1}$, $\alpha_{k,2}$ und $\alpha_{k,3}$ bekannt sind, lautet die quadratische Interpolationsfunktion $q(\alpha_k)$ durch diese Punkte (vgl. Kugi 2011, S. 29):

$$q(\alpha_k) = \sum_{i=1}^3 g_i \frac{\prod_{j \neq i} (\alpha_k - \alpha_{k,j})}{\prod_{j \neq i} (\alpha_{k,i} - \alpha_{k,j})} \quad (A.1)$$

Heuristische Verfahren beruhen, im Gegensatz zu numerischen Verfahren, auf heuristischen Abbruchkriterien und verwenden z.B. die auf einer Taylorreihe basierende Armijo-Regel, oder der Wolfe-Bedingung, wenn die Ableitungen der Gütefunktion $g(\alpha_k)$ einfach berechnet werden können (vgl. Kugi 2011, S. 30-32).

A.2.2 Das Newton-Verfahren

Das Newton-Verfahren kann zur Lösung nichtlinearer Gleichungssysteme als auch zur Minimierung nichtlinearer Funktionen verwendet werden (vgl. Ulbrich und Ulbrich 2012, S. 41-48; Dietmar Hömberg 2005, F. 58-64 und F. 246-258).

Das Newton-Verfahren für Gleichungssysteme: Das Verfahren wird zunächst in der Form $f(x) = 0$ mit einer stetig differenzierbaren Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Für ein berechnetes $\mathbf{x}^k \in \mathbb{R}^n$ gilt in äquivalenter Weise $f(\mathbf{x}^k + \mathbf{s}) = 0$. Die Taylor-Entwicklung approximiert die Funktion $f(\mathbf{x}^k + \mathbf{s})$ wie folgt: $f(\mathbf{x}^k + \mathbf{s}) = f(\mathbf{x}^k) + f'(\mathbf{x}^k) \cdot \mathbf{s} + \rho(\mathbf{s})$, wobei für kleine \mathbf{s} das Restglied $\rho(\mathbf{s})$ sehr klein ist. In der k -ten Iteration wird die ursprünglich angesetzte Gleichung $f(\mathbf{x}^k + \mathbf{s}) = 0$ durch eine linearisierte Form ersetzt: $f(\mathbf{x}^k) + f'(\mathbf{x}^k) \cdot \mathbf{s} = 0$. Die folgenden zentralen Schritte aus Alg. A.2 fassen das Konzept kurz zusammen (vgl. Ulbrich und Ulbrich 2012, S. 41f.):

Algorithmus A.2 Lokales Newton-Verfahren für Gleichungssysteme (vgl. Ulbrich und Ulbrich 2012, S. 42)

Auswahl eines Startpunktes $\mathbf{x}^0 \in \mathbb{R}^n$. Für $k = 0, 1, 2, \dots$:

1. Abbruch, wenn $f(\mathbf{x}^k) = 0$.
 2. Berechnung des *Newton-Schrittes* $\mathbf{s}^k \in \mathbb{R}^n$ durch Lösen der *Newton-Gleichung* $f'(\mathbf{x}^k) \cdot \mathbf{s}^k = -f(\mathbf{x}^k)$.
 3. Setze $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k$. Setze $k = k + 1$ und gehe zu Schritt 1.
-

Das Newton-Verfahren für Optimierungssysteme: Das Newton-Verfahren kann auch zur Minimierung zweimal stetig differenzierbarer Funktionen verwendet werden. Prinzipiell gibt es zwei Herleitungen. Die erste beruht auf einer quadratischen Approximation der zu minimierenden Funktion f in jedem Iterationsschritt. Diese quadratische Approximation wird mit Hilfe einer Taylorreihe der Funktion f in Bezug auf den aktuellen Punkt bestimmt. An Stelle der Funktion f bestimmt man das Minimum der approximierenden Funktion. Notwendige Bedingung für ein Minimum ist das Verschwinden der ersten Ableitung dieser Funktion. Die Lösung dieser Gleichung ist das Parametertupel des nächsten Iterationsschritts. Die Berechnung eines Schrittes kann sich wegen der ersten und zweiten Ableitung relativ aufwändig gestalten. Falls diese nur über Differenzenquotienten berechenbar sind, steigt der Aufwand quadratisch mit der Anzahl der Parameter. Newton-Verfahren konvergieren bei genügend glatten Zielfunktionen in der Nähe des Optimums quadratisch. Eine andere Herleitung fordert die notwendige Bedingung erster Ordnung ein, nämlich die Gleichung $\nabla f(\mathbf{x}) = 0$. Sei f zweimal stetig differenzierbar, dann ist der Gradient $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}$ stetig differenzierbar und es kann das Newton-Verfahren auf das Gleichungssystem $\nabla f(\mathbf{x}) = 0$ angewendet werden, siehe Alg. A.3 (vgl. Meywerk 2007, S. 282f.; Ulbrich und Ulbrich 2012, S. 46f.).

Algorithmus A.3 Das Newton-Verfahren für Optimierungsprobleme (vgl. Ulbrich und Ulbrich 2012, S. 47)

Auswahl eines Startpunktes $\mathbf{x}^0 \in \mathbb{R}^n$. Für $k := 0, 1, 2, \dots$:

1. Abbruch, wenn $\nabla f(\mathbf{x}^k) = 0$.
 2. Berechnung des *Newton-Schrittes* $\mathbf{s}^k \in \mathbb{R}^n$ durch Lösen der *Newton-Gleichung*: $\nabla^2 f(\mathbf{x}^k) \mathbf{s}^k = -\nabla f(\mathbf{x}^k)$.
 3. Setze $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k$. Setze $k = k + 1$ und gehe zu Schritt 1.
-

A.3 Das Nelder-Mead-Simplex Verfahren

Definition: Seien $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^n$ linear unabhängige Vektoren. Dann heißt die konvexe Hülle der Punkte $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(n)}$

$$\mathbb{S} = \left\{ \sum_{i=0}^n \lambda_i \mathbf{x}^{(i)} \mid \lambda_i \geq 0; i = 0, \dots, n; \sum_{i=0}^n \lambda_i = 1 \right\} \quad (A.2)$$

n -dimensionales Simplex mit den Ecken $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(n)}$ (vgl. Kux 2011, S. 18f.).

Der Punkt $\mathbf{s}^{(j)}$ mit $j \in \{1, \dots, n\}$ mit

$$\mathbf{s}^{(j)} = \frac{1}{n} \sum_{i=0, i \neq j}^n \mathbf{x}^{(i)} \quad (A.3)$$

wird als Schwerpunkt von $\mathbf{x}^{(j)}$ bezeichnet. Das Verfahren startet mit einem Anfangssimplex und wird an den Eckpunkten ausgewertet. Darauf basierend wird der Simplex schrittweise verändert, indem Punkte nach bestimmten Regeln durch neue ersetzt werden. Dabei wird in jeder Iteration die schlechteste Ecke des Simplex durch einen besseren Punkt ausgetauscht. Dies wird als Austauschoperation bezeichnet. Es gibt dabei die folgenden *Konstruktionsprinzipien*, um den jeweils schlechtesten Punkt zu ersetzen (vgl. Meywerk 2007, S. 279-281; Kux 2011, S. 19f.):

Punktspiegelung (Reflexion): Der schlechteste Punkt $\mathbf{x}^{(j)}$ wird am Schwerpunkt $\mathbf{s}^{(j)}$ reflektiert, sodass man den reflektierten Punkt \mathbf{x}_R erhält.

Expansion: Der reflektierte Punkt \mathbf{x}_R wird weiter in Richtung von $\mathbf{x}_R - \mathbf{s}^{(j)}$ verschoben, sodass man den expandierten Punkt \mathbf{x}_E erhält.

Kontraktion: Bei der Kontraktion unterscheidet man zwischen den folgenden drei Typen:

- Innere Kontraktion in Richtung $\mathbf{x}^{(j)} - \mathbf{s}^{(j)}$ ergebend den Punkt $\mathbf{x}_C = \mathbf{s}^{(j)} + \alpha(\mathbf{x}^{(j)} - \mathbf{s}^{(j)})$; mit $0 < \alpha < 1$
- Äußere Kontraktion in Richtung $\mathbf{x}_R - \mathbf{s}^{(j)}$ ergebend den Punkt $\mathbf{x}_C = \mathbf{s}^{(j)} + \alpha(\mathbf{x}_R - \mathbf{s}^{(j)})$; mit $0 < \alpha < 1$
- Schrumpfen bzw. totale Kontraktion bezüglich $\mathbf{x}^{(k)}$ mit $k \in \{1, \dots, n\}$, wobei alle Punkte $\mathbf{x}^{(i)}$ mit $i = 0, \dots, n, i \neq k$ durch die Punkte $\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} + \frac{1}{2}(\mathbf{x}^{(k)} - \mathbf{x}^{(i)})$ ersetzt werden. Als Kontraktionszentrum wird der beste Punkte $\mathbf{x}^{(k)}$ des Simplex herangezogen.

Meywerk und Kux erklären die Konstruktionsoperatoren auch grafisch. Der Pseudocode für Algorithmus A.4 fasst die bisherigen Schritte strukturiert zusammen. Daraus ist unmittelbar ersichtlich, dass das Verfahren im n -dimensionalen Raum lokal konvergiert. In Abhängigkeit vom Startsimplex sowie der Optimierungsaufgabe werden die Reflektions-, Expansions- und Kontraktionsoperatoren ausgeführt, bis das nächste lokale Minimum erreicht bzw. eine Abbruchbedingung erfüllt worden ist. Auf Grund der lediglich lokalen Optimierung des Algorithmus scheidet die Methodik im Rahmen dieser Arbeit aus, obwohl die Idee hinter dem Downhill-Simplex deutlich vielfältiger ist als die eines reinen Hill-Climbing Verfahrens wie z.B. dem Gradienten Verfahren (vgl. Meywerk 2007, S. 279-281; Kux 2011, S. 18-20).

Algorithmus A.4 Das Nelder-Mead-Simplex Verfahren (vgl. Kux 2011, S. 20)**Input:** *Optimierungsaufgabe***Output:** *Opt.Simplex*

- 1: Startsimplex erzeugen.
- 2: **while** Abbruchbedingungen nicht erfüllt **do**
- 3: Finde schlechteste \mathbf{x}^j und beste \mathbf{x}^k Ecke des Simplex.
- 4: Berechne Schwerpunkt \mathbf{s}^j des Simplex in Bezug auf \mathbf{x}^j .
- 5: Reflektiere \mathbf{x}^j an $\mathbf{s}^j \leftarrow \mathbf{x}_R$.
- 6: **if** x_R ist besser als alle anderen Ecken **then**
- 7: Führe Expansion aus $\leftarrow \mathbf{x}_E$.
- 8: **else**
- 9: **if** x_R ist größer als alle anderen Ecken **then**
- 10: Führe innere/äußere Kontraktion aus $\leftarrow \mathbf{x}_C$.
- 11: **else**
- 12: Führe totale Kontraktion aus.
- 13: **end if**
- 14: **end if**
- 15: **end while**

A.4 Lokale Konstruktions- und Verbesserungsheuristiken

Dieser Abschnitt diskutiert wesentliche lokale Konstruktions- und Verbesserungsheuristiken, welche sowohl eigenständig als auch als Teil von anderen (z.B. genetischen) Algorithmen eingesetzt werden können Sathyan et al. 2015, S. 1-16; Tseng und Lin 2009, S. 84-92; Karapetyan und Gutin 2011, S. 221-232).

A.4.1 K-Opt Algorithmus

Ein konkretes Beispiel liefert der bekannte k -Opt-Algorithmus, welcher häufig für das TSP oder auch ein PFSP verwendet wird. Das Prinzip beruht auf dem für $k = 2$ paarweisen Austausch (englisch: *pairwise exchange*) von benachbarten Lösungsteilen, siehe Alg. A.5 (vgl. Boelter 2013, S.3; Jungnickel 2010, S. 480; Sathyan et al. 2015, S. 6).

Algorithmus A.5 Der k-Opt Algorithmus (vgl. Boelter 2013, S.3)**Input:** Eine Instanz (K_n, c) des TSP, wobei K_n der vollständige Graph mit n Knoten und $c : E(K_n) \rightarrow \mathbb{R}_+$ die Kostenfunktion darstellt**Output:** Eine k -optimale Rundreise T

1. Sei T eine beliebige Rundreise in Bezug auf die Instanz.
2. Sei S die Familie der k -elementigen Teilmengen von $E(T)$, also $S := \{X \subseteq E(T) : |X| = k\}$.
3. **for all** $X \in S$ und Touren T' mit $E(T') \supseteq E(T) \setminus X$ **do**
 if $c(E(T')) < c(E(T))$ **then** setze $T := T'$, Gehe zu Schritt 2.

Ein wesentliches Merkmal des Algorithmus ist Enumeration aller T' , die sich in bis zu k Kanten von T unterscheiden, siehe Schritt 3 aus Alg. A.5. Eine Rundreise oder Tour wird als k -opt bezeichnet, wenn sie mit dem k -Opt Algorithmus nicht weiter verbessert werden kann. Zumeist wird für $k = 2$ oder $k = 3$ verwendet, um sogenannte 2-optimale bzw. 3-optimale Touren zu erhalten Während ein größerer k -Wert zumeist eine bessere Lösung ergibt, steigt auch die Laufzeit exponentiell, d.h. mit $\mathcal{O}(n^k)$, an. Es gibt auch den 2.5-opt Austauschschritt. Hierbei werden drei Kanten entfernt, wobei auch die Stadt, die von den zwei zu entfernenden Kanten umgeben ist, entfernt und an einer anderen Stelle wieder eingefügt wird. Die neu erhaltene Tour wird dabei durch die Einfügung zweier neuer Kanten

zur verschobenen Stadt sowie einer Kante zwischen den beiden Teilen der Tour gebildet (vgl. Boelter 2013, S.3; Karapetyan und Gutin 2011, S. 221-232; Sathyan et al. 2015, S. 6).

Sand betrachtet die Distanzveränderung der einzelnen Moves sowie die Unterschiede zwischen Inter- und Intra-Route-2-Opt-Moves und illustriert die Systematik des Verfahrens im Rahmen des in seiner Arbeit behandelten Capacitated-Vehicle-Routing-Problems (vgl. Sand 2013, S. 17-20).

A.4.2 Insertion Search

Auf Basis von Ergebnissen aus Optimierungsläufen sind Tseng und Lin der Meinung, dass ihr vorgestelltes *Insertion Search* (IS) Verfahren (vgl. Tseng und Lin 2009, S. 84-92), speziell für PFSP, anderen lokalen Suchverfahren wie z.B. dem paarweisen Austausch bei $k - Opt$, überlegen ist. Sie stellen außerdem ein globales Suchverfahren, *Insertion Search with Cut-and-Repair* (ISCR), vor. ISCR stellt eine Kombination aus IS und dem Cut-and-Repair (CR) Verfahren dar. Schritt 3 aus Algorithmus A.6 beschreibt den Kern des Insertion Search Verfahrens. Die Idee beruht darauf, $2 \cdot n \cdot \alpha$ Lösungen in der Umgebung in Bezug auf die aktuelle Lösung durch ein Remove and Insert basiertes Verfahren abzusuchen bzw. zu bewerten und die jeweils beste Lösung aus der aktuellen Nachbarschaft auszuwählen. Anschließend wird, ausgehend von der aktuell besten Lösung, die Suchliste re-initialisiert und Schritt 3. wiederholt sich. Erwähnenswert ist dabei, neben dem Operator, dass die Lösung durch mehrere Nachbarschaften *wandern* kann. Dies führt dazu, dass vielversprechendere Lösungsbereiche, in einem definierten Radius, untersucht werden (vgl. Tseng und Lin 2009, S. 87).

Algorithmus A.6 Insertion Search (Π, C, α) (vgl. Tseng und Lin 2009, S. 87)

Input: Basislösung Π , Zielfunktion C , Reichweite α

Output: Optimierte Lösung Π' , Optimierte Zielfunktion C'

1. Initialisierung der Suchliste $SL \leftarrow \{1, 2, \dots, n\}$.
 2. **if** (SL is not empty) **then**
 - Selektiere einen zufällig ermittelten Index p aus SL und entferne p aus SL .
 - else**
 - stop** and **return** Π' und C' .
 3. Execute Remove and Insert mit den Operatoren (p, k) auf Π , wobei $k = \{p - 1, p - 2, \dots, \max(p - \alpha, 1)\}$ und $p = \{p + 1, p + 2, \dots, \min(p + \alpha, n)\}$.
 4. Berechne C für alle durchgeführten Änderungsvorschläge und wähle den Vorschlag mit besten C' .
 5. **if** ($C' < C$) **then**
 - Update $\Pi \rightarrow \Pi'$ auf Basis von $C \rightarrow C'$ und gehe zu Schritt 1 (Re-Initialisierung von SL).
 - else**
 - Gehe zu Schritt 2.
-

Das Cut-and-Repair Verfahren ist sehr problemspezifisch in Bezug auf ein PFSP. Es beruht auf dem *greedy* Ansatz, zwei Paare benachbarter Aufträge zu ermitteln, welche die größten Leerzeiten (englisch: idle times) aufweisen und entsprechend auszuschneiden sind. Anschließend wird ein anderer Auftrag in die ausgeschnittene Position eingefügt in der Hoffnung, dass die Leerzeiten dadurch reduziert werden. Konkret werden mehrere an die Cut-Position einzufügende Punkte bewertet, wobei die besten Punkte gespeichert werden. Das Verfahren arbeitet insofern *semi-stochastisch*, als zu 50% die beste Lösung ausgewählt wird, bzw. zu den anderen 50% eine zufällige aus den besten Lösungen ausgewählt wird (vgl. Tseng und Lin 2009, S. 84-92).

A.4.3 Lin-Kernighan Heuristik

Die Lin-Kernighan Heuristik (LKH) ist eine der erfolgreichsten Heuristiken in Bezug auf das Travelling Salesman Problem und kann auch auf andere Probleme angewendet werden. Prinzipiell stellt das LKH Verfahren eine Generalisierung des $k - Opt$ Verfahrens dar. Das LKH Verfahren sucht nicht die

gesamte k -optimale Nachbarschaft ab, sondern nur einige vielversprechende Teile daraus. Es entfernt eine Kante aus einer Tour und ordnet den entstandenen Weg neu an, sodass die Zielfunktionsbewertung minimiert wird. Abschließend wird die eine entfernte Kante, die bei der Optimierung des Weges nicht berücksichtigt worden ist, wieder neu eingefügt. Dadurch kann die gesamte Rundreise verbessert werden. Das Prinzip der Neuordnung beruht auf einem Schema, welches den Weg in zwei Teile aufteilt, einen davon invertiert, und anschließend wieder verbindet, siehe auch Abbildung A.1 diesbezüglich. Dabei wird deutlich, dass das Verfahren eine Kombination aus einem austauschbasierten Verfahren und einem remove and insert basiertem Verfahren darstellt. Die Gewichtung des neu entstandenen Weges reduziert sich um den folgenden Term: $w(x \rightarrow y) - w(x \rightarrow e)$ (vgl. Karapetyan und Gutin 2011, S. 221-232).

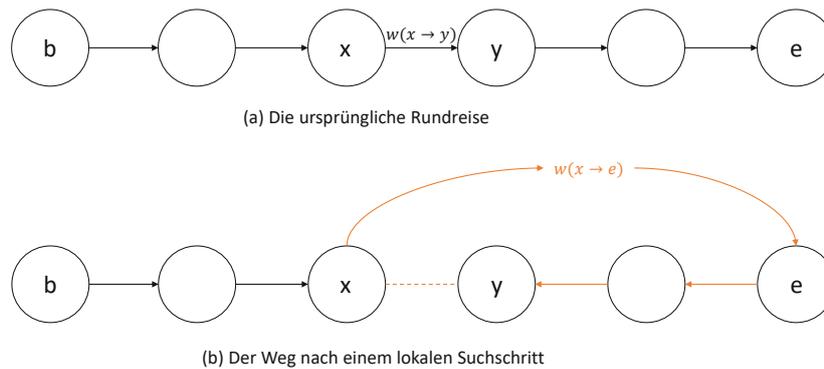


Abb. A.1: Lin-Kernighan Verfahren: Lokale Neuordnung (vgl. Karapetyan und Gutin 2011, S. 222)

Die grundlegenden Schritte zum Verfahren sind in Algorithmus A.7 dargestellt. In Bezug auf Schritt (2a) wird auf Abbildung A.1 verwiesen. Karapetyan beschreibt abgesehen vom grundlegenden Lin-Kernighan Verfahren auch diverse rekursive bzw. lokale Adaptionen.

Algorithmus A.7 Die Lin-Kernighan Heuristik (vgl. Karapetyan und Gutin 2011, S. 222)

Input: Basislösung T

Output: Optimierte Rundreise T'

1. Sei T die originale Rundreise.
 2. **for each** Kante $e \rightarrow b \in T$ **do**
 - (a) Sei $P = b \rightarrow \dots \rightarrow e$ der erhaltene Pfad durch das Entfernen der Kante $e \rightarrow b$.
 - (b) **Rearrange** P , um die Bewertung zu minimieren.
Jedes Mal, wenn eine Verbesserung gefunden worden ist, versuche den Weg zu einer Tour zu schließen. Wenn diese Tour besser bewertet ist, speichere die Tour als $T \rightarrow T'$ ab. **Repeat procedure**.
 - (c) Wenn keine verbesserte Tour gefunden wird, setze mit der nächsten Kante fort (Schritt 2) oder **Stop**.
-

A.5 Goal-Chasing Heuristik

Monden stellt 1983 die auf Toyota angewendete *Goal-Chasing Heuristik* (GCH) vor. Diese erreicht einen gleichmäßigen Vorproduktbedarf durch eine schrittweise Minimierung der folgenden Gleichung (A.4), welche die gesamtheitlich normierte Abweichung in Bezug auf die komplette Menge an C Komponenten in Bezug auf eine ideale Verbrauchsannahme wie folgt darstellt:

$$D_{m,n} = \sum_{j=1}^C [n \cdot \frac{c_j}{N} - (X_{j,n-1} + c_{jm})]^2 \quad (A.4)$$

Dabei stellt N die Gesamtzahl der montierenden Bauteile für ein Endprodukt m dar, c_j die erforderliche Gesamtzahl an Komponenten zur Erstellung von N . Dadurch ergibt sich der erste Term zu unter der

Summe als erwartete ideale Nutzung einer Komponente j , die für die Montage der ersten n Teile benötigt wird. Der Term $X_{j,n-1}$ stellt die benötigte Menge der Komponente j dar, um die ersten $n - 1$ Teile zu montieren. Der Ausdruck c_{jm} stellt die Menge der Komponente j dar, welche zum Zusammenbau einer Einheit des Endprodukts benötigt wird. In der Abfolge wird schrittweise ein Modell m^* ausgewählt, das die Minimierung von $D_{m,n}$ ermöglicht, wobei $D_{m,n}$ den gesamten Bedarf an n Teilen für ein Endprodukt m darstellt (vgl. Celano et al. 2004, S. 575).

Bautista beschreibt die Goal-Chasing Heuristik als Methode, welche auf einem progressiv konstruierten Weg entlang eines Graphen aufgebaut wird. Der Graph wird dabei in jedem Schritt um eine hinzugefügte Kante, welche in Bezug auf ihren beitragenden Wert ausgewählt wird, erweitert (vgl. Bautista et al. 1996, S. 104ff.).

Das Ziel besteht in der Findung einer Endproduktreihenfolge, welche die Schwankungen der Vorproduktbedarfe reduziert. Die Heuristik ist insbesondere bei der Verwendung rekursiver Prozeduren¹ sehr schnell, jedoch bleiben durch ihre *greedy* Charakterisierung häufig ungünstige Kanten für das Ende der Sequenzierung übrig, wodurch das Gesamtergebnis erheblich negativ beeinträchtigt wird. Bautista schlägt deshalb einige Verbesserungen z.B. bezüglich Symmetrie und Betrachtungshorizont vor, um die genannten negativen Effekte zu reduzieren. So empfiehlt es sich in Bezug auf den Betrachtungshorizont nicht nur die Auswirkung eines Schrittes, sondern z.B. zweier aufeinanderfolgender Schritte zu betrachten, um so einerseits günstige Kanten-Kombinationen als auch andererseits einen ausgedehnteren Betrachtungshorizont für das Problem zu entwickeln. Celano hält auch fest, dass die Betrachtung des jeweils nur nächsten Schrittes einen Mangel der Heuristik darstellt. Dementsprechend empfiehlt er die Auswahl des n -ten Produkts innerhalb der Sequenz in Bezug auf den Ausdruck (A.4) für die Positionen n und $n + 1$ zu bewerten. So ergibt sich für ein Endprodukt m^* in Schritt n der Ausdruck $D_{m^*,n} + \min_m \{D_{m,n+1}\}$. Ein weiterer schwerwiegender Nachteil der GCH tritt auf, wenn die verbleibenden zu sequenzierenden Teile bereits zuvor ausgebeutet bzw. entleert worden sind. So empfiehlt es sich einen dementsprechend erhaltenden Mechanismus zu haben, der auf die Verbrauchsrate achtet bzw. in Bezug auf den Teileverbrauch ungünstige Zusammenstellungen vermeidet (vgl. Celano et al. 2004, S. 575; Bautista et al. 1996, S. 104ff.; Tegel 2012, S. 34f.).

Celano betrachtet in seinem Beitrag neben der Goal-Chasing Heuristik für sein Sequenzierungsproblem auch Erweiterungen bzw. Modifikationen und vergleicht diese Ansätze mit einem traditionellen SA. Im Zuge einer statistischen Auswertung von drei problemspezifisch angepassten Heuristiken und dem SA bzw. eines dazugehörigen Vergleichs von 90 Probleminstanzen aufgeteilt in 9 Problemklassen zeigt sich, dass der SA in den meisten Instanzen (74, 82 bzw. 90 aus 90 Szenarien) den greedy heuristisch basierten Ansätzen leicht überlegen ist. In Analogie zu den einfacheren (greedy-basierten) Heuristiken aus der Graphentheorie, siehe Kruskal's Algorithmus 3.1 in dieser Arbeit, zeigt sich anhand der Ergebnisse auch hier, dass gegen Ende der jeweiligen Heuristik relativ schlechte Züge übrig bleiben. Celano schlussfolgert auf Basis der relativ flachen Funktionen, dass deshalb die Unterschiede zwischen den Verfahren einerseits relativ gering sind und es sich andererseits als schwierig gestaltet eine Sequenz zu finden, die das Ziel der Komponentenverwendung erfüllt. Dafür ist die GCH im Vergleich mit Metaheuristiken deutlich schneller in der Laufzeitausführung (vgl. Celano et al. 2004, S: 573-581).

A.6 Weitere Ersatzoptimierungsverfahren

A.6.1 Methoden ohne Präferenz

Methoden ohne Präferenz berücksichtigen die Meinung der Stakeholder*innen nicht bzw. ist dessen Präferenz explizit nicht ausgedrückt oder verfügbar. Diese Methoden können auch dazu verwendet werden, um einen Startpunkt für z.B. interaktive Methoden zu generieren (vgl. Branke et al. 2008, S. 13).

¹Eine Rekursion stellt eine sich selbst aufrufende Methode dar.

A.6.1.1 Global Criterion Methode

Im Zuge der Global Criterion Methode wird die Distanz zwischen einem erstrebenswerten Referenzpunkt im Zielraum und eines zulässigen Punkts im Lösungsraum minimiert. Dazu kann die L_p -Metrik oder auch Tchebycheff Metrik (L_∞) verwendet werden. Der*die Analyst*in wählt den zu verwendenden Referenzpunkt aus und setzt diesen als idealen Vektor z^* oder utopischen Vektor z^{**} und löst das folgende Problem (vgl. Branke et al. 2008, S. 14):

$$\text{minimiere } \sum_{i=1}^k |f_i(\mathbf{x}) - z_i^*|^p \quad (\text{A.5}) \text{ für alle } \mathbf{x} \in S$$

oder, durch Weglassen des Exponenten $1/p$:

$$\text{minimiere } \max_{i=1, \dots, k} [|f_i(\mathbf{x}) - z_i^*|] \quad (\text{A.6}) \text{ für alle } \mathbf{x} \in S.$$

Die Wahl der Metrik hat einen erheblichen Einfluss auf die Lösung. Im Falle unterschiedlicher Magnituden müssen die einzelnen Funktionen zuerst skaliert werden, um eine (dimensionslose) Norm zu erhalten. Mietinen liefert einen Beweis für die Pareto-Optimalität der erhaltenen Lösungen (vgl. Branke et al. 2008, S. 14):.

A.6.1.2 Neutrale Kompromisslösung

Im Zuge der Anwendung dieser einfachen Methode wird auf einen Punkt (irgendwo) in der Mitte der Zielwertebereiche in der Pareto-optimalen Menge projiziert, sodass dieser zulässig wird. Eine neutrale Kompromisslösung wird durch die Lösung des folgenden Problems erreicht (vgl. Branke et al. 2008, S. 14f.):

$$\text{minimiere } \max_{i=1, \dots, k} \left[\frac{f_i(\mathbf{x}) - ((z_i^* + z_i^{nad})/2)}{z_i^{nad} - z_i^{**}} \right] \quad (\text{A.7}) \text{ für alle } \mathbf{x} \in S$$

Hier verwendet man den (nicht notwendigerweise zulässigen) utopischen als auch Nadir Zielfunktionsvektor bzw. verlässliche Approximationen in Bezug auf die Zielfunktionsreichweiten in der Pareto-optimalen Lösungsmenge für Skalierungszwecke (vgl. Branke et al. 2008, S. 14f.).

A.6.2 Weitere interaktive Präferenz ausdrückende Methoden

A.6.2.1 Interaktive Surrogate Worth Trade-Off Methode

Diese Methode basiert auf der Epsilon-Constraint Methode, wobei eines der Ziele in Bezug auf obere Schranken bei allen anderen Zielsetzungen (vgl. Ehrgott und Gandibleux 2003, S. 232f.) minimiert und eine Pareto-optimale Lösung erhalten wird (vgl. Branke et al. 2008, S. 35). Das Minimierungsproblem hat die folgende Form:

$$\min f_l(\mathbf{x}) \text{ wobei: } (f_j(\mathbf{x}) \leq \varepsilon_j \text{ für alle } j = 1, \dots, k, j \neq l, \text{ für alle } \mathbf{x} \in S) \quad (\text{A.8}),$$

wobei $l \in \{1, \dots, k\}$ und ε_j als obere Schranken für die anderen Teilziele fungieren. Dadurch, dass die anderen Ziele als Nebenbedingungen ausgedrückt werden, wird das vorliegende multikriterielle Problem in ein einkriterielles Problem verwandelt. Gültige Lösungen sind schwach Pareto-optimal. Die oberen Schranken des Epsilon-Constraint Problems werden basierend auf den Stellvertreter-Werten angepasst. Die Idee der Methode basiert darauf, eine zugrunde liegende *Nutzenfunktion* maximal zu approximieren. Der Ablauf der Methode gliedert sich in vier Schritte (vgl. Ehrgott und Gandibleux 2003, S. 233f.):

1. Man wähle eine zu minimierende Funktion f_l aus und belegt das Problem mit oberen Schranken für die anderen Zielfunktionen. Man setze $h = 1$.

2. Man löse das zugrunde liegende formal bestimmte Minimierungsproblem, um eine Lösung \mathbf{z}^h zu erhalten. Die Trade-Off Informationen werden über die Karush-Kuhn-Tucker Multiplikatoren erhalten (vgl. Boyd und Vandenberghe 2004, S. 243f.).
3. Der*die Entscheidungsträger*in wird in Bezug auf den Stellvertreter-Wert im Intervall $[-10, +10]$ an der Lösung \mathbf{z}^h gefragt.
4. Abbruch gemäß einem Kriterium oder Update der oberen Schranken mit Hilfe des Feedbacks aus Phase 3 sowie der gelösten ε -*constraint* Probleme. Der*die Entscheidungsträger*in wählt die vielversprechendste Alternative \mathbf{z}^{h+1} aus und setzt $h = h + 1$. Gehe zu Schritt 3.

Die zugrunde liegende Idee hinter der Methode liegt also im Konzept des Stellvertreter Werts (*surrogate worth*), welcher eine Intervall begrenzte Bewertung des durchgeführten Trade-Offs durch den*die Entscheidungsträger*in an einer Pareto-optimalen Lösung darstellt. Die Schwachstellen der Methodik liegen in der Bereitstellung konsistenter Stellvertreter-Werte von Seiten des*der Entscheidungsträger*in über den gesamten Optimierungsprozess. Die Leichtigkeit des Vergleichs der Alternativen hängt von der Anzahl der Zielstellungen u. den personellen Ressourcen des*der Entscheidungsträger*in ab. Eine Nutzenfunktion muss implizit verfügbar sein (vgl. Ehrgott und Gandibleux 2003, S. 234).

A.6.2.2 Geoffrion-Dyer-Feinberg Methode

Diese Methode beruht auf der Maximierung einer zugrunde liegenden Nutzenfunktion und verwendet dabei den Frank-Wolfe-Algorithmus, der auf einer linearen Approximation der Nutzenfunktion basiert (vgl. Miettinen Kaisa 2001, S. 10f.). Dabei werden die von dem*der Entscheidungsträger*in spezifizierten Grenzersatzwerte m_i^h an der aktuellen Lösung \mathbf{x}^h verwendet, um die Richtung des steilsten Anstiegs der Nutzenfunktion zu approximieren, welche in Bezug auf die referenzierende Zielfunktion f_l abnimmt. Der Algorithmus gliedert sich in fünf Schritte (vgl. Ehrgott und Gandibleux 2003, S. 234f.):

1. Der*die Entscheidungsträger*in wird in Bezug auf eine referenzierende Zielfunktion f_l für die Ermittlung einer Ausgangslösung gefragt. Setze $h = 1$.
2. Der*die Entscheidungsträger*in soll die lokalen Grenzzraten der Substitution zwischen f_l und den anderen Zielen an der aktuellen Lösung \mathbf{z}^h (mit Hilfe des*der Analyst*in) definieren.
3. Anwendung der Methode von Frank und Wolfe: $\max \left(\sum_{i=1}^k -m_i^h \nabla_x f_i(\mathbf{x}^h) \right)^T y$ für alle $y \in S$.
Bestimmung der besten Verbesserungsrichtung mittels $\mathbf{d}^h = \mathbf{y}^h - \mathbf{x}^h$. Abbruch für $\mathbf{d}^h = 0$.
4. Bestimme mit der Hilfe des*der Entscheidungsträger*in die geeignete (nutzenmaximale) Schrittweite t^h in Richtung \mathbf{d}^h . Bezeichne die neu erhaltene Lösung $\mathbf{z}^{h+1} = f(\mathbf{x}^h + t^h \mathbf{d}^h)$.
5. Setze $h = h + 1$. Gehe zu Schritt 2, solange von dem*der Entscheidungsträger*in gewünscht, oder Abbruch.

Aus praktischer Sicht ist diese Ersatzoptimierungs-Methode nicht empfehlenswert. Bei jeder Iteration werden $k - 1$ korrekte Grenzersatzwerte konsistent bestimmt. Der Auswahlprozess wird für den*die Entscheidungsträger*in mit der Zunahme verschiedener Teilzielfunktionen schwieriger (vgl. Ehrgott und Gandibleux 2003, S. 234f.).

A.6.2.3 Das STEM-Verfahren

STEM (englisch: *step method*) basiert auf einem Idealpunktansatz und der sukzessiven Verringerung des Lösungsraums sowie der Klassifikation von Zielfunktionen an der aktuellen Lösung $\mathbf{z}^h = f(\mathbf{x}^h)$. Dementsprechend beruht die Methode auf der Annahme, dass der*die Entscheidungsträger*in angibt, wann eine Funktion annehmbare Funktionswerte aufweist, bzw. wann die Werte zu hoch oder unannehmbar sind. Dann wird dem*der Entscheidungsträger*in auferlegt, einen geringen Teil des akzeptierten Wertebereichs $f_i(i \in I^>)$, aufzugeben, um einige Werte aus dem bislang nicht akzeptierten Wertebereich $i \in I^<$ verbessern zu können. Der*die Entscheidungsträger*in muss also obere Schranken

$\varepsilon_i^h > f_i(\mathbf{x}^h)$ für Funktionen aus $I^>$ definieren. Demnach liegt die einzige Anforderung der Methode in der Definition von oberen Schranken, weil die Distanzen in Bezug auf den globalen (idealen) Zielfunktionsvektor gemessen werden. STEM verwendet die *gewichtete Tschebyscheff-Norm* (siehe Glg. 3.5) zur Minimierung. Die aus der Minimierung (siehe A.9 für alle $\mathbf{x} \in S$, für Vorschläge betreffend ε_i vgl. Ehrgott und Gandibleux 2003, S. 238) erhaltenen Lösungen sind schwach Pareto-optimal. Die Idee besteht ferner darin, sich entlang von schwach Pareto-optimalen Lösungen entlang zu bewegen. Nach der erfolgten Klassifikation der Zielfunktionen wird der zulässige Bereich mittels der Informationen von Seiten des*der Entscheidungsträger*in beschränkt. Die Gewichtungen der relaxierten Zielfunktionen werden gleichermaßen auf 0 gesetzt, sodass $e_i = 0$ gilt für alle $i \in I^>$. Anschließend wird das neu erhaltene Distanzminimierungsproblem:

$$\text{minimize } \max_{i=1, \dots, k} \left[\frac{e_i}{\sum_{j=1}^k e_j} f_i(\mathbf{x}) - z_i^* \right] \quad (\text{A.9})$$

unter den folgenden Bedingungen gelöst (vgl. Branke et al. 2008, S. 47; Ehrgott und Gandibleux 2003, S. 237-239):

$$f_i(\mathbf{x}) \leq \varepsilon_i^h \text{ für alle } i \in I^> \text{ sowie } f_i(\mathbf{x}) \leq f_i(x^h) \text{ für alle } i \in I^<, \mathbf{x} \in S \quad (\text{A.10}).$$

Die Prozedur wird solange fortgesetzt, bis der*die Entscheidungsträger*in keine Änderung an der aktuellen Lösung \mathbf{z}^h mehr vornehmen möchte oder sich verbesserte neue Lösungen berechnen lassen. Das Prinzip beruht darauf, sich von einer schwach Pareto-optimalen Lösung zur nächsten zu bewegen. Obwohl die Idee der Klassifikation relativ einfach ist, erweist es sich ggf. als schwierig, geeignete Schrittweiten zu schätzen, um entsprechende Verbesserungen zu erzielen (vgl. Branke et al. 2008, S. 47f.; Ehrgott und Gandibleux 2003, S. 238-239).

A.7 Monte-Carlo Verfahren

Das Prinzip des Monte-Carlo Verfahrens beruht darauf, Werte der Optimierungsvariablen im zulässigen Bereich zufällig zu bestimmen, auszuwerten und anschließend die beste Lösung daraus auszuwählen. Die Optimierungsvariablen werden als gleichverteilte Zufallsgrößen betrachtet. Das Verfahren ist für beliebige Probleme anwendbar. Es arbeitet nicht effektiv und hat andererseits auch keinerlei Anforderungen (bezüglich der Glattheit) an die Zielfunktion. Alle Varianten werden voneinander unabhängig betrachtet, daher kommt es speziell bei einer kleiner Anzahl an Varianten vor, dass der zulässige Bereich nicht gleichmäßig durchsucht bzw. die Pareto-Front gefunden wird. Daher ist es für die Praxis ungeeignet und wird als Referenz zur Beurteilung der Effektivität anderer Optimierungsverfahren herangezogen (vgl. Meywerk 2007, S. 29f.; Kux 2011, S. 29f.).

Zwei Spezialfälle des Monte-Carlo-Verfahrens werden teilweise in der Praxis verwendet, nämlich die *Latin-Hypercube-Methode* (LHM) und der sogenannte *Hill-Climbing-Algorithmus*. Mittels LHM ist es möglich, L Punkte eines N -dimensionalen Raums zu erzeugen, die den ganzen Raum möglichst gleichmäßig ausfüllen. Unter der Annahme, dass die Vektoren der unteren bzw. oberen Schranken, $x^{(U)}$ und $x^{(O)}$, pro Dimension gegeben sind, wird für jede Dimension i das Intervall $[x_i^{(U)}, x_i^{(O)}]$ in L Teile zerlegt, woraus sich L^N Zellen ergeben. Anschließend wird, basierend auf der Elimination von Zeilen und Spalten, ein Algorithmus entworfen, der jeden Parameter in seinem Wertebereich mit L Punkten bewertet. Dadurch kann zumindest der gesamte Wertebereich gleichmäßig abgesucht bzw. bewertet werden. Diese Methode kann als Initialisierung verwendet werden, wenn keine weiteren Informationen über die Fitnesslandschaft vorhanden sind und ist einer zufälligen Initialisierung deutlich überlegen (vgl. Stöcker 2007, S. 44; Prügel-Bennett 2004, S. 135-153; Press 2002, S. 315).

Das Hill-Climbing Verfahren beruht hingegen auf einem zufällig gewählten Startpunkt, von dem aus eine zufällige Veränderung bzw. Mutation (d.h. eine kleine Änderung in Bezug auf die aktuelle Lösung)

vorgenommen wird. Das Verfahren akzeptiert die neu erhaltene Lösung nur dann, wenn sie auch besser als die vorherige Lösung ist. Das Hill-Climbing Verfahren konvergiert rasch zu einem lokalen Minimum. Deshalb werden anstelle einer auch n multiple Mutationen vorgenommen. Vor dem nächsten Schritt werden die Variablen in der Form angepasst, sodass sie nur aus der Nachbarschaft um die beste Lösung (quasi als Zentrum) bestehen. Für relativ glatte Zielfunktionen, wo die Intervalle in jedem Iterationsschritt vergleichsweise stark reduziert werden können, ist das Verfahren in Kombination mit einkriteriellen Optimierungsaufgaben gut einsetzbar, sodass eine schnelle Konvergenz erreicht wird. Für multikriterielle Optimierungsaufgaben werden die Intervalle anhand der Pareto-Front festgelegt. Dies erschwert den Einsatz und reduziert die Effektivität des Verfahrens, weil die Pareto-Front i.d.R. vorab nicht bekannt ist (vgl. Kux 2011, S. 30f.; Prügel-Bennett 2004, S: 135-153).

Eine Generalisierung der genannten Verfahren stellt die Gruppe der als *Random Search* bekannten Methoden dar, zu denen eine Kombination aus Zufallszahlen und Greedy Heuristik basierten Verfahren gehört. Zu den Random Search Methoden zählt beispielsweise *Adaptive Random Search* oder auch die *Greedy Randomized Adaptive Search Procedure*, *GRASP* (vgl. Wari und Zhu 2016, S. 328-343). GRASP funktioniert auf Basis von zwei aufeinanderfolgenden Schritten: Der Konstruktion einer gültigen Lösung mittels einer Greedy Heuristik (vgl. Dorigo und Stützle 2004, S. 30), und der darauf angewendeten lokalen Suche (vgl. Boussaïd et al. 2013, S. 87; Glover und Kochenberger 2003, S. 219-249).

A.8 Weitere metaheuristische Optimierungsverfahren

In diesem Abschnitt werden weitere Metaheuristiken, die nicht in Kapitel 3 erläutert und diskutiert bzw. in dieser Arbeit nicht angewendet werden, beschrieben.

A.8.1 Ant Colony Optimization (ACO)

Die Gruppe der Ameisenalgorithmen, erstmals vorgestellt von Marco Dorigo und seinen Kollegen, v.a. Thomas Stützle, repräsentiert die bekannteste konstruktive Metaheuristik zur Lösung kombinatorischer Optimierungsprobleme. Das Funktionsprinzip beruht auf den Prinzipien einer Ameisenkolonie, diese beruht auf dem Phänomen der sogenannten *Schwarmintelligenz*. Dementsprechend klassifiziert Boussaïd Ameisenalgorithmen als Schwarmintelligenz basierende Algorithmen. Schwarmintelligenz stellt ein innovatives und verteiltes, intelligentes Paradigma für Optimierungsprobleme dar, welches vom Verhalten einer Gruppe von *sozialen Insekten* abstammt. Diese Systeme bestehen typischerweise aus einer Population von einfachen Agenten, die lokal miteinander und mit ihrer Umgebung interagieren. Jede einzelne Entität daraus ist in der Lage einzelne begrenzte Handlungen zu setzen (vgl. Dorigo und Stützle 2004, S. 1-221; Boussaïd et al. 2013, S. 98).

Definition 18. Schwarmintelligenz: „Dieser Begriff beschreibt ein System, also eine Gruppe von Akteuren, die einen Schwarm bilden, der in der Lage ist, als Gesamtheit „intelligentes Verhalten“ zu zeigen, zu dem die einzelnen Mitglieder der Gruppe alleine nicht in der Lage sind“ (Schmickl 2009, S. 143f.).

Gemäß Schmickl müssen fünf Grundvoraussetzungen erfüllt sein, damit eine Gruppe von Akteuren als schwarmintelligent definiert werden kann (vgl. Schmickl 2009, S. 145):

1. Das Prinzip der *Nachbarschaft*: Individuen des Schwarmes dürfen ihre Aktionen und ihre Wahrnehmung nur in einem lokal begrenzten Radius ausführen können.
2. Das Prinzip der *Qualität*: Der Schwarm soll dazu fähig sein auf Umweltreize angemessen zu reagieren.
3. Das Prinzip der *verschiedenartigen Antwort*: Der Schwarm soll über ein ausreichendes Verhaltensrepertoire verfügen.
4. Das Prinzip der *Stabilität*: Nicht jede kleinste Änderung an den Umweltbedingungen soll zu einer Verhaltensänderung des Schwarmes führen.

5. Das Prinzip der *Anpassungsfähigkeit*: Der Schwarm stellt sein Verhalten bei genügend großen Veränderungen der Umwelt entsprechend um, wenn dies zu einer besseren Überlebensfähigkeit beiträgt.

Während einige Punkte kausal voneinander in Abhängigkeit stehen, widersprechen sich andere. Das letzte Prinzip ist stark angelehnt an das *survival of the fittest* Prinzip beim GA. Der ACO-Algorithmus ist durch die Selbstorganisation von Ameisen bei ihrer Futtersuche inspiriert. Im Zuge des Doppelbrückenexperiments von Goss wurde 1989 herausgefunden bzw. bestätigt, dass Ameisen in der Lage sind, den schnellsten, d.h. besten, Weg zu ihrer Futterquelle zu finden (vgl. Petri 2006, S. 147ff.).

Erkenntnisse aus dem Doppelbrückenexperiment

Das Experiment beruht auf einem autokatalytischen (*reinforcement*) Prozess, wodurch der kürzere Weg mit zunehmender Zeit systematisch verstärkt gewählt wird. Auf dem Hinweg zur Futterquelle können Ameisen unabhängig voneinander einen kürzeren bzw. einen längeren Weg wählen. Auf dem Rückweg werden nun die langsameren von den schnelleren Ameisen, welche bereits auf dem Rückweg befindlich sind, auf Basis der abgelegten Pheromone beeinflusst. Dieser Effekt verstärkt sich mit zunehmender Zeit. Petri liefert dafür auch ein Zahlenbeispiel inklusive einer dazugehörigen Abbildung. Weiters wurde aus dem Brückenexperiment festgestellt, dass ein neu geschaffener, noch kürzerer Weg kaum ausgewählt wird, weil dort kein Pheromon liegt. Diese Erkenntnis zeigt gleichermaßen auch die Grenzen der Metaheuristik bzw. der Natur auf, dass sich Ameisen mit einem (vergleichsweise) guten Weg begnügen und nicht unbedingt den optimalen Weg zur Futtersuche finden bzw. benötigen. Ein weiteres Experiment liefert die Einsicht, dass bei Verfügbarkeit von zwei gleich guten Wegen nicht beide gleichermaßen genutzt werden, sondern sich einer der beiden Wege pheromonbedingt durchsetzt (vgl. Graf 2003, S. 15-53).

Abbildung A.2 beschreibt die Fähigkeit der Wegfindung von Ameisenkolonien bei der Futtersuche. Es kristallisiert sich nach einiger Zeit, trotz einiger Hindernisse, rasch ein bester Weg (Abb. A.2, rechter Teil) heraus, den schließlich (fast) alle Ameisen benutzen, während die Ameisen anfangs (Abb. A.2, linker Teil) auch andere (schlechtere) Wege ausprobieren bzw. *erforschen*. Gemäß der Graphentheorie sind hier beide Knotenpunkte (Ameisenhügel, Futterstelle) mit v_s bzw. v_d bezeichnet (vgl. Deepa und Senthilkumar 2016, S. 11f.).

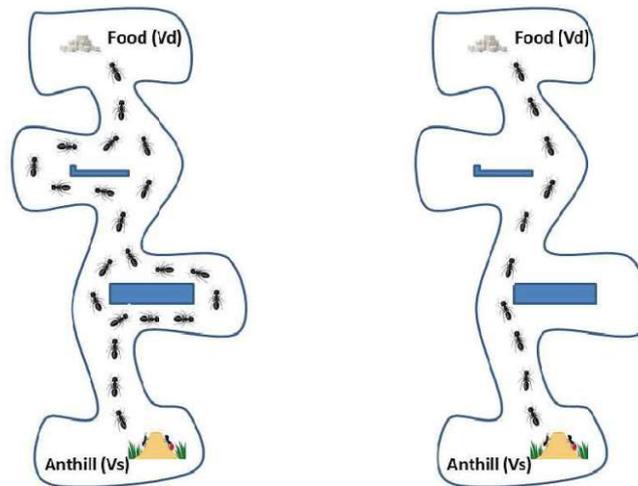


Abb. A.2: Die Fähigkeit der Wegfindung bei der Futtersuche von Ameisenkolonien (Deepa und Senthilkumar 2016, S. 12)

Diese Ergebnisse waren der Auslöser für die Entwicklung des Ameisenalgorithmus durch Dorigo im Jahr 1990. Die künstlich generierten Ameisen konstruieren und verändern eine Lösung auf Basis globaler Informationen, welche in der Pheromonmatrix abgelegt sind, sowie lokaler Informationen in Bezug auf die unmittelbare Nachbarschaft zu anderen Ameisen. Diese lokalen Informationen stellen das Verhalten jeder einzelnen Ameise bzw. ihr Gedächtnis dar und beurteilen die aktuelle Situation in Bezug auf

ihre Nachbarschaft. Den Schlüssel stellen die Pheromone dar, welche die Ameisen entlang ihres Pfades ablegen. Die Pheromone verdunsten und stellen somit ein wichtiges Kriterium für die Häufigkeit der Benutzung eines Pfades einerseits bzw. der grundsätzlichen Länge eines Weges andererseits dar. Das System besteht dementsprechend aus (einer Menge von) Ameisen, diskreten Zeitschritten, einer global verfügbaren Pheromonmatrix sowie einer lokalen Informationsmatrix (vgl. Petri, 2006, S. 147ff.).

Grundlegender Ablauf von Ameisenalgorithmen

Ameisenalgorithmen verfügen über den folgenden grundlegenden Ablauf (vgl. Boussaïd et al. 2013, S. 98f.):

1. Konstruktion einer Lösung durch jede einzelne Ameise in Abhängigkeit von lokalen bzw. globalen Informationen (Pheromonmatrix).
2. Durchführung des *Pheromon-Updates*: Auftragen neuer Pheromone durch Lösungswahl der Ameisen bzw. Verdunstung vorhandener Pheromone.
3. Berücksichtigung problemspezifischer Charakteristika.

Das Pheromon-Update kann dabei sowohl schrittweise (online), d.h. während der Konstruktion der Lösung, als auch einmalig nach der vollständig konstruierten Lösung erfolgen. Das Verdunsten von Pheromonen ist eine von den Ameisen unabhängige Funktion. Algorithmus A.8. beschreibt diese drei grundlegenden Schritte im Ablauf eines ameisenbasierten Algorithmus in Form von Pseudocode. Nach Initialisierung der Pheromonmatrix wird das Verfahren bis zu einem definierten Abbruchkriterium ausgeführt (Alg. A.8, Zeile 2 u. 14). Ab Zeile 3 konstruiert jede Ameise aus der Kolonie ihre individuelle Lösung. Dementsprechend muss die Menge der für die Konstruktion zulässigen Lösungselemente (z.B. Aufträge im Rahmen eines PFSP, Städte im Rahmen eines TSP, etc.) initialisiert werden. In Zeile 5 wird die leere Teillösung initialisiert, welche in den Zeilen 6 - 10 zusehends, jeweils in Bezug auf die aktuelle Nachbarschaft (Zeile 7) vervollständigt wird. Nach der Fertigstellung einer durch eine Ameise konstruierten Lösung wird diese optional lokal verbessert (Zeile 11). Nach erfolgter Konstruktion aller Lösungen erfolgt das Pheromon-Update (Petri 2006, S. 152-154).

Algorithmus A.8 Basisstruktur des ACO Algorithmus (Petri 2006, S. 153)

```

1: Initialisiere Pheromonmatrix  $T$ 
2: do {
3:   for ( $ant = 1$ ;  $ant \leq Ants$ ;  $ant = ant + 1$ ) {
4:     Initialisiere Menge der nicht ausgewählten Elemente:  $U = 1, 2, \dots, N$ 
5:     Initialisiere aktuelle Teillösung:  $\pi' = 0$ 
6:     while ( $U \neq 0$ ) {
7:       Wähle ein Element  $n$  aus  $U$  aus
8:       Füge  $n$  in  $\pi'$  ein
9:        $U = U \setminus n$ 
10:    } // end while
11:    Führe lokale Suche aus // optional
12:  } // end for
13:  Aktualisiere Pheromonmatrix  $T$ 
14: } while (!Abbruchbedingung)

```

Die vier Prinzipien der Selbstorganisation des ACO

Bevor die Beschäftigung mit den Prinzipien der Selbstorganisation erfolgt, wird zuerst die entsprechende Definition benötigt:

Definition 19. Selbstorganisation: „*Self-organization is a process in which a pattern at the global level of a system emerges solely from numerous interactions among lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern*” (Eisenberg 2007, S. 100).

Die vier Prinzipien der Selbstorganisation von Bonabeau finden sich im ACO Ansatz, basierend auf dem Doppelbrückenexperiment, wie folgt wieder (vgl. Petri 2006, S. 100; Bonabeau et al. 1999, S. 9-12):

1. *Positives Feedback*: Mit der zunehmenden Nutzung eines Weges (bzw. allgemein einer Lösung) erhöht sich die Pheromonkonzentration, sodass dieser in der Folge von weiteren Ameisen benutzt wird. Dies wird auch als autokatalytischer Prozess bezeichnet.
2. *Negatives Feedback*: Dem Ablegen von neuen Pheromonen wirkt der Prozess der Pheromonverdunstung entgegen.
3. *Stochastischer Charakter*: Jede Ameise entscheidet prinzipiell zufällig, wobei die Pheromonwerte nur die Wahrscheinlichkeiten bei der Wahl beeinflussen. Bonabeau hält fest, dass Selbstorganisation auf der Verstärkung von Fluktuationen (Zufälligen Wegen → Random Walks, Fehler, etc.) beruht. Aus Fehlern lernt der Ameisenalgorithmus teilweise mehr als aus der Ausnutzung vorhandener Pheromonwert. Dies liegt an den Ameisen, welche sich auf vermeintlich schlechteren Wegen befinden und dabei auch neue Wege erschließen, welche mit einer z.B. stochastisch basierten Veränderung, zu einer besseren Gesamtlösung führen. Dieses Prinzip ähnelt dem Mutationsprinzip von genetischen Algorithmus, wobei die Stochastik bei einer Lösung während der Konstruktion ggf. mehrfach auftritt.
4. *Multiple Interaktion*: Wenn eine Ameise ihre Umwelt (lokal in Bezug auf ihre Nachbarschaft als auch global bezüglich der Pheromonmatrix) verändert, beeinflusst sie damit indirekt auch das Verhalten anderer Ameisen, welche diesen Umweltzustand dann auswerten und darauf entsprechend reagieren.

Dabei kommt den Mechanismen der Rückkopplung und Stigmergie, dem Informationsaustausch in einem dezentralen System durch Veränderung der lokalen Umgebung, eine besondere Rolle zu. Jede Ameise besitzt zudem, je nach Umsetzung, eine Art Gedächtnis, wodurch die bisher konstruierten Lösungselemente bewertet sowie auf Zulässigkeit geprüft werden. Weitere Details wie die Parametrisierung, die Modellierung des autokatalytischen Prozesses sowie die Unterschiede zwischen den einzelnen ACO Varianten, speziell den Unterschieden des Ant Systems (AS) gegenüber dem Ant Colony System (ACS), sei der*die Leser*in an dieser Stelle auf die entsprechende Fachliteratur verwiesen (vgl. Dorigo und Stützle 2004; Bonabeau et al. 1999).

Zusammenfassung und Fazit

Deepa hält sowohl die Vorteile (vgl. Deepa und Senthilkumar 2016, S. 13)

- Die Anpassungsfähigkeit des ACO (lokal und global) bezüglich veränderlicher Umweltbedingungen.
- Positives Feedback befähigt Ameisen zur Erforschung guter Lösungen.
- Die Robustheit des Verfahrens durch einen populationsbasierten Ansatz.

als auch die entsprechenden Nachteile wie folgt fest:

- Die Pheromonspur und heuristisch besetzte Werte sind stark von der jeweiligen Problemkomplexität abhängig.
- Konvergenz wird garantiert, aber die Zeit bis zum Erreichen der Konvergenz ist ungewiss.
- Anfälligkeit für das Auffinden bzw. den Verbleib in lokalen Lösungen.

Der ACO-Ansatz eignet sich im Vergleich mit genetischen Algorithmen grundsätzlich weniger gut für Auftragsreihenfolgeprobleme, wenn bereits eine relativ gute Ausgangslösung vorliegt (vgl. Kamhuber 2010, S. 117-121). Die Gründe liefern (a) Holland's Schema-Theorem bzw. die Building-Block-Hypothese und (b) die Tatsache, dass durch den konstruktiven Ansatz beim ACO prinzipiell auf Erbinformationen verzichtet wird, weil die komplette Struktur durch jede Ameise vollständig neu generiert wird. Andererseits kann ein ACO sehr schnell konvergieren bzw. eine sehr gute Lösung finden, weil er, im Gegensatz zum GA, sich nicht über viele Generationen hinweg verändern muss, sondern theoretisch bereits die erste Ameise die Chance hat, die optimale Lösung zu ermitteln, weil sie sich die Lösung

komplett neu zusammenstellt (vgl. Kamhuber 2010, S. 109). Somit eignen sich ACO-Varianten v.a. in (Teil-) Problemen, wo sehr viele Änderungsschritte notwendig sind oder keine *gute* Ausgangslösung vorliegt, wofür ein GA vergleichsweise lange bzw. länger benötigt. Trotz des geringeren Einsatzes von ACO, verglichen mit GA, verweist Deepa abschließend auf die aktuellen Anwendungen in Bezug auf verschiedenste Probleme aus dem Bereich der kombinatorischen Optimierung (TSP, PFSP, etc.) (vgl. Deepa und Senthilkumar 2016, S. 16). Dorigo verweist in Bezug auf aktuelle Trends für den ACO auf dynamische Probleme, stochastische Probleme und multikriterielle Probleme (vgl. Dorigo und Stützle 2004, S. 263).

A.8.2 Partikelschwarmoptimierung (PSO)

Partikelschwarmoptimierung, eingeführt von Kennedy und Eberhart 1995 (vgl. Kennedy und Eberhart 1995, S. 1942-1948), basiert auf der Schwarmintelligenz von Vögeln. Die einzelnen Individuen (auch: Partikel, Teilchen) orientieren sich in der klassischen Variante an Position bzw. Verhalten des besten (Alpha-) Individuums \mathbf{P}_g im Schwarm bzw. ihrer eigenen besten Position \mathbf{P}_i und passen sich auf Basis ihrer kognitiven Fähigkeiten dementsprechend im D -dimensionalen Raum an. Jedes Individuum verfügt also über einen individuellen Positionsvektor \mathbf{X}_i als auch einen Geschwindigkeitsvektor \mathbf{V}_i , welche es bei jeder Iteration, entlang jeder Dimension d des Suchraums, entsprechend verändert. Das Geschwindigkeits-Update bestimmt die Richtung und die Distanz, in welche das Individuum fortschreitet (vgl. Boussaïd et al. 2013, S. 99f.):

$$V_{id}(t+1) = V_{id}(t) + C_1\varphi_1(P_{id}(t) - X_{id}(t)) + C_2\varphi_2(P_{gd}(t) - X_{id}(t)), \quad (A.11)$$

wobei $i = 1, 2, \dots, N$ die Partikel bzw. N die Schwarmgröße definiert, während φ_1 und φ_2 zwei stochastisch ermittelte Zufallszahlen im Intervall $[0, 1]$ darstellen. C_1 und C_2 stellen konstante multiplikative Terme dar und werden auch als *Beschleunigungskoeffizienten* bezeichnet. Diese repräsentieren die Attraktion in Bezug auf ihren eigenen Erfolg, also ihre *kognitiven Fähigkeiten*, bzw. den Erfolg ihrer Nachbarn und damit ihre *sozialen Fähigkeiten*. Die Position jedes Partikels ergibt sich aus dem Geschwindigkeits-Update (vgl. Boussaïd et al. 2013, S. 100):

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (A.12)$$

Damit sich die Partikel nicht aus dem Suchraum heraus bewegen können, hat Eberhart einen limitierten Geschwindigkeits-Bereich $[-V_{max}, +V_{max}]$ eingeführt. Die Wahl dieser Schranken bestimmt somit maßgeblich die Balance zwischen Erforschung eines neuen versus Ausnutzung eines vorhandenen Suchraumes. Die Grundstruktur des PSO Algorithmus ist in Algorithmus A.9 in Anlehnung an Boussaïd angegeben und orientiert sich v.a. an den beiden besprochenen Updates bezüglich Position und Geschwindigkeit (vgl. Boussaïd et al. 2013, S. 99-101).

Algorithmus A.9 Partikelschwarmoptimierung (vgl. Boussaïd et al. 2013, S. 100)

```

1: Initialisierung einer Population bestehend aus Teilchen mit zufälligen Positions- und Geschwindigkeitsvektoren im  $D$ -dimensionalen Raum
2: while Abbruchbedingung nicht erfüllt do
3:   for each Partikel  $i$  do
4:     Anpassung der Geschwindigkeit gemäß Glg. (A.11)
5:     Update der Position gemäß Glg. (A.12)
6:     Bewertung der Fitness von  $f(\mathbf{X}_i)$ 
7:     if ( $f(\mathbf{X}_i) < f(\mathbf{P}_i)$ ) then
8:        $\mathbf{P}_i \leftarrow \mathbf{X}_i$ 
9:     end if
10:    if ( $f(\mathbf{X}_i) < f(\mathbf{P}_g)$ ) then
11:       $\mathbf{P}_g \leftarrow \mathbf{X}_i$ 
12:    end if
13:  end for each
14: end while

```

Verglichen mit der Vielzahl an Anwendungen eines GA besteht, gemäß dem No-Free-Lunch-Theorem, kein prinzipiell bedingter Vorteil des PSO gegenüber dem GA. Eine der größten Gefahren besteht, wie auch bei anderen metaheuristischen Verfahren, in der frühzeitigen Konvergenz gegen ein lokales Minimum. Boussaïd erwähnt außerdem, dass es viele erfolgreiche Anwendungen auch in Kombination mit vom GA bekannten genetischen Operatoren gibt. (vgl. Boussaïd et al. 2013, S. 101).

A.8.3 Tabu Search (TS)

Das Tabu Search Verfahren, formalisiert von Glover, greift im Rahmen des Auffindens neuer Lösungen explizit auf die Historie der vergangenen Suchschritte zurück, um einerseits so lokalen Minima zu entkommen und andererseits explorativ zu arbeiten. Das Hauptmerkmal, die *Tabu-Liste* (TL), von wo der Name des Verfahrens herrührt, ist dabei inspiriert vom menschlichen Gedächtnis (vgl. Glover und Kochenberger 2003, S. 37-54). Diese Tabu-Liste speichert die letzten besuchten Lösungen (Züge) und verbietet, dass diese erneut aufgesucht werden. Die Liste agiert als Kurzzeitgedächtnis und drängt den Algorithmus dementsprechend, neue noch nicht besuchte Lösungen in der (ferneren) Nachbarschaft aufzusuchen. Die grundlegenden Schritte des TS (Kurzzeitgedächtnis) sind wie folgt definiert (vgl. Boussaïd et al. 2013, S. 85f.):

1. Selektiere die beste Lösung $s' \in N(s) \setminus TL$.
2. $s \leftarrow s'$.
3. Update TL .

Das kontinuierliche Update der Tabu Liste sowie dessen Länge steuert den Suchprozess. Während ein geringer Listenumfang zur Konzentration der Suche auf einen kleinen Bereich führt, wird eine große Listenlänge die Suche explorativer gestalten. Die Länge der Suche kann während der Optimierung angepasst werden, dies ergibt einen *Reactive Tabu Search Algorithm*. Während ein einfaches TS Verfahren nur über ein Kurzzeitgedächtnis verfügt, werden auch sogenannte Mittel- und Langzeitgedächtnisse implementiert, um die Suche zu intensivieren bzw. diversifizieren. Die Einfügung sogenannter *Aspirationskriterien* kann die Suche verbessern. Der Algorithmus ist, in jeder Variante, grundsätzlich wegen der Liste(n) speicherintensiv und mit einer (lauf-) zeitaufwändigen Überprüfung verbunden. Dafür garantiert es, dass Lösungen, welche vor kurzem besucht worden sind, in den nächsten Schritten nicht wieder angesteuert werden. Die Wahl der Listenlänge ist eine wesentliche und kritische Einstellung. Obwohl das Konzept der Vermeidung von Redundanzen prinzipiell als gut erachtet wird, erscheint die Verwendung von tabuisierenden Listen für den Suchprozess als die nicht am besten geeignete Strategie zu sein. Anstelle einer redundanten Bewertung ist es jedoch prinzipiell bei jedem Verfahren möglich, sich zu einer (jeden) Lösung die dazugehörige Funktionsbewertung zu speichern, um so die Laufzeit auf Kosten von zunehmendem Speicherbedarf zu reduzieren. Eine Tabu-Liste kann dadurch auch bei anderen Verfahren ergänzend eingesetzt werden (vgl. Boussaïd et al. 2013, S. 86; Glover und Kochenberger 2003, S. 37-54)).

A.8.4 Variable Neighborhood Search (VNS)

Das Variable Neighborhood Search Verfahren, vorgestellt von Hansen und Mladenovic, beruht auf der Erforschung von sich dynamisch ändernden Nachbarschaften. Initial werden N Nachbarschaften definiert. Die drei zentralen Schritte des Verfahrens stellen das *Shaking*, gefolgt von Local Search (siehe Alg. 3.2) und einem Move Schritt dar (vgl. Boussaïd et al. 2013, S. 87; Glover und Kochenberger 2003, S. 145-184):

1. Shaking: Auswahl einer zufälligen Lösung s' in der $n - ten$ Nachbarschaft von s .
2. Local Search: Anschließend wird s' als Initiallösung für ein lokales Suchverfahren benutzt, um s'' zu generieren. Wenn s'' besser ist als s , wird zu Schritt 1. zurückgesprungen. Andernfalls zu Schritt 3.

3. Move: Falls s'' schlechter bewertet wird als s , wandert der Algorithmus in die nächste Nachbarschaftsstruktur und setzt danach mit Schritt 1 fort. Abbruch, wenn alle Nachbarschaften durchsucht worden sind.

Das Verfahren wird in der Praxis relativ häufig eingesetzt und oft mit GRASP kombiniert. Prinzipiell können die Nachbarschaften jeweils Elemente einer anderen beinhalten, d.h. $N_1 \in N_2 \in \dots N_{max}$. Dies kann zu einer ineffizienten Suche führen. Anstelle vordefinierter Nachbarschaften existieren auch reaktive Nachbarschaftsstrukturen (vgl. Boussaïd et al. 2013, S. 87).

Auf Grund der mangelnden Verfügbarkeit einer geeigneten Anzahl an vielfältigen, diversifizierten Nachbarschaftsstrukturen (v.a. für die Initialisierung) bietet sich dieses Verfahren im Rahmen dieser Arbeit nicht ideal für die Anwendung an, obwohl es prinzipiell vielversprechend ist.

A.8.5 Iterierte Lokale Suche (ILS)

Die einfache Idee hinter der ILS Metaheuristik besteht darin, ein lokales Optimum auf Basis eines lokalen Suchverfahrens zu finden und darauf die Methode der *Perturbation* anzuwenden, um diesem lokalen Optimum daraufhin idealerweise wieder zu entkommen. ILS generiert sich also die Startlösung der nächsten Iteration durch die Perturbation des zuvor lokal ermittelten Optimums in der aktuellen Iteration. Der Perturbations-Mechanismus ist der Kern vom ILS Verfahren, welches ggf. eine gewisse Ähnlichkeit zum SA Verfahren aufweist. Das Akzeptanzkriterium gewährleistet, gemeinsam mit dem Störmechanismus der Perturbation, einen geeigneten Trade-Off zwischen der Intensivierung und Diversifizierung der Suche. Das Akzeptanzkriterium wird auf verschiedenste Art und Weise implementiert. Im Falle eines better Akzeptanzkriteriums wird von *lokaler Metasuche* gesprochen. Bei Random Walks, welche eine neu erhaltene Lösung, unabhängig von ihrer Bewertung, immer akzeptieren, spricht man von *zufälliger Metasuche*. Bernardino hält im Rahmen seines Beitrags fest, dass sein ILS für große TSP Instanzen die oberen Schranken der Literatur in Bezug auf eine theoretische Instanz (leicht) verbessert (vgl. Bernardino und Paia 2018, S. 460f.; Boussaïd et al. 2013, S. 90; Blum et al. 2011, S. 4136f.; Glover und Kochenberger 2003, S. 321-353).

Das Verfahren ist stark von der konkreten Implementierung abhängig, die Meta-Struktur für dieses einzellösungsbasiertem Verfahren ist relativ schwach ausgeprägt. Deshalb kommt es im Zuge dieser Arbeit nicht in die engere Wahl.

A.9 Konvexe Optimierung

Der Begriff der **konvexen Optimierung** wird nach Boyd bzw. Stein wie folgt definiert (vgl. Stein 2018, S. 40f.; Boyd und Vandenberghe 2004, S. 7):

$$\text{Minimiere } (f_0(\mathbf{x}))$$

$$\text{für alle : } f_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, k,$$

wobei für die Funktionen f_1, \dots, f_k folgender Sachverhalt gilt: Alle Abbildungen gemäß $\mathbb{R}^n \rightarrow \mathbb{R}$ sind konvex, d.h. sie genügen der folgenden Bedingung:

$$f_i(\alpha \mathbf{x} + \beta \mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \quad (A.13)$$

für alle $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ und alle $\alpha, \beta \in \mathbb{R}$, wobei $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$ (vgl. Boyd und Vandenberghe 2004, S. 7). Der Funktionswert an der Konvexkombination zweier Punkte liegt dementsprechend unterhalb der Konvexkombination der Funktionswerte an den beiden Punkten. Ein Optimierungsproblem wird dementsprechend als konvex bezeichnet, wenn sowohl Menge als auch Funktion konvex sind. Darüber hinaus ist auch die Menge aller Minimalpunkte dieses Problems konvex. Jeder lokale Minimalpunkt ist daher auch ein globaler Minimalpunkt dieses Problems. Daher genügt es, bei Vorliegen eines konvexen Problems, lokale Minima zu finden.

Mehrdimensionale erste Ableitungen Für eine nichtleere offene Menge $U \subseteq \mathbb{R}^n$ und dazugehörige Funktion $f : U \rightarrow \mathbb{R}$ wird $\partial x_i f(\bar{\mathbf{x}})$ als partielle Ableitung von f nach x_i an der Stelle $\bar{\mathbf{x}} \in U$ bezeichnet, sofern diese Ableitung existiert. Als erste Ableitung von f an $\bar{\mathbf{x}}$ wird der jeweilige Zeilenvektor betrachtet, während der Spaltenvektor $\nabla f(\bar{\mathbf{x}})$ auch als Gradient von f an $\bar{\mathbf{x}}$ bezeichnet wird. So erhält man z.B. für $f(\mathbf{x}) = x_1^2 + x_2$ als erste Ableitung den Zeilenvektor $Df(\bar{\mathbf{x}}) = \partial x_1 f(\bar{\mathbf{x}}), \dots, \partial x_n f(\bar{\mathbf{x}})$. bzw. konkret $\partial x_1 f(\mathbf{x}) = 2x_1$ sowie $\partial x_2 f(\mathbf{x}) = x_2$. Derart erhält der*die Analyst*in für $f(\mathbf{x})$ als erste Ableitung und Gradient bzw. Spaltenvektor $\nabla f(\bar{\mathbf{x}}) := (Df(\bar{\mathbf{x}}))^T$ an der Stelle $\bar{\mathbf{x}} = (1, -1)^T$ gemäß (Stein 2018, S. 44)

$$Df(\bar{\mathbf{x}}) = (2, 1) \text{ bzw. } \nabla f(\bar{\mathbf{x}}) = \begin{pmatrix} 2 \\ 1 \end{pmatrix}. \quad (\text{A.14})$$

Die Funktion f auf U heißt stetig differenzierbar, wenn ∇f in Bezug auf U existiert und eine stetige Funktion von \mathbf{x} ist. Die Schreibweise ist dann kurz $f \in C^1(U, \mathbb{R})$. In Analogie dazu spricht man von C^0 im Falle einer stetigen bzw. C^{-1} im Falle einer unstetigen Funktion (vgl. Stein 2018, S. 45).

Diese Charakterisierung stellt ein wichtiges Unterscheidungskriterium in Bezug auf die Auswahl eines Optimierungsverfahrens, teils in Abhängigkeit von den Eigenschaften der Zielfunktion, dar (vgl. Meywerk 2007, S. 272).

Die C^1 -Charakterisierung spielt eine zentrale Rolle bei der Untersuchung konvexer Funktionen und besagt, dass auf einer konvexen Menge $X \subseteq \mathbb{R}^n$ eine Funktion $f \in C^1(X, \mathbb{R})$ genau dann konvex ist, wenn gilt (vgl. Stein 2018, S. 47f.):

$$\forall \mathbf{x}, \mathbf{y} \in X : f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \quad (\text{A.15})$$

Optimalitätsbedingungen für unrestringierte konvexe Probleme: Probleme mit offener *zulässiger* Menge werden auch als *unrestringiert* bezeichnet. Das bedeutet, dass bei diesen Problemen keine „Randeffekte“ auftreten. Ein Punkt $\bar{\mathbf{x}} \in \mathbb{R}^n$ heißt kritisch, wenn für $f \in C^1(\mathbb{R}^n, \mathbb{R})$ folgende Bedingung gilt: $\nabla f(\bar{\mathbf{x}}) = 0$ (vgl. Stein 2018, S. 50).

Diese Regel gilt ohne Konvexitätsannahme und ist auch als **notwendige Optimalitätsbedingung** bzw. **Fermat'sche Regel** bekannt (vgl. Stein 2018, S. 50f.):

Wenn der Punkt $\bar{\mathbf{x}} \in \mathbb{R}^n$ lokal minimal ist für $f \in C^1(\mathbb{R}^n, \mathbb{R})$, dann ist $\bar{\mathbf{x}}$ kritischer Punkt von f .

Dementsprechend ist nicht jeder kritische Punkt notwendigerweise lokaler Minimalpunkt einer C^1 Funktion. Gleichzeitig stellt auch nicht jedes lokale Minimum einer allgemeinen (unbeschränkten) Funktion f gleichzeitig ein globales Minimum dar. Abbildung A.3 verdeutlicht diesen Zusammenhang (vgl. Stein 2018, S. 51).

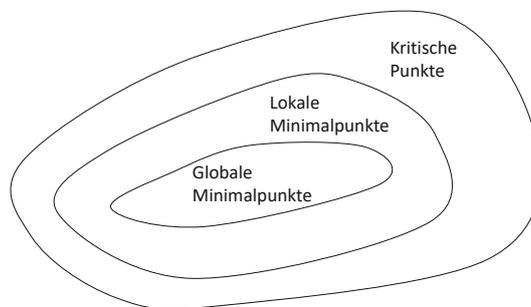


Abb. A.3: Notwendige Optimalitätsbedingung (Quelle: eigene Darstellung, in Anlehnung an Stein 2018, S. 51)

Im Falle des Vorliegens einer konvexen Funktion $f \in C^1(\mathbb{R}^n, \mathbb{R})$ wird die notwendige Optimalitätsbedingung in eine **hinreichende Optimalitätsbedingung** verwandelt. Dann ist jeder kritische Punkt von f gleichzeitig auch globaler Minimalpunkt von f . Stein liefert dafür auch den Beweis. In diesem

Fall genügt es dementsprechend bereits, kritische Punkte zu suchen. Das globale Minimierungsproblem ist damit auf das Lösen der Gleichung $\nabla f(\mathbf{x}) = 0$ zurückgeführt. Das Problem der kleinsten Quadrate bzw. lineare Optimierungsprobleme sind Spezialfälle der klassischen konvexen Optimierung (vgl. Stein 2018, S. 52f.).

A.10 Entwicklung von ERP- und APS-Systemen

Auf Grund der zunehmenden Abhängigkeit der Unternehmen von anderen Partner*innen (sowohl Lieferant*innen als auch Kund*innen) innerhalb der gesamten Lieferkette kommt unternehmensübergreifenden Schnittstellen eine übergeordnete Rolle in der Planung zu. Zur Unterstützung einer möglichst ganzheitlichen Planung einer Lieferkette werden heute Supply-Chain-Management (SCM) verwendet, wobei Lösungen im Bereich der Produktionsplanung- und Steuerung als Advanced Planning and Scheduling (APS)-System bezeichnet werden (vgl. Kurbel 2016, S. 4). MRP/PPS-Systeme gehen von einem Produktionsprogramm (Primärbedarf) aus und ermitteln daraus die erforderlichen Materialbedarfe (Sekundärbedarf). Allerdings fehlen in dieser Betrachtung einerseits die zeitliche Abfolge als auch die konkrete Maschinenbelegungsplanung bzw. allgemeiner die Kapazitätsplanung. Eine Erweiterung stellen gemäß Abb. A.4 sogenannte Manufacturing Resource Planning (MRP II) Systeme dar, welche auch die Zeit- und Kapazitätsplanung unterstützen. Die Produktionsprogrammplanung kann dabei, je nach Variantenspektrum des Unternehmens, zwei Stufen umfassen, nämlich die aggregierte Planung auf Produktgruppenebene als auch die Enderzeugnisebene (vgl. Kurbel 2016, S. 97-164).

Enterprise Resource Planning (ERP)-Systeme basieren auf MRP II bzw. Produktionsplanungs- und Steuerungssystemen, siehe Abb. A.4 (vgl. Hachtel und Holzbaur 2010, S. 111).

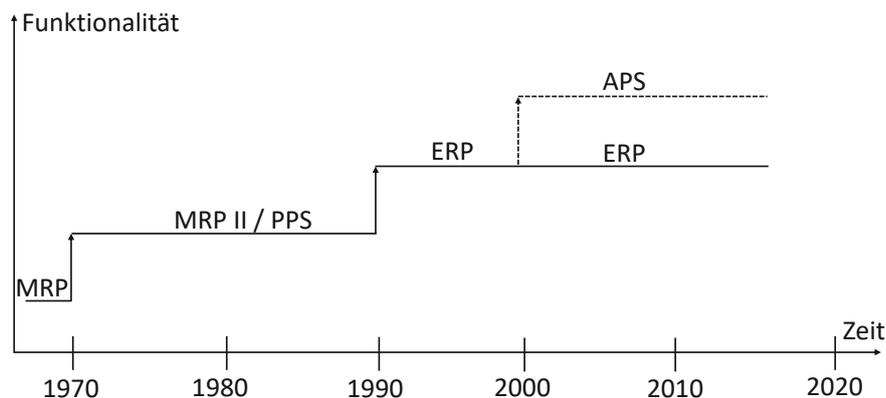


Abb. A.4: Entwicklung von Systemen und Begriffen (eigene Darstellung, angelehnt an Hachtel und Holzbaur 2010, S. 111)

Der ERP Begriff umfasst als Erweiterung zu MRP II auch andere für die Produktion erforderliche Ressourcen. Die größten Probleme bereiten nicht-integrierte Teilsysteme, wodurch Redundanz, Inkonsistenzen bzw. mangelnde Integrität verursacht wird. Advanced Planning and Scheduling (APS)-Systeme setzen mittels Schnittstellen auf ERP-Systeme auf und unterstützen die (überbetriebliche) Planung von Logistikketten mit weiteren Funktionen zur Steuerung und Kontrolle der zugehörigen Abläufe (vgl. Kurbel 2016, S. 203-250).

APS-Systeme sind vor allem durch die folgende Funktionalität gekennzeichnet (vgl. Kurbel 2016, S. 438f.):

- Zusätzliche Modellierungsmöglichkeiten → Unterstützung vielstufiger Stücklisten und Arbeitspläne, Simulationmöglichkeiten (Szenarien)
- Leistungsfähige Algorithmen, z.B. mathematische Optimierung und (meta-) heuristische Verfahren (Engpass-Orientierung)

- Übergreifende Verbindung von Beschaffungs-, Produktions-, Transport-, und Kund*innenaufträgen (Produktionsnetzwerkbetrachtung)

Komponenten eines APS-Systems decken sowohl strategische Aufgaben als auch Aufgaben der operativen Planung ab. Darüber hinaus wird auch die netzwerkbezogene Nachfrageplanung mit den, jeweils angebotsspezifischen, verfügbaren Prognosetechniken bedient. Dementsprechend wird also eine ganzheitliche, unternehmensübergreifende Planung ermöglicht. Das APS-System lässt sich gemäß Koch daher in drei grundlegende Bereiche einteilen (vgl. Koch 2015, S. 266ff.):

1. Netzwerkgestaltung (Strategic Network Design): befasst sich mit unternehmensübergreifenden Entscheidungen.
2. Planung (Supply Chain Planning): mit entsprechenden Planungstätigkeiten versehen.
3. Betrieb (Supply Chain Execution): zur physischen Durchführung von Logistikprozessen.

Die Supply Chain Planning Matrix von Rohde, siehe Abb. A.5, stellt die zentralen kurz-, mittel- und langfristigen Planungsaufgaben eines APS-Systems strukturiert dar. Diese zeigt dabei vor allem die Koordination inklusive den Datenflüssen zwischen den verschiedenen APS Modulen (Stadtler und Kilger 2005, S. 243).

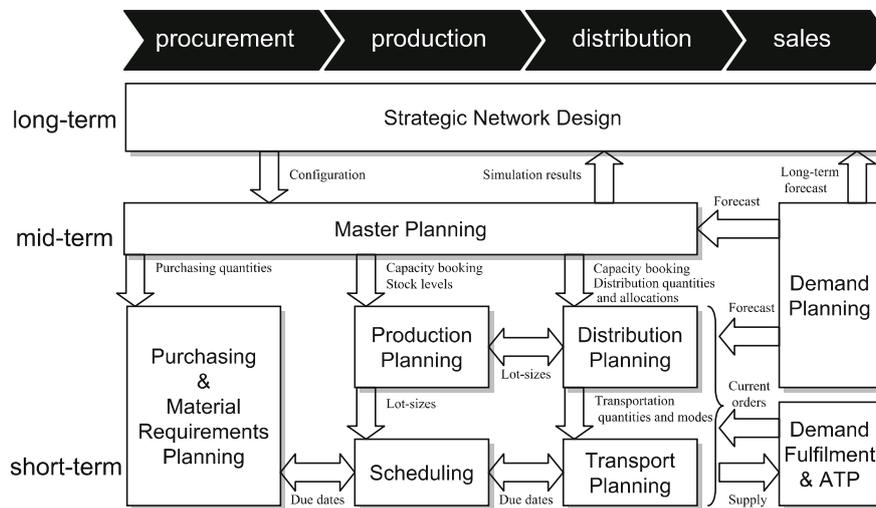


Abb. A.5: Supply Chain Planning Matrix und APS Module (Stadtler und Kilger 2005, S. 243)

Die zentralen Softwaremodule inklusive deren Planungsaufgaben werden in Folge kurz beschrieben (vgl. Stadtler und Kilger 2005, S. 242ff.):

- Strategic Network Design: Langfristplanung für alle vier Kernprozesse.
- Demand Planning: Mittel- und langfristige Absatzplanung.
- Demand Fulfillment & Available to Promise (ATP): Angebot für die kurzfristige Absatzplanung.
- Master Planning: Planung von Beschaffung, Produktion und Distribution für den mittelfristigen Planungshorizont.
- Production Planning and Scheduling: Detaillierte Planung auf Basis von Stammdaten, Kapazitätsbelegungsplanung und Bestandsniveaus auf Losgrößenebene und Lieferterminbasis.
- Transport Planning and Distribution Planning: Transportplanungsmodul.
- Purchasing & Material Requirements Planning: Materialbeschaffung und Stücklistenauflösung.

Ohne an dieser Stelle näher auf die einzelnen Module einzugehen, die von Seiten des*der Anbieter*in (SAP, Siemens, etc.) in unterschiedlicher Form implementiert sind, kommt eine erste Analyse auf Basis des Planungsmoduls von SAP, dem Advanced Planning and Optimization (APO), zu dem folgenden Schluss (vgl. Bothe und Nissen 2003, S. 55ff.; Haarkam 2012, S. 52ff.):

- FMCG branchenspezifisch übliche Aktionen lassen sich schnell und einfach in APO auf Plankundenebene einplanen.
- Gute Eignung der APS Software für die Netzwerkplanung mittels des Supply Network Planning Moduls.
- Grundsätzliche Berücksichtigung und Parametrisierbarkeit von Beständen.
- Losgrößengerechte Planung auf Basis des Standardlosgrößenverfahrens (feste oder exakte Losgröße, spezielle Losgrößenverfahren).
- Material- und Kapazitätsplanung mit Hilfe von (vordefinierten) Planungsheuristiken.
- Verfügbarkeit einer mehrstufigen Verfügbarkeitsplanung, z.B. werden Mischkartons bedarfsgerecht aufgelöst.
- Transportplanung und Fahrzeugterminierung.

A.11 Zufallszahlen

Allgemein wird zwischen echten oder physikalischen Zufallszahlen und scheinbaren Zufallszahlen (*Pseudozufallszahlen*) unterschieden (vgl. Blumenstein 2006, S. 4f.). Während echte Zufallszahlen auf Realisierungen zufälliger Prozesse aus physikalischen Experimenten basieren, werden in der Informatik meistens Pseudozufallszahlen verwendet, die durch sogenannte *Pseudozufallszahlengeneratoren* (PRNG, engl.: *pseudo random number generator*) nach einer bestimmten Erzeugungsvorschrift erstellt worden sind. Streng betrachtet handelt es sich also nicht um Zufallszahlen, weil sie einerseits eine (lange) Periodizität aufweisen und andererseits bei Kenntnis der jeweils angewendeten Methodik in Bezug auf die Erzeugungsvorschrift die weiteren Glieder determiniert, d.h. vorbestimmt sind. Für die Praxis ist die Periodizität i.d.R. genügend umfangreich, und man erhält in kurzer Zeit viele Zufallszahlen schnell und kostengünstig. Eine Zufallszahlenfolge stellt dabei eine Folge voneinander möglichst unabhängiger Zufallszahlen dar, die einer gegebenen explorativen Verteilung genügt (vgl. Hinze 2012, F. 32). Ein gemeinsames Prinzip aller Pseudozufallszahlengeneratoren ist, dass ausgehend von einem Startwert und geeigneter Parameterbesetzungen die Bestimmung durch eine spezifische, rekursive Berechnungsvorschrift erfolgt. Die folgenden grundlegenden Anforderungen, neben der Zufälligkeit einer Zahl, werden an Zufallszahlengeneratoren in der Praxis gestellt (vgl. Hinze 2012, F. 46; Kux 2011, S. 28f.; Blumenstein 2006, S. 7):

- Uniformität: Gleichverteilung der erzeugten Zufallszahlen.
- Statistische Unabhängigkeit: Die erzeugten Zufallszahlen unterliegen keinen aufeinanderfolgenden Wechselbeziehungen bzw. Korrelationen.
- Periodenlänge: Möglichst große Periodenlänge ρ der generierten Zufallszahlen-Folge.
- Effiziente Implementierbarkeit in Bezug auf Geschwindigkeit und Speicherbedarf.

Anwendungsspezifisch soll ein geeigneter Zufallszahlengenerator außerdem eine Reproduzierbarkeit oder Irreproduzierbarkeit aufweisen. Weitere Anforderungen stellen häufig eine genau bekannte Verteilung sowie die Nicht-Vorhersagbarkeit von Zufallszahlen bzw. ihren Folgen dar. Pseudozufallszahlengeneratoren lassen sich üblicherweise in Kongruenzgeneratoren und lineare rückgekoppelte Schieberegister einteilen. Innerhalb der ersten Klasse gibt es sowohl (gemischt) lineare als auch multiplikative sowie nichtlineare Kongruenzverfahren. Außerdem kann man PRNGs anhand ihrer Eigenschaften klassifizieren (vgl. Blumenstein 2006, S. 10):

- Periodizität versus Nicht-Periodizität.
- Einfach versus mehrfach rekursiv.
- Lineare versus nichtlineare Rekursionsvorschrift.

Zufallszahlengeneratoren: Eines der ältesten Verfahren, der Quadratmittengenerator bzw. die *middle-square Methode*, wird heute auf Grund der relativ kurzen Periodenlänge und der damit verbundenen hohen Periodizität nicht mehr verwendet. Die Methode ist auf Grund der früher begrenzten Rechenkapazität bewusst sehr einfach gehalten (vgl. Blumenstein 2006, S. 18f.):

1. Auswahl eines Startwerts x_0 mit $2j$ Stellen ($j \in \mathbb{N}^*$).
2. Erhalt einer neuen Zahl $x_i (i = 1, 2, 3, \dots)$ durch Quadrieren des Vorgängers x_{i-1} und Extraktion der $2j$ mittleren Zahlen.
3. Für eine ungerade Stellenanzahl von $2j$ wird eine Null vorangestellt.

(Gemischt) Lineare Kongruenzgeneratoren, z.B. das Verfahren von Lehmer, oder auch multiplikative Kongruenzmethoden zählen zu den am häufigsten eingesetzten Verfahren. Ein multiplikativer Kongruenzgenerator hat beispielsweise eine Erzeugungsvorschrift für eine Zufallszahl z_i gemäß der Rekursionsvorschrift: $z_i = (a \cdot z_{i-1}) \bmod m$ ($a, m \in \mathbb{N}$ mit $a \geq 2$, $a < m$, $m > 1$) (vgl. Hinze 2012, F. 66ff.). Der gemischt lineare Kongruenzgenerator verwendet im multiplikativen Term zusätzlich noch eine additive Konstante $c \in \mathbb{N}^+$. Die Besonderheit von mehrfach rekursiven Generatoren besteht darin, dass diese ihre Zufallszahlen aus den letzten k Zufallszahlen ($k \in \mathbb{N}$) erzeugen (vgl. Blumenstein 2006, S. 24f.). Die Rekursion lautet somit $z_i = \sum_{k=1}^r a_k \cdot z_{i-k} \bmod m$. Der Fibonacci-Generator ist ein Spezialfall aus dieser Klasse. Weitere Details über Kongruenzverfahren bzw. linear rückgekoppelte Schieberegister sowie entsprechende statistische Tests sind der Literatur zu entnehmen (vgl. Blumenstein 2006, S. 4-61; Hinze 2012, F. 45ff.; Mertens und Bauke 2004, S. 1-4).

A.12 Verteilungsplot

Der Verteilungsplot (siehe Abb. A.6), welcher aus dem Englischen auch als *Boxplot* bekannt ist, wird im Zuge der explorativen Datenanalyse eingesetzt, um statistische Vergleiche von Ergebnisgrößen wie z.B. Stammdatenparametern übersichtlich und statistisch vergleichbar zu visualisieren.

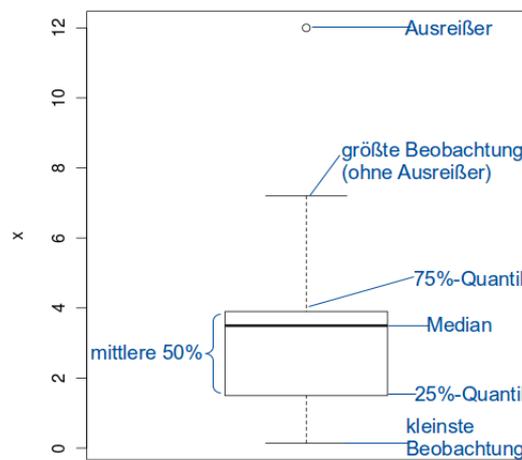


Abb. A.6: Boxplot (Quelle: Keller 2013)

Die zentrale Darstellung bezieht sich auf die Darstellung der Verteilung sämtlicher beobachteten Werte, wodurch ablesbar ist, mit welchen Häufigkeiten diese einzelnen Wertebereiche vertreten sind. Zwecks Übersichtlichkeit werden zusätzliche in der Statistik gebräuchliche Werte (Minimum, Maximum, Median, etc.) sowie diverse Quantile neben dem Zielwert eingezeichnet. Das Quantil ist ein Lagemaß in der Statistik, wobei der Prozentsatz den Schwellwert ausdrückt, wie viele Beobachtungen darunter liegen. Die Bewertung ist gut, wenn die Werte mit hoher Dichte und einer geringen Streuung um den Zielwert auftreten (vgl. Kux 2011, S. 123; Galarnyk 2018).

A.13 Gewichtete-Summen-Methode: Praktisches Anwendungsbeispiel

In der praktischen Anwendung ist es manchmal schwierig eine geeignete Funktion zu erstellen, speziell vorab der Optimierung, wenn Pareto-optimale Lösungen noch nicht bekannt sind. Im Zuge dieser Methodik werden oft verschiedene Maßeinheiten auf einen gemeinsamen Nenner gebracht. Darauf wird daher entsprechend darauf geachtet, selbst wenn eine *Normierung* (englisch: *normalization*) der Teilziele praktiziert wird. Darüber hinaus lassen sich bestimmte Präferenzen teilweise schwer auf einen gemeinsamen Nenner (z.B. Kosten in Form von €) formulieren, weil z.B. nicht sämtliche Kriterien in Form von Kosten vorliegen bzw. ermittelbar sind. Dennoch liegt jederzeit die vollständige Übersicht vor, wie viel (absolut bzw. relativ) jedes einzelne Teilziel bereits verbessert bzw. verschlechtert worden ist. Bereits kleine Veränderungen an der Gewichtung bedeuten ggf. große Änderungen für den Zielfunktionsvektor (vgl. Mietinen Kaisa 2001, S. 6; Freitas 2004, S. 80).

Das folgende Beispiel verdeutlicht die Problematik der vorgestellten Methodik anhand von Tabelle A.1, der Bestimmung eines optimalen Staubsaugerroboters.

Tab. A.1: Maximierungsproblem: Bestimmung des optimalen Staubsaugerroboters (Quelle: eigene Darstellung, in Anlehnung an Hakanen et al. 2013, F. 21)

Modell	Preis	Marke	Design	Funktionsumfang	Lautstärke	Akku	Σ
Robo	1	10	10	10	10	10	7.3
Robi	5	5	5	5	5	5	5
Robu	10	1	1	1	1	1	3.7
Gewichtung	0.3	0.2	0.2	0.1	0.1	0.1	

Die folgenden Probleme treten bei der Anwendung der Methode im Rahmen dieses Beispiels auf (vgl. Hakanen et al. 2013, F. 21):

1. Das aus Perspektive des Gesamtnutzens optimale Modell, Robo, schneidet bei dem wichtigsten Kriterium (in diesem Beispiel: Preis) mit der schlechtesten Bewertung aller drei Modelle ab.
2. Ein intuitiv ansprechender Kompromiss aus allen vorhandenen Zielsetzungen (die sogenannte *goldene Mitte*, Robi) kann durch die Methode nicht als optimales Modell ausgewählt werden, unabhängig von der Wahl der einzelnen Gewichtungen.

Das Problem von möglicherweise fehlenden weiteren notwendigen oder bereits vorhandenen und nicht notwendigen Kriterien tritt überdies auch bei anderen Methoden auf. Außerdem ändern sich ggf. die Kriterien im Laufe der Zeit in ihrer Wichtigkeit oder Ausprägung. Dies ist jedoch nicht abbildbar, weil die Lösungsfindung anhand der aktuellen Kriterien bzw. ihrer Ausprägungen stattfindet. Das Hauptproblem dabei ist nun, dass evtl. *nicht intuitive Lösungen* ausgewählt werden (vgl. Hakanen et al. 2013, F. 20-22).

Dieses Problem lässt sich in der industriellen Praxis zumeist bewältigen, weil die Teilziele dort entweder in Form von Kosten bewertbar vorliegen oder einem gewissen *Schwellwert* genügen. Wenn im obengenannten Beispiel also für den Preis als auch für das Design mindestens eine Bewertung von einer 4 vorliegen muss, kann nur Robi als optimales Modell bestimmt werden, weil es die einzige gültige Lösung darstellt. Die Wahl solcher Schwellwerte ist allerdings vorab nicht immer einfach, sodass bei einer Anhebung der genannten Kriterien von einer 4 auf eine 6 beispielsweise keines der genannten Modelle diese Bewertung erfüllt und keine Lösung auswählbar ist. Daher empfiehlt es sich, diese Schwellwerte im Rahmen der Anwendung einstellbar zu gestalten oder durch Testläufe mit dem*der Anwendungspartner*in vorab zu bestimmen. Eine weitere Möglichkeit stellt die Hybridisierung mit einer weiteren Methode dar, welche diese Schwellwerte im Zuge ihrer Ausführung implizit voraussetzt oder garantiert (vgl. Blum et al. 2011, S. 4135-4151).

A.14 Schema-Theorem: Beispiel

Holland stellt in seiner Arbeit mehrere Argumente für die Funktionsweise von genetischen Algorithmen zur Verfügung, um zu erklären, wie die GA-Suche zu einem impliziten *Sampling* bzw. einer Stichprobenauswahl von Hyperebenen-Partitionen eines Suchraums führt. Jedes Schema beschreibt eine sogenannte *Hyperebene* der Dimension $l - \delta(H)$, siehe Abb. A.7 in Bezug auf einen dreidimensionalen Würfel, wobei jede Lösung eine Ecke dieses Würfels darstellt. Die Ecken werden durch binäre Zeichenketten dargestellt, wobei sich alle benachbarten Ecken durch exakt je ein Zeichen unterscheiden. Ein Problem, das mit drei Genen kodiert ist, wird nun als Würfel mit dem Schema (bzw. der Spezifikation) $H = 000$ im Ursprung dargestellt. So kennzeichnen Lösungsmengen der Schemata $H = 0 * 0$ bzw. $H = *00$ z.B. eine Linie oder Kante, während Mengen der Schemata $H = *1*$ bzw. $H = 0 **$ mit nur einem festgelegten Allel jeweils eine Ebene darstellen. Das Schema $H = ***$ mit keinem festgelegten Allel deckt den gesamten Würfel ab. Jedes Schema entspricht also einer bestimmten Hyperebene im Suchraum. Ein wesentlicher Teil der intrinsischen oder *impliziten Parallelität* des GA ergibt sich aus der Tatsache, dass bei der Auswertung einer Population viele Hyperebenen untersucht werden. Implizite Parallelität bedeutet, dass diese Wettbewerbe zwischen den Hyperebenen simultan und nebeneinander gelöst werden (vgl. Whitley 2007, S.6-9; Whitley und Sutton 2012, S. 649-653; Gerdes et al. 2004, S. 48f.; Bodenhofer 2003, S. 35f.; Holland 2010).

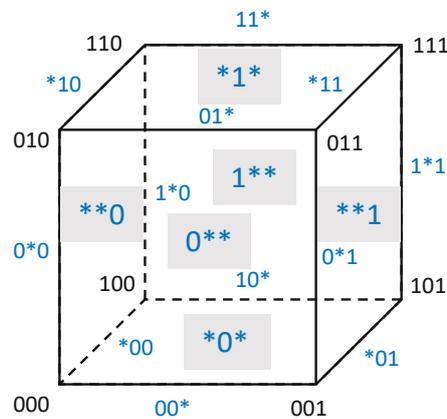


Abb. A.7: Ecken (Spezifikationen) und Kanten sowie Ebenen (jeweils Schemata) eines dreidimensionalen Würfels (Quelle: eigene Darstellung, in Anlehnung an Whitley 2007, S.8)

A.15 Kritik am Schema-Theorem und der Building-Block-Hypothese

Das Schema-Theorem gilt (nur) im Mittel, d.h. für möglichst (unendlich) große Populationen. Dementsprechend liegen i.d.R. statistische Abweichungen, bedingt z.B. durch die Stochastik bzw. den Zufallscharakter der drei genetischen Operatoren (Selektion, Rekombination und Mutation) vor. Die durchschnittliche Fitness stellt die mittlere Bewertung der aktuellen Population, jedoch nicht die mittlere Bewertung aller Individuen, dar. Außerdem sind nicht alle Schemata in der (jeweils aktuellen) Population vertreten bzw. sind die vertretenen Schemata nicht unabhängig voneinander. Zur Umgehung des Problems der mittleren Bewertung aus dem genannten Kritikpunkt wurde eine weitere Hypothese aufgestellt. Diese besagt, dass bei einer vorliegenden Aufteilung des Suchraums durch eine Hyperebenenpartition mit geringer Ordnung zu erwarten ist, dass der GA zu der Hyperebene mit der besten *statischen durchschnittlichen Fitness*² konvergiert. Wie die ursprüngliche Building-Block-Hypothese hat auch die statische Building-Block-Hypothese zwei Kritikpunkte: Einerseits verliert die Hypothese ihre Gültigkeit, wenn die Population bereits zu konvergieren beginnt, andererseits ist die in einem Schema immanente Varianz ein Problem (vgl. Gerdes et al. 2004, S. 55f.; Whitley und Sutton 2012, S. 653-655).

²Die statische, durchschnittliche Fitness beschreibt die Fitness in Bezug auf den gesamten Suchraum, d.h. über die Population hinweg.

A.16 Ausschnitt aus dem Quellcode (Zielfunktionsberechnung in Java)

```

/**
 * Diese Methode berechnet den aktuellen Zielfunktionswert auf Basis der
 * übergebenen Parameter. Aus dem übergebenen Chromosom wird der
 * aktuelle Bestandsverlauf benötigt. Der Zielfunktionswert besteht aus:
 * (1) Werks-Gradient, (2) Artikel-Gradient, (3) Bestands-Gradient
 * (4) Raumauslastungs-Gradient, bestehend aus 5 Raumgruppen-Gradienten und
 * (5) Mindestanzahl erforderlicher Rüstvorgänge im Planungshorizont
 * @param zeilen
 * @param stammdaten
 * @param args
 * @param init
 * @param rückgabe
 * @param bestand_init
 * @param raumauslastung
 * @param chr_akt
 * @return resultRating
 */
private static ArrayList<Double> GA_evaluateTargetFunction(Zeilen zeilen, Stammdaten stammdaten, String[] args,
String[] init, ap_pp_ges rückgabe, ArrayList<Double[]> bestand_init, Raumauslastung raumauslastung, Chromosome
chr_akt) {

    double produktionsArtikelK_Abs = 0.0;
    double bestandsK_Abs = 0.0;
    ArrayList<Double> resultRating = new ArrayList<Double>();
    Double[][] initial0Speck_HT3_intern = new Double[summenKW][Produktionsartikel.size()];
    Double[][] initial1Speck_HT3_intern = new Double[summenKW][Produktionsartikel.size()];
    // Matrizen aktualisieren: Je Produktionsartikel die Wochensummen bilden, einmal für initial0 & initial1, SPECK_HT3
    initial0Speck_HT3_intern = writeInitial0(zeilen, Produktionsartikel, initial0Speck_HT3);
    initial1Speck_HT3_intern = writeInitial1(zeilen, Produktionsartikel, initial1Speck_HT3);
    // tonnageSpeckProKW aktualisieren
    Double[] tonnageSpeckHT3ProKW = new Double[summenKW];
    tonnageSpeckHT3ProKW = berechneTonnage(initial0Speck_HT3_intern, initial1Speck_HT3_intern);
    // (1) Teilziel: globalen Werks-Gradienten berechnen
    double werksGradient = 0.0;
    // Aktualisierung Werksgradient im Rahmen der Evaluierung
    for(int i = 0; i < tonnageSpeckHT3ProKW.length - 1; i++) {
        double tonnage1 = tonnageSpeckHT3ProKW[i]; double tonnage2 = tonnageSpeckHT3ProKW[i+1];
        // dynamischer Abs.-Werksgradient pro Periode
        werksGradient = werksGradient + Math.abs(tonnage2 - tonnage1);
    }
    resultRating.add(werksGradient);
    // ProductRowMap0,1 beschreiben über initial 0,1: Key = Product, Values:= KW
    HashMap<Integer, List<Double>>productRowMap0SPECK_HT3 = new HashMap<Integer, List<Double>>();
    HashMap<Integer, List<Double>>productRowMap1SPECK_HT3 = new HashMap<Integer, List<Double>>();

```

```

productRowMap0SPECK_HT3 = (HashMap<Integer, List<Double>>) writeProductRowMap(initial0Speck_HT3_intern,
Produktionsartikel);

productRowMap1SPECK_HT3 = (HashMap<Integer, List<Double>>) writeProductRowMap(initial1Speck_HT3_intern,
Produktionsartikel);

ArrayList<Double> list0 = new ArrayList<Double>(); ArrayList<Double> list1 = new ArrayList<Double>();
int rüstvorgänge = 0;
// (2) Teilziel: Absolutes Produktionsartikel-K berechnen
for(int i = 0; i < Produktionsartikel.size(); i++) {
    int artikel = Produktionsartikel.get(i);
    list0 = (ArrayList<Double>) productRowMap0SPECK_HT3.get(artikel);
    list1 = (ArrayList<Double>) productRowMap1SPECK_HT3.get(artikel);
    ArrayList<Double> list = new ArrayList<Double>();
    // Artikel-list mit Gewichten berechnen
    for(int j = 0; j < list0.size(); j++) {
        double qty = 0; qty = list0.get(j) + list1.get(j);
        list.add(qty);
        // Mindest-Rüstaufwand in dieser Woche bewerten: Positive Mengen mit 1, Nullmengen mit 0 Rüstaufwand
        if(qty > 0.1) { rüstvorgänge++; }
    }
    // sub Artikel k berechnen
    for(int m = 0; m < list.size() - 1; m++) {
        double weekBefore = list.get(m);
        double weekAfter = list.get(m+1);
        double delta = Math.abs(weekBefore - weekAfter);
        produktionsArtikelK_Abs = produktionsArtikelK_Abs + delta;
    }
}
resultRating.add(produktionsArtikelK_Abs);
// (3) Teilziel: Bestandslevel-Gradient
// Bestands-K aktualisiert berechnen ab Step 0
for(int i = 0; i < Produktionsartikel.size(); i++) {
    int artikel = Produktionsartikel.get(i);
    // Smax für jeden Artikel einmalig besorgen
    int index = stammdaten.getIndex(artikel); double Smax = stammdaten.getMaxPuffer(index - 1);
    // Bestandsverlauf abrufen
    Double[] list = chr_akt.getBestandsverlauf().get(artikel);
    // sub Artikel k berechnen
    for(int m = 0; m < list.length - 1; m++) {
        double week = list[m];
        // Das Bestands-Delta ist in jeder Periode der Abstand zum Soll-Bestand (~ S_max)
        double delta = Math.abs(week - Smax);
        bestandsK_Abs = bestandsK_Abs + delta;
    }
}
}

```

```

resultRating.add(bestandsK_Abs);
// (4) Teilziel: 5-dimensionaler Raumkapazitäts-Gradient
// Der Raumauslastungsgradient wird über die Raumprozent-Auslastung aktualisiert bewertet
double[][] raumprozent = raumauslastung.raumprozent;
double k1 = 0.0; double k2 = 0.0; double k3 = 0.0; double k4 = 0.0; double k5 = 0.0;
int size = raumprozent.length - 1;
// Wenn bei 0 begonnen wird, wird die aktuelle Bestandssituation mitbewertet
for(int i = 0; i < size; i++) {
    k1 = k1 + Math.abs(raumprozent[i+1][0] - raumprozent[i][0]);
    k2 = k2 + Math.abs(raumprozent[i+1][1] - raumprozent[i][1]);
    k3 = k3 + Math.abs(raumprozent[i+1][2] - raumprozent[i][2]);
    k4 = k4 + Math.abs(raumprozent[i+1][3] - raumprozent[i][3]);
    k5 = k5 + Math.abs(raumprozent[i+1][4] - raumprozent[i][4]);
}
Double raumauslastungsGradient_gesamt = k1 + k2 + k3 + k4 + k5;
resultRating.add(raumauslastungsGradient_gesamt);
// (5) Teilziel: Anzahl minimal erforderlicher Rüstvorgänge (Aufwände) bewerten
resultRating.add((double) rüstvorgänge);
// resultRating in resultRatings updaten
resultRatings.put(resultRatings.size(), resultRating);
// first Rating ever? wird nur das erste Mal beschrieben
if (firstRating.size() == 0) {
    firstRating = resultRating;
    bestRating = resultRating;
}
return resultRating;
}

```

A.17 Wissenschaftliche Publikationen mit konkretem Bezug zum Thema dieser Dissertation

Im Zuge des Entstehungsprozesses dieser Dissertation wurden, in Einklang mit der *Research Contributions* Richtlinie gemäß Design Science, mehrere wissenschaftliche Beiträge veröffentlicht, die im direkten oder indirekten Zusammenhang mit der Produktionsproblembeschreibung bzw. der methodischen Entwicklung (Kapitel 4 – 5), den dazugehörigen Forschungsergebnissen (Kapitel 6) sowie Conclusio und Ausblick (Kapitel 7) aus dieser Arbeit stehen. Dieser Teil des Anhangs zeigt den zeitlichen Zusammenhang der jeweiligen Veröffentlichung mit der Dissertation auf. Die tabellarische Zusammenstellung auf Tab. A.2 ist zeitlich absteigend sortiert und umfasst zwei Journal-Veröffentlichungen (vgl. Kamhuber et al. 2020, S. 1-16; Sihm et al. 2018, S. 447-450) sowie drei Publikationen auf internationalen Konferenzen (vgl. Sobottka et al., 2017, S. 3495-3506; Kamhuber et al. 2018, S. 101-116; Kamhuber et al. 2019, S. 2108-2118;), wovon zwei IEEE³ Konferenzen repräsentieren.

³Institute of Electrical and Electronics Engineers

Tab. A.2: Publikationsverzeichnis des Autors (Quelle: eigene Darstellung)

Titel	Autoren	Journal/Konferenz	Zusammenhang mit der Dissertation
An efficient hybrid multi-criteria optimization approach for rolling production smoothing of a European food manufacturer	Felix Kamhuber, Thomas Sobottka, Bernhard Heinzl, Jan Henjes, Wilfried Sihm	Computers & Industrial Engineering Volume 147, September 2020	Der methodisch umfangreichste und wertvollste Beitrag bzw. Feedback für den Hauptautor. Durch die Forderung eines algorithmischen Vergleichs von einem Reviewer wurde ein mit dem SA gleichwertiger GA entwickelt und mit dem SA hinsichtlich Lösungsqualität verglichen. Diese Veröffentlichung betrachtet schwerpunktmäßig die hybride Optimierung sowie den metaheuristisch-algorithmischen Vergleich inklusive Laufzeitkomplexität.
An Efficient Multi-Objective Hybrid Simheuristic Approach for Advanced Rolling Horizon Production Planning	Felix Kamhuber, Thomas Sobottka, Bernhard Heinzl, Wilfried Sihm	Proceedings of the 2019 Winter Simulation Conference	Diese Veröffentlichung beleuchtet die prinzipiell vollständig entwickelte, rollierende Optimierungsmethodik bestehend aus Heuristik und SA sowie integrierter Simulation. Außerdem wird die Methodik mit Bezugnahme auf die statistische Datenanalyse publiziert.
An Innovative Heuristic Mixed-Integer Optimization Approach for Multi-Criteria Optimization based Production Planning in the context of Production Smoothing	Felix Kamhuber, Thomas Sobottka, Peter Schieder, Maximilian Ulrich, Wilfried Sihm	Proceedings of META'2018	Diese Veröffentlichung demonstriert die Details des entwickelten, lexikographisch rollierend planenden, heuristischen Optimierungsverfahrens.
Interdisciplinary Multi-criteria Optimization using Hybrid Simulation to pursue Energy Efficiency through Production Planning	Wilfried Sihm, Thomas Sobottka, Bernhard Heinzl, Felix Kamhuber	CIRP Annals Volume 67, Issue 1, 2018	Diese Veröffentlichung diskutiert eine entwickelte, hybride Simulationsmethodik, die anhand eines industriellen Fallbeispiels mit einer gewichteten Zielfunktion mittels eines GA optimiert worden ist. Der Zusammenhang zur Dissertation besteht in der mehrstufigen Ablauflogik der Optimierung.
A case study for simulation and optimization based planning of production and logistics systems	Thomas Sobottka, Felix Kamhuber, Jan Henjes, Wilfried Sihm	Proceedings of the 2017 Winter Simulation Conference	Diese Veröffentlichung zeigt die Fallstudie ganzheitlich in Bezug auf die Planung der Fabrik- und Produktionsanlagen. In diesem Zusammenhang wird auch die Produktionsglättungsheuristik in groben Zügen zum Einsatz in der Fabrikplanung als statisches Planungstool vorgestellt.

A.18 Lebenslauf des Autors

Auf Abb. A.7 ist der Lebenslauf des Autors der vorliegenden Arbeit ersichtlich. In diesem Zusammenhang ist die Berufserfahrung als Absatzplaner bei einem frz. Konzern aus der FMCG Branche erwähnenswert. Im Zuge der Berufspraxis hat der Autor die Thematik und Herausforderungen bei der Berücksichtigung von Aktions- und Kurantmengen im Zuge der Absatzplanung mit SAP-APO erlebt.

Curriculum vitae

Dipl.-Ing. Felix Kamhuber, BSc

1.) Persönliche Daten: Felix Kamhuber

Wohnhaft: Anton-Baumgartner-Straße 44, C8/102, A-1230 Wien
 Geburtsdatum: 18.9.1985 in Wien
 Mobiltelefon: 0650/5015504
 E- Mail: atreju85@hotmail.com



Wien, am 15.06.2022

2.) Ausbildung

18.8.2022	Rigorosum: Geplanter Abschluss des Doktoratsstudiums: Dr. techn.
10/2018 – 08/2022	Absolvierung des 6 – semestrigen Dokoratsstudiums Wirtschaftsingenieurwesen-Maschinenbau (TU Wien)
11.3.2010	Masterprüfung: Abschluss des Masterstudiums: Diplom-Ingenieur
10/2008 – 03/2010	Absolvierung des 3 – semestrigen Masterstudiums Industrielogistik Schwerpunkte: IT, Automation, Logistik- Management
11.6.2008	Bachelorprüfung: Abschluss des Bachelorstudiums: Bachelor of Science
10/2004 – 06/2008	Absolvierung des 7 – semestrigen Bachelorstudiums Industrielogistik an der Montanuniversität Leoben Schwerpunkte: Ingenieur- und Naturwissenschaften, Beschaffungs-, Produktions-, Distributions-, Transportlogistik, Informationstechnologien
06/2003	Maturaabschluss mit ausgezeichnetem Erfolg
09/1995 – 06/2003	AHS Maroltingergasse in 1160 Wien

3.) Berufserfahrung

Seit 10/2014	Wissenschaftlicher Mitarbeiter: Fraunhofer Austria Research GmbH Aufgaben: Bearbeitung und Leitung von Industrie- und Forschungsprojekten im Fachbereich Simulation, Operations Research und Optimierung; Branchen: FMCG, Kunststoff- und metallverarbeitende Industrie
06/2010 – 05/2014	Demand- and Supply Planner: Bongrain GmbH (Wien) Aufgaben: Absatzplanung/Absatzanalyse FMCG mit SAP-DP/SNP/APO, Bestandsverwaltung, Co-Packing, Lagerbevorratung, SC & KPI Reporting
07/2009 – 01/2010	Diplomand: Profactor Produktionsforschungs GmbH (Steyr) Masterarbeit: „Simulationsbasierte metaheuristische Optimierung eines realen Auftragsreihenfolgeproblems“ Tätigkeit: Software- und Algorithmenentwicklung mit Eclipse in Java
08/2008 – 09/2008	Praktikant: Novomatic Gruppe, AGI GmbH (Gumpoldskirchen) Aufgaben und Tätigkeiten in der Qualitätssicherung: Erstellung von Platinenmappen in der Halbleiterprintfertigung, Mitarbeit bei der Programmierung einer automated optical inspection Anlage
07/2007 – 11/2007	Praktikant: Wiener Hafen GmbH & Co KG sowie bei der Tochtergesellschaft WienCont Management GmbH Aufgaben: Ist- und Soll-Prozesserstellung im Zuge der Einführung eines neuen EDV Stellplatzsystems bei WienCont
10/2003 – 09/2004	Tätigkeit im Zivildienst für Kinder mit Behinderung im SPZ Breitensee

Abb. A.8: Lebenslauf Felix Kamhuber

Literaturverzeichnis

- [Abele et al. 2008] ABELE, Eberhard ; RUMPEL, Guido ; KUHN, Sven M.: Flexible Produktionskonzepte für die saisonale Produktion. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 103 (2008), Nr. 9, S. 585–588. <http://dx.doi.org/10.3139/104.101328>. – DOI 10.3139/104.101328. – ISSN 0947–0085
- [Absi und Kedad-Sidhoum 2007] ABSI, Nabil ; KEDAD-SIDHOUM, Safia: MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs. In: *RAIRO - Operations Research* 41 (2007), Nr. 2, S. 171–192. <http://dx.doi.org/10.1051/ro:2007014>. – DOI 10.1051/ro:2007014. – ISSN 1290–3868
- [Al-Ashhab 2016] AL-ASHHAB, M. S.: An Optimization Model for Multi-period Multi-Product Multi-objective Production Planning. In: *International Journal of Engineering & Technology* 16 (2016), Nr. 01, 43–56. http://ijens.org/Vol_16_I_01/164601-7575-IJET-IJENS.pdf
- [Altenberg 1995] ALTENBERG, Lee: The Schema Theorem and Price's Theorem. Version: 1995. <http://dx.doi.org/10.1016/B978-1-55860-356-1.50006-6>. In: WHITLEY, L. D. (Hrsg.) ; VOSE, MICHAEL D. (Hrsg.): *Foundations of Genetic Algorithms* Bd. 3. Elsevier, 1995. – DOI 10.1016/B978-1-55860-356-1.50006-6. – ISBN 1081–6593, 23–49
- [Amous et al. 2008] AMOUS, Sawsan K. ; LOUKIL, Taicir ; ELAOU, Semya ; DHAENENS, Clarisse: A new genetic algorithm for the travelling salesman problem. In: *International Journal of Pure and Applied Mathematics* 2008 (2008), Nr. Volume 48, No. 2, S. 1–16
- [Ansari et al. 2017] ANSARI, Fazel ; SCHENKELBERG, Kai ; SEIDENBERG, Ulrich ; MADJID, Fathi: Problem Solving in the Digital World: Synoptic Formalism, Incrementalism and Heuristics. 2nd Edition (2016) (2017), S. 1–9
- [Arnold et al. 2008] ARNOLD, Dieter ; ISERMANN, Heinz ; KUHN, Axel ; TEMPELMEIER, Horst ; FURMANS, Kai: *Handbuch Logistik*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. <http://dx.doi.org/10.1007/978-3-540-72929-7>. <http://dx.doi.org/10.1007/978-3-540-72929-7>. – ISBN 978–3–540–72928–0
- [Austin 2020] AUSTIN, Lee D. ; STACKOVERFLOW.COM (Hrsg.): *What are fitness sharing and niche count in evolutionary computation?* <https://stackoverflow.com/questions/37836751/what-are-fitness-sharing-and-niche-count-in-evolutionary-computation/38174559> Bildquelle: <https://i.stack.imgur.com/yLJMv.png>. Version: 2020. Zuletzt geprüft am: 11.06.2020
- [Banzi et al. 2012] BANZI, Adam S. ; NOBRE, Tiago ; PINHEIRO, Gabriel B. ; ARIAS, Joao Carlos G. ; POZO, Aurora ; VERGILIO, Silvia R.: Selecting mutation operators with a multiobjective approach. In: *Expert Systems with Applications* 39 (2012), Nr. 15, S. 12131–12142. <http://dx.doi.org/10.1016/j.eswa.2012.04.041>. – DOI 10.1016/j.eswa.2012.04.041. – ISSN 09574174
- [Baresel et al. 2002] BARESEL, André ; STHAMER, Harmen ; SCHMIDT, Michael: Fitness function design to improve evolutionary structural testing. In: *GECCO'02: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation* (2002), 1329–1336. <https://dl.acm.org/doi/10.5555/2955491.2955736>
- [Bautista et al. 1996] BAUTISTA, J. ; COMPANYS, R. ; COROMINAS, A.: Heuristics and exact algorithms for solving the Monden problem. In: *European Journal of Operational Research* 88 (1996), Nr. 1, S. 101–113. [http://dx.doi.org/10.1016/0377-2217\(94\)00165-0](http://dx.doi.org/10.1016/0377-2217(94)00165-0). – DOI 10.1016/0377–2217(94)00165–0. – ISSN 03772217

- [Beer 2014] BEER, Albert: *Der Bullwhip-Effekt in einem komplexen Produktionsnetzwerk: Entwicklung eines realitätsadäquaten Simulationsmodells in Anlehnung an eine Realbeispiel und Quantifizierung der Wirksamkeit von Maßnahmen gegen den Bullwhip-Effekt: Zugl.: Duisburg-Essen, Univ., Diss., 2014*. Wiesbaden : Springer Fachmedien Wiesbaden GmbH, 2014 (Springer-Gabler Research). – ISBN 978–3–658–06117–3
- [Bergeron et al. 2009] BERGERON, D. ; JAMALI, M. A. ; AIT-KADI, D.: Coupling Simulation and Optimization for an Integrated Production and Maintenance Planning in Manufacturing Systems. In: *IFAC Proceedings Volumes* 42 (2009), Nr. 4, S. 1091–1095. <http://dx.doi.org/10.3182/20090603-3-RU-2001.0549>. – DOI 10.3182/20090603-3-RU-2001.0549. – ISSN 14746670
- [Bernardino und Paias 2018] BERNARDINO, Raquel ; PAIAS, Ana: Solving the family traveling salesman problem. In: *European Journal of Operational Research* 267 (2018), Nr. 2, S. 453–466. <http://dx.doi.org/10.1016/j.ejor.2017.11.063>. – DOI 10.1016/j.ejor.2017.11.063. – ISSN 03772217
- [Blum et al. 2011] BLUM, Christian ; PUCHINGER, Jakob ; RAIDL, Günther R. ; ROLI, Andrea: Hybrid metaheuristics in combinatorial optimization: A survey. In: *Applied Soft Computing* 11 (2011), Nr. 6, S. 4135–4151. <http://dx.doi.org/10.1016/j.asoc.2011.02.032>. – DOI 10.1016/j.asoc.2011.02.032. – ISSN 15684946
- [Blumenstein 2006] BLUMENSTEIN, Dirk: *Untersuchung und effiziente Implementierung von Algorithmen zur Erzeugung von Pseudozufallszahlen*. Dresden, TU Dresden, Diplomarbeit, 2006
- [Bodenhofer 2003] BODENHOFER, Ulrich: *Genetic Algorithms: Theory and Applications: Lecture Notes*. <https://www.fl11.jku.at/div/teaching/Ga/GA-Notes.pdf>. Version: 2003.
- [Boelter 2013] BOELTER, Tobias: *K-Opt und die Lin-Kernighan-Heuristik für das allgemeine TSP*. Köln, Universität zu Köln, Seminararbeit, 2013
- [Bonabeau et al. 1999] BONABEAU, Eric ; THERAULAZ, Guy ; DORIGO, Marco: *Swarm intelligence: From natural to artificial isystems*. New York : Oxford University Press, 1999 <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=144023>. – ISBN 0–19–513158–4
- [Bothe und Nissen 2003] BOTHE, Matthias ; NISSEN, Volker: *SAP APO in der Praxis*. Wiesbaden : Vieweg+Teubner Verlag, 2003. <http://dx.doi.org/10.1007/978-3-322-90812-4>. <http://dx.doi.org/10.1007/978-3-322-90812-4>. – ISBN 978–3–528–05833–3
- [Boussaïd et al. 2013] BOUSSAÏD, Ilhem ; LEPAGNOT, Julien ; SIARRY, Patrick: A survey on optimization metaheuristics. In: *Information Sciences* 237 (2013), S. 82–117. <http://dx.doi.org/10.1016/j.ins.2013.02.041>. – DOI 10.1016/j.ins.2013.02.041. – ISSN 00200255
- [Boyd und Vandenberghe 2004] BOYD, Stephen ; VANDENBERGHE, Lieven: *Convex Optimization*. Cambridge : Cambridge University Press, 2004. <http://dx.doi.org/10.1017/CB09780511804441>. <http://dx.doi.org/10.1017/CB09780511804441>. – ISBN 9780511804441
- [Branke et al. 2008] BRANKE, Jürgen ; DEB, Kalyanmoy ; MIETTINEN, Kaisa ; SŁOWIŃSKI, Roman: *Multiobjective Optimization*. Bd. 5252. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. <http://dx.doi.org/10.1007/978-3-540-88908-3>. <http://dx.doi.org/10.1007/978-3-540-88908-3>. – ISBN 978–3–540–88907–6
- [Breunig 1995] BREUNIG, Franz: *Mathematische Grundlagen: Übersicht: Gradient, Divergenz und Rotation*. <https://www.cg.tuwien.ac.at/research/vis/seminar9596/1-math/grad.html>. Version: 1995
- [Bundeswettbewerbsbehörde 2014] BUNDESWETTBEWERBSBEHÖRDE ; BUNDESWETTBEWERBSBEHÖRDE (Hrsg.): *Standpunkt zu vertikalen Preisbindungen der BundesWettbewerbsBehörde*. https://www.bwb.gv.at/fileadmin/user_upload/PDFs/BWB-Leitfaden%20-%20Standpunkt%20zu%20vertikalen%20Preisbindungen.pdf. Version: 2014. Zuletzt geprüft am: 18.10.2020

- [Bundeswettbewerbsbehörde 2018] BUNDESWETTBEWERBSBEHÖRDE ; BUNDESWETTBEWERBSBEHÖRDE (Hrsg.): *Kartelle und Marktmachtmissbrauch: Entscheidungen*. <https://www.bwb.gv.at/KartelleUndMarktmachtmissbrauch/Entscheidungen/Seiten/default.aspx>. Version: 2018. Zuletzt geprüft am: 14.01.2018
- [Campos-Ciro et al. 2016] CAMPOS-CIRO, Guillermo ; DUGARDIN, Frédéric ; YALAOUI, Farouk ; KELLY, Russell: A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints. In: *IFAC-PapersOnLine* 49 (2016), Nr. 12, S. 1272–1277. <http://dx.doi.org/10.1016/j.ifacol.2016.07.690>. – DOI 10.1016/j.ifacol.2016.07.690. – ISSN 24058963
- [Caumont et al. 2009] CAUMOND, A. ; LACOMME, P. ; MOUKRIM, A. ; TCHERNEV, N.: An MILP for scheduling problems in an FMS with one vehicle. In: *European Journal of Operational Research* 199 (2009), Nr. 3, S. 706–722. <http://dx.doi.org/10.1016/j.ejor.2008.03.051>. – DOI 10.1016/j.ejor.2008.03.051. – ISSN 03772217
- [Celano et al. 2004] CELANO, Giovanni ; COSTA, Antonio ; FICHERA, Sergio: A comparative analysis of sequencing heuristics for solving the Toyota Goal Chasing problem. In: *Robotics and Computer-Integrated Manufacturing* 20 (2004), Nr. 6, S. 573–581. <http://dx.doi.org/10.1016/j.rcim.2004.07.009>. – DOI 10.1016/j.rcim.2004.07.009. – ISSN 07365845
- [Céline Gicquel et al. 2008] CÉLINE GICQUEL ; MICHEL MINOUX ; YVES DALLERY: *Capacitated Lot Sizing models: a literature review*. 2008. <https://hal.archives-ouvertes.fr/hal-00255830>
- [Chang et al. 2002] CHANG, Pei-Chann ; HSIEH, Jih-Chang ; LIN, Shui-Geng: The development of gradual-priority weighting approach for the multi-objective flowshop scheduling problem. In: *International Journal of Production Economics* 79 (2002), Nr. 3, S. 171–183. [http://dx.doi.org/10.1016/S0925-5273\(02\)00141-X](http://dx.doi.org/10.1016/S0925-5273(02)00141-X). – DOI 10.1016/S0925-5273(02)00141-X. – ISSN 09255273
- [Chiandussi et al. 2012] CHIANDUSSI, G. ; CODEGONE, M. ; FERRERO, S. ; VARESI, F. E.: Comparison of multi-objective optimization methodologies for engineering applications. In: *Computers & Mathematics with Applications* 63 (2012), Nr. 5, S. 912–942. <http://dx.doi.org/10.1016/j.camwa.2011.11.057>. – DOI 10.1016/j.camwa.2011.11.057. – ISSN 08981221
- [Chu und Beasley 1997] CHU, P. C. ; BEASLEY, J. E.: A genetic algorithm for the generalised assignment problem. In: *Computers & Operations Research* 24 (1997), Nr. 1, S. 17–23. [http://dx.doi.org/10.1016/S0305-0548\(96\)00032-9](http://dx.doi.org/10.1016/S0305-0548(96)00032-9). – DOI 10.1016/S0305-0548(96)00032-9. – ISSN 03050548
- [Danet 2013] DANET, Christian-Paul: *Classical maximum principle. some extensions and applications*. Saarbrücken, Deutschland : LAP LAMBERT Academic Publishing, 2013. – ISBN 978-3-659-40556-3
- [Dangelmaier 2009] DANGELMAIER, Wilhelm: *Theorie der Produktionsplanung und -steuerung: Im Sommer keine Kirschpralinen?* Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2009 (VDI-Buch). <http://dx.doi.org/10.1007/978-3-642-00633-3>. <http://dx.doi.org/10.1007/978-3-642-00633-3>. – ISBN 978-3-642-00632-6
- [Deb et al. 2014] DEB, Suash (Hrsg.) ; SERGUIIEVA, Antoaneta (Hrsg.) ; FONG, Simon (Hrsg.): *2014 2nd International Symposium on Computational and Business Intelligence - ISCBI 2014: 7-8 December 2014, New Delhi, India : proceedings*. Piscataway, NJ : IEEE, 2014 . – ISBN 978-1-4799-7551-8
- [Debnath und Mourshed 2018] DEBNATH, Kumar B. ; MOURSHED, Monjur: Forecasting methods in energy planning models. In: *Renewable and Sustainable Energy Reviews* 88 (2018), S. 297–325. <http://dx.doi.org/10.1016/j.rser.2018.02.002>. – DOI 10.1016/j.rser.2018.02.002. – ISSN 13640321
- [Deepa und Senthilkumar 2016] DEEPA, O. ; SENTHILKUMAR, A.: Swarm Intelligence from Natural to Artificial Systems: Ant Colony Optimization. In: *International Journal on Applications of Graph Theory In wireless Ad Hoc Networks And sensor Networks* 8 (2016), Nr. 1, S. 9–17. <http://dx.doi.org/10.5121/jgraphoc.2016.8102>. – DOI 10.5121/jgraphoc.2016.8102. – ISSN 09757260
- [Delahaye et al. 2019] DELAHAYE, D. ; CHAIMATANAN, Supatcha ; MONGEAU, M.: Simulated Annealing: From Basics to Applications. In: *undefined* (2019). <https://www.semanticscholar.org/>

[paper/Simulated-Annealing%3A-From-Basics-to-Applications-Delahaye-Chaimatanan/bdd4dee106f5a5dead34e8227cc2fd84a99f4e31](https://doi.org/10.1016/j.jmsy.2016.12.013)

- [Dellagi et al. 2017] DELLAGI, Sofiene ; CHELBI, Anis ; TRABELSI, Wajdi: Joint integrated production-maintenance policy with production plan smoothing through production rate control. In: *Journal of Manufacturing Systems* 42 (2017), S. 262–270. <http://dx.doi.org/10.1016/j.jmsy.2016.12.013>. – DOI 10.1016/j.jmsy.2016.12.013. – ISSN 02786125
- [Dias et al. 2016] DIAS, Luis M. S. ; VIEIRA, Antonio A. C. ; PEREIRA, Guilherme A. B. ; OLIVEIRA, Jose A.: Discrete simulation software ranking — A top list of the worldwide most popular and used tools. In: ROEDER, Theresa M. (Hrsg.) ; FRAZIER, Peter I. (Hrsg.) ; SZECHTMAN, Robert (Hrsg.) ; ZHOU, Enlu (Hrsg.): *Simulating complex service systems*. Piscataway, NJ : IEEE, 2016. – ISBN 978-1-5090-4486-3, S. 1060–1071
- [Dick et al. 2017] DICK, Jeremy ; HULL, Elizabeth ; JACKSON, Ken: *Requirements engineering*. 4th ed. Cham : Springer, 2017 (Springer eBook Collection Computer Science). <http://dx.doi.org/10.1007/978-3-319-61073-3>. <http://dx.doi.org/10.1007/978-3-319-61073-3>. – ISBN 9783319610733
- [Dickersbach 2009] DICKERSBACH, Jörg T.: *Supply chain management with SAP APO: Structures, modelling approaches and implementation of SAP SCM 2008*. 3. ed. Dordrecht : Springer, 2009. <http://dx.doi.org/10.1007/978-3-540-92942-0>. <http://dx.doi.org/10.1007/978-3-540-92942-0>. – ISBN 9783540929413
- [Dickmann 2015] DICKMANN, Philipp (Hrsg.): *Schlanker Materialfluss: Mit Lean Production, Kanban und Innovationen*. 3. Auflage. Berlin : Springer Vieweg, 2015 (VDI-Buch). <http://dx.doi.org/10.1007/978-3-662-44869-4>. <http://dx.doi.org/10.1007/978-3-662-44869-4>. – ISBN 9783662448694
- [Die Presse 2016] DIE PRESSE: Bis zu 257 Prozent Preisunterschied bei gleichem Produkt. In: *Die Presse* (2016). <https://diepresse.com/home/wirtschaft/verbraucher/5074649/Bis-zu-257-Prozent-Preisunterschied-bei-gleichem-Produkt>
- [Diekmann 2010] DIEKMANN, Cornelius ; LEHRSTUHL ROBOTICS AND EMBEDDED SYSTEMS, FAKULTÄT FÜR INFORMATIK (Hrsg.): *Evolutionsstrategien: Seminar Evolutionary Algorithms SS 2010*. <http://home.in.tum.de/~diekmann/evolutionsstrategien.pdf>. Version: 2010
- [Dietmar Hömberg 2005] DIETMAR HÖMBERG: *Nichtlineare Optimierung*. Berlin, TU Berlin, Skriptum zur Vorlesung, 2005
- [Domschke und Scholl 2006] DOMSCHKE, W. ; SCHOLL, A.: *Heuristische Verfahren*. Jena, Universität Jena, Arbeits- und Diskussionspapiere, 2006
- [Dorigo und Stützle 2004] DORIGO, Marco ; STÜTZLE, Thomas: *Ant colony optimization*. Cambridge, Mass : MIT Press, 2004 <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=122502>. – ISBN 9780262042192
- [Dyckhoff und Spengler 2010] DYCKHOFF, Harald ; SPENGLER, Thomas S.: *Produktionswirtschaft: Eine Einführung*. 3., überarb. und erw. Aufl. Berlin : Springer, 2010 (Springer-Lehrbuch). <http://dx.doi.org/10.1007/978-3-642-13684-9>. – ISBN 978-3-642-13683-2
- [E. Suwondo und H. Yulianto 2012] E. SUWONDO ; H. YULIANDO: DYNAMIC LOT-SIZING PROBLEMS: A Review on Model and Efficient Algorithm. In: *undefined* (2012). <https://www.semanticscholar.org/paper/DYNAMIC-LOT-SIZING-PROBLEMS%3A-A-Review-on-Model-and-Suwondo-Yulianto/c7faca6e6793c057db83ec06f71139db0b268528>
- [Edis 2021] EDIS, Emrah B.: Constraint programming approaches to disassembly line balancing problem with sequencing decisions. In: *Computers & Operations Research* 126 (2021), S. 105111. <http://dx.doi.org/10.1016/j.cor.2020.105111>. – DOI 10.1016/j.cor.2020.105111. – ISSN 03050548

- [Ehrgott und Gandibleux 2003] EHRGOTT, Matthias (Hrsg.) ; GANDIBLEUX, Xavier (Hrsg.): *International Series in Operations Research & Management Science*. Bd. 52: *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Boston, MA : Kluwer Academic Publishers, 2003. <http://dx.doi.org/10.1007/b101915>. <http://dx.doi.org/10.1007/b101915>. – ISBN 0–306–48107–3
- [Eisenberg 2007] EISENBERG, Wolfgang (Hrsg.): *Naturwissenschaftliche und technische Systeme im Fokus von Fremd- und Selbstorganisation: Symposium 2005*. Leipzig : Leipziger Univ.-Verl., 2007. – ISBN 978–3–86583–218–4
- [Ellinger et al. 2003] ELLINGER, Theodor ; BEUERMANN, Günter ; LEISTEN, Rainer: *Operations Research: Eine Einführung*. 6. Berlin and Heidelberg : Springer, 2003 (Springer-Lehrbuch). <http://dx.doi.org/10.1007/978-3-642-55582-4>. <http://dx.doi.org/10.1007/978-3-642-55582-4>. – ISBN 978–3–642–55582–4
- [Elsevier B.V. 2021a] ELSEVIER B.V.: *ScienceDirect*. <https://www.sciencedirect.com/>. Version: 2021. Zuletzt geprüft am: 20.12.2021
- [Elsevier B.V. 2021b] ELSEVIER B.V.: *Scopus*. <https://www.scopus.com/>. Version: 2021. Zuletzt geprüft am: 20.12.2021
- [Esteso et al. 2018] ESTESO, Ana ; ALEMANY, M.M.E. ; ORTIZ, Ángel ; PEIDRO, David: A multi-objective model for inventory and planned production reassignment to committed orders with homogeneity requirements. In: *Computers & Industrial Engineering* 124 (2018), 180–194. <http://dx.doi.org/10.1016/j.cie.2018.07.025>. – DOI 10.1016/j.cie.2018.07.025. – ISSN 03608352
- [Felzl 2003] FELTL, Harald: *Ein Genetischer Algorithmus für das Generalized Assignment Problem*. Wien, TU Wien, Diplomarbeit, 2003
- [Fernandez-Viagas et al. 2018] FERNANDEZ-VIAGAS, Victor ; MOLINA-PARIENTE, Jose M. ; FRAMINAN, Jose M.: New efficient constructive heuristics for the hybrid flowshop to minimise makespan: A computational evaluation of heuristics. In: *Expert Systems with Applications* 114 (2018), S. 345–356. <http://dx.doi.org/10.1016/j.eswa.2018.07.055>. – DOI 10.1016/j.eswa.2018.07.055. – ISSN 09574174
- [FICO 2021] FICO: *Fico Xpress-Optimizer*. <https://www.fico.com/de/products/fico-xpress-optimization>. Version: 2021. Zuletzt geprüft am: 21.12.2021
- [Framinan und Leisten 2003] FRAMINAN, J. M. ; LEISTEN, R.: An efficient constructive heuristic for flowtime minimisation in permutation flow shops. In: *Omega* 31 (2003), Nr. 4, S. 311–317. [http://dx.doi.org/10.1016/S0305-0483\(03\)00047-1](http://dx.doi.org/10.1016/S0305-0483(03)00047-1). – DOI 10.1016/S0305-0483(03)00047-1. – ISSN 03050483
- [Freitag und Hildebrandt 2016] FREITAG, Michael ; HILDEBRANDT, Torsten: Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization. In: *CIRP Annals* 65 (2016), Nr. 1, 433–436. <http://dx.doi.org/10.1016/j.cirp.2016.04.066>. – DOI 10.1016/j.cirp.2016.04.066. – ISSN 0007–8506
- [Freitas 2004] FREITAS, Alex A.: A Critical review of Multi-Objective Optimization in Data Mining. In: *ACM SIGKDD Explorations Newsletter* 2004 (2004), Nr. Volume 6 Issue 2, December 2004, S. 77–87
- [Fu et al. 2019] FU, Yaping ; TIAN, Guangdong ; FATHOLLAHI-FARD, Amir M. ; AHMADI, Abbas ; ZHANG, Chaoyong: Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint. In: *Journal of Cleaner Production* 226 (2019), S. 515–525. <http://dx.doi.org/10.1016/j.jclepro.2019.04.046>. – DOI 10.1016/j.jclepro.2019.04.046. – ISSN 09596526
- [Gaafar 2006] GAAFAR, Lotfi: Applying genetic algorithms to dynamic lot sizing with batch ordering. In: *Computers & Industrial Engineering* 51 (2006), Nr. 3, S. 433–444. <http://dx.doi.org/10.1016/j.cie.2006.08.006>. – DOI 10.1016/j.cie.2006.08.006. – ISSN 03608352
- [Galarnyk 2018] GALARNYK, Michael ; TOWARDS DATA SCIENCE INC. (Hrsg.): *Understanding Boxplots*. <https://towardsdatascience.com/understanding-boxplots-5e2df7bcdb51>. Version: 2018

- [GAMS Software GmbH 2021] GAMS SOFTWARE GMBH: *GAMS: General Algebraic Modeling System*. https://www.gams.com/latest/docs/S_MAIN.html. Version: 2021. Zuletzt geprüft am: 21.12.2021
- [Garre et al. 2020] GARRE, Alberto ; RUIZ, Mari C. ; HONTORIA, Eloy: Application of Machine Learning to support production planning of a food industry in the context of waste generation under uncertainty. In: *Operations Research Perspectives* 7 (2020), 100147. <http://dx.doi.org/10.1016/j.orp.2020.100147>. – DOI 10.1016/j.orp.2020.100147. – ISSN 22147160
- [Gendreau und Potvin 2019] GENDREAU, Michel (Hrsg.) ; POTVIN, Jean-Yves (Hrsg.): *International Series in Operations Research & Management Science*. Bd. 272: *Handbook of Metaheuristics*. 3rd ed. 2019. Cham : Springer International Publishing, 2019. <http://dx.doi.org/10.1007/978-3-319-91086-4>. <http://dx.doi.org/10.1007/978-3-319-91086-4>. – ISBN 978-3-319-91086-4
- [Gerdes et al. 2004] GERDES, I. ; KLAWONN, F. ; KRUSE, R.: *Evolutionäre Algorithmen: Genetische Algorithmen – Strategien und Optimierungsverfahren – Beispielanwendungen*. Wiesbaden : Vieweg+Teubner Verlag, 2004 (Computational Intelligence). <https://books.google.at/books?id=TBd6rWMLCqC>. – ISBN 9783322868398
- [Ghasemalizadeh et al. 2016] GHASEMALIZADEH, Omid ; KHALEGHIAN, Seyedmeysam ; TAHERI., Saied: A Review of Optimization Techniques in Artificial Networks. In: *International Journal of Advanced Research* 4 (2016), Nr. 9, S. 1668–1686. <http://dx.doi.org/10.21474/IJAR01/1627>. – DOI 10.21474/IJAR01/1627. – ISSN 23205407
- [Glover und Kochenberger 2003] GLOVER, Fred (Hrsg.) ; KOCHENBERGER, Gary A. (Hrsg.): *International Series in Operations Research & Management Science*. Bd. 57: *Handbook of metaheuristics*. Boston : Kluwer Academic Publishers, 2003 <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=99360>. – ISBN 9781402072635
- [Goldberg 1989] GOLDBERG, David E.: *Genetic algorithms in search, optimization, and machine learning*. Reprinted with corrections. Reading (MA) : Addison-Wesley, 1989. – ISBN 0-201-15767-5
- [Goldberg 2012] GOLDBERG, David E.: *Genetic algorithms in search, optimization, and machine learning*. 30. print. Boston : Addison-Wesley, 2012. – ISBN 0201157675
- [Gonçalves und Oliveira 2018] GONÇALVES, M.L.N. ; OLIVEIRA, F. R.: An inexact Newton-like conditional gradient method for constrained nonlinear systems. In: *Applied Numerical Mathematics* 132 (2018), S. 22–34. <http://dx.doi.org/10.1016/j.apnum.2018.05.006>. – DOI 10.1016/j.apnum.2018.05.006. – ISSN 01689274
- [Google LLC 2021] GOOGLE LLC: *Google Scholar*. <https://scholar.google.com/>. Version: 2021. Zuletzt geprüft am: 20.12.2021
- [Gorman und Brannon 2000] GORMAN, Michael F. ; BRANNON, James I.: Seasonality and the production-smoothing model. In: *International Journal of Production Economics* 65 (2000), Nr. 2, S. 173–178. [http://dx.doi.org/10.1016/S0925-5273\(99\)00049-3](http://dx.doi.org/10.1016/S0925-5273(99)00049-3). – DOI 10.1016/S0925-5273(99)00049-3. – ISSN 09255273
- [Gottschalk 2005] GOTTSCHALK, Lutz L.: *Flexibilitätsprofile: Analyse und Konfiguration von Strategien zur Kapazitätsanpassung in der industriellen Produktion*. ETH Zurich, 2005. <http://dx.doi.org/10.3929/ethz-a-005150357>. <http://dx.doi.org/10.3929/ethz-a-005150357>
- [Graf 2003] GRAF, Sabine: *Auswahl und Implementierung eines Ameisenalgorithmus‘ zur Steuerung von Patienten im Planspiel „INVENT“*. Wien, Universität Wien, Diplomarbeit, 2003. http://www.wit.at/people/graf/publications/Diplomarbeit_Sabine_Graf.pdf
- [Grefenstette 1987] GREFENSTETTE, John J.: *Genetic algorithms and their applications: Proceedings of the second International Conference on Genetic Algorithms : July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*. Hillsdale, N.J. : L. Erlbaum Associates, 1987. – ISBN 1138414190

- [Grundmann und Reinsch] GRUNDMANN, S. ; REINSCH, S.: Flexible Kapazitäten im Maschinen- und Anlagenbau. Umfrageergebnisse von Unternehmen des Maschinen- und Anlagenbaus. In: *Industrie Management, Heft 5* Bd. 2006, S. 42–44
- [Haarkam 2012] HAARKAM, Maximilian: *Anforderungsanalyse zur Einführung von SAP APO*. Leoben, Montanuniversität Leoben, Masterarbeit, 2012
- [Hachtel und Holzbaur 2010] HACHTEL, Günther ; HOLZBAUR, Ulrich: *Management für Ingenieure: Technisches Management für Ingenieure in Produktion und Logistik*. Wiesbaden : Vieweg + Teubner, 2010. <http://dx.doi.org/10.1007/978-3-8348-9338-3>. <http://dx.doi.org/10.1007/978-3-8348-9338-3>. – ISBN 978–3–8348–9338–3
- [Hader 2000] HADER, Sven: *Hybride Optimierung als Werkzeug zur Lösung von KMU-relevanten Planungs- und Steuerungsproblemen: Forschungsbericht*. <https://www.sven-hader.de/papers/h0002.pdf>. Version: 2000
- [Hakanen et al. 2013] HAKANEN, Jussi ; HARTIKAINEN, Markus ; SINDYA KARTHIK: A Priori Methods in Multiobjective Optimization: Nonlinear Multiobjective Optimization. In: *Skriptum 2013* (2013). <http://users.jyu.fi/~jhaka/uppsala/>
- [Haraszko und Németh 2015] HARASZKÓ, Csaba ; NÉMETH, István: DES Configurators for Rapid Virtual Prototyping and Optimisation of Manufacturing Systems. In: *Periodica Polytechnica Mechanical Engineering* 59 (2015), Nr. 3, S. 143–152. <http://dx.doi.org/10.3311/PPme.7888>. – DOI 10.3311/PPme.7888. – ISSN 03246051
- [Heinzl 2020] HEINZL, Bernhard: *Phasenmodell zur Entwicklung einer werkzeug- und maschinenzustandsbedingten Produktionsreihenfolgeoptimierung*. Wien, Technische Universität Wien, Dissertation, 2020
- [Heizer und Render 2008] HEIZER, Jay H. ; RENDER, Barry: *Operations management*. 9th ed. Upper Saddle River, N.J. : Pearson Prentice Hall, 2008. – ISBN 0138128782
- [Hellwig 2009] HELLWIG, Michael: *Multikriterielle Optimierung mittels adaptiver Blocknormen*. Dortmund, Universität Dortmund, Diplomarbeit, 2009. <https://homepages.fhv.at/hemi/wp-content/uploads/2015/07/Diplomarbeit.pdf>
- [Hennebel et al. 2017] HENNEBEL, Veerle ; SIMPER, Richard ; VERSCHELDE, Marijn: Is there a prison size dilemma? An empirical analysis of output-specific economies of scale. In: *European Journal of Operational Research* 262 (2017), Nr. 1, S. 306–321. <http://dx.doi.org/10.1016/j.ejor.2017.03.055>. – DOI 10.1016/j.ejor.2017.03.055. – ISSN 03772217
- [Hevner Alan R. 2007] HEVNER ALAN R.: A Three Cycle View of Design Science Research. In: *Scandinavian Journal of Information Systems Vol. 19: Iss. 2* (2007). <https://aisel.aisnet.org/sjis/vol19/iss2/4>
- [Hevner Alan R. et al. 2004] HEVNER ALAN R. ; MARCH, Salvatore T. ; PARK, Jinsoo ; RAM, Sudha: Design Science in Information Systems Research. In: *MIS Quarterly* 28 (2004), Nr. 1, S. 75. <http://dx.doi.org/10.2307/25148625>. – DOI 10.2307/25148625. – ISSN 02767783
- [Hillier und Lieberman 2015] HILLIER, Frederick S. ; LIEBERMAN, Gerald J.: *Introduction to operations research*. 10. ed. New York, NY : McGraw-Hill Education, 2015. – ISBN 978–0–07–352345–3
- [Hinze 2012] HINZE, Thomas: *Prinzipien zur Erzeugung von Zufallszahlen in der Informatik*. Cottbus, TU Cottbus, Skriptum zur Probevorlesung im Habilitationsverfahren, 2012
- [Holland 2010] HOLLAND, John H.: *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. [Nachdr.]. Cambridge, Mass. : MIT Press, 2010 (A Bradford book). <https://books.google.at/books?id=5EgGaBkwvWcc>. – ISBN 9780262581110
- [Hua et al. 2008] HUA, Zhongsheng ; ZHANG, Bin ; XU, Xiaoyan: A new variable reduction technique for convex integer quadratic programs. In: *Applied Mathematical Modelling* 32 (2008), Nr. 2, S. 224–231. <http://dx.doi.org/10.1016/j.apm.2006.11.011>. – DOI 10.1016/j.apm.2006.11.011. – ISSN 0307904X

- [IBM ILOG 2021] IBM ILOG: *CPLEX Optimizer*. <https://www.ibm.com/analytics/cplex-optimizer>. Version: 2021. Zuletzt geprüft am: 21.12.2021
- [Imam et al. 1999] IMAM, Ibrahim (Hrsg.) ; KODRATOFF, Yves (Hrsg.) ; EL-DESSOUKI, Ayman (Hrsg.) ; ALI, Moonis (Hrsg.): *Lecture Notes in Computer Science*. Bd. 1611: *Multiple Approaches to Intelligent Systems: 12th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE-99, Cairo, Egypt, May 31 - June 3, 1999. Proceedings*. Berlin and Heidelberg : Springer, 1999. <http://dx.doi.org/10.1007/b72302>. <http://dx.doi.org/10.1007/b72302>. – ISBN 978-3-540-48765-4
- [J. Arabas et al. 1994] J. ARABAS ; Z. MICHALEWICZ ; J. MULAWKA: GAVaPS-a genetic algorithm with varying population size. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 1994, S. 73–78 vol.1
- [Jans und Degraeve 2007] JANS, Raf ; DEGRAEVE, Zeger: Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. In: *European Journal of Operational Research* 177 (2007), Nr. 3, S. 1855–1875. <http://dx.doi.org/10.1016/j.ejor.2005.12.008>. – DOI 10.1016/j.ejor.2005.12.008. – ISSN 03772217
- [Jianbang Gan et al. 1995] JIANBANG GAN ; J.P. COLLETTI ; S.H. KOLISON: A compromise programming approach to integrated natural resource management. In: *Proceedings of the 1994 Symposium on Systems Analysis in Forest Resources: (1995)*, S. 378–386
- [Jimenez et al. 2002] JIMENEZ, F. ; GOMEZ-SKARMETA, A.F. ; SANCHEZ, G. ; DEB, K.: An evolutionary algorithm for constrained multi-objective optimization. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* Bd. 2, 2002, S. 1133–1138 vol.2
- [Jin et al. 2019] JIN, Yaochu ; WANG, Handing ; CHUGH, Tinkle ; GUO, Dan ; MIETTINEN, Kaisa: Data-Driven Evolutionary Optimization: An Overview and Case Studies. In: *IEEE Transactions on Evolutionary Computation* 23 (2019), Nr. 3, S. 442–458. <http://dx.doi.org/10.1109/TEVC.2018.2869001>. – DOI 10.1109/TEVC.2018.2869001. – ISSN 1089-778X
- [Juan et al. 2015] JUAN, Angel A. ; FAULIN, Javier ; GRASMAN, Scott E. ; RABE, Markus ; FIGUEIRA, Gonçalo: A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. In: *Operations Research Perspectives* 2 (2015), S. 62–72. <http://dx.doi.org/10.1016/j.orp.2015.03.001>. – DOI 10.1016/j.orp.2015.03.001. – ISSN 22147160
- [Jungnickel 2010] JUNGNICHEL, Dieter: *Algorithms and computation in mathematics*. Bd. 5: *Graphs, networks and algorithms: With 9 tables*. 3. ed. Berlin : Springer, 2010. – ISBN 978-3-540-72779-8
- [Kallrath und Maindl 2006] KALLRATH, Josef ; MAINDL, Thomas I.: *Real Optimization with SAP APO*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2006. <http://dx.doi.org/10.1007/3-540-34624-4>. <http://dx.doi.org/10.1007/3-540-34624-4>. – ISBN 978-3-540-22561-4
- [Kamburowski 1997] KAMBUROWSKI, Jerzy: The nature of simplicity of Johnson's algorithm. In: *Omega* 25 (1997), Nr. 5, S. 581–584. [http://dx.doi.org/10.1016/S0305-0483\(97\)00021-2](http://dx.doi.org/10.1016/S0305-0483(97)00021-2). – DOI 10.1016/S0305-0483(97)00021-2. – ISSN 03050483
- [Kamhuber 2010] KAMHUBER, Felix: *Simulationsbasierte metaheuristische Optimierung eines realen Auftragsreihenfolgeproblems*. Leoben, Montanuniversität Leoben, Masterarbeit, 2010. https://online.unileoben.ac.at/mu_online/wbAbs.showThesis?pThesisNr=19129&pOrgNr=1
- [Kamhuber et al. 2020] KAMHUBER, Felix ; SOBOTTKA, Thomas ; HEINZL, Bernhard ; HENJES, Jan ; SIHN, Wilfried: An efficient hybrid multi-criteria optimization approach for rolling production smoothing of a European food manufacturer. In: *Computers & Industrial Engineering* 147 (2020), Nr. 2020, S. 1–16. <http://dx.doi.org/10.1016/j.cie.2020.106620>. – DOI 10.1016/j.cie.2020.106620. – ISSN 03608352
- [Kamhuber et al. 2019] KAMHUBER, Felix ; SOBOTTKA, Thomas ; HEINZL, Bernhard ; SIHN, Wilfried: An Efficient Multi-Objective Hybrid Simheuristic Approach for Advanced Rolling Horizon Production Planning. In: MUSTAFEE, N. (Hrsg.): *2019 Winter Simulation Conference (WSC)*. [Piscataway, NJ] : IEEE, 2019. – ISBN 978-1-7281-3283-9, S. 2108–2118

- [Kamhuber et al. 2018] KAMHUBER, Felix ; SOBOTTKA, Thomas ; SCHIEDER, Peter ; ULRICH, Maximilian ; SIHN, Wilfried: An Innovative Heuristic Mixed-Integer Optimization Approach for Multi-Criteria Optimization based Production Planning in the context of Production Smoothing. In: SCIENCES-CONF.ORG (Hrsg.): *Proceedings of META'2018*, 2018, S. 101–116
- [Karaivanova et al. 1993] KARAIVANOVA, Jasmina N. ; NARULA, Subhash C. ; VASSILEV, Vassil: An interactive procedure for multiple objective integer linear programming problems. In: *European Journal of Operational Research* 68 (1993), Nr. 3, S. 344–351. [http://dx.doi.org/10.1016/0377-2217\(93\)90190-X](http://dx.doi.org/10.1016/0377-2217(93)90190-X). – DOI 10.1016/0377-2217(93)90190-X. – ISSN 03772217
- [Karapetyan und Gutin 2011] KARAPETYAN, D. ; GUTIN, G.: Lin–Kernighan heuristic adaptations for the generalized traveling salesman problem. In: *European Journal of Operational Research* 208 (2011), Nr. 3, S. 221–232. <http://dx.doi.org/10.1016/j.ejor.2010.08.011>. – DOI 10.1016/j.ejor.2010.08.011. – ISSN 03772217
- [Karimi-Nasab und Aryanezhad 2011] KARIMI-NASAB, M. ; ARYANEZHAD, M. B.: A multi-objective production smoothing model with compressible operating times. In: *Applied Mathematical Modelling* 35 (2011), Nr. 7, S. 3596–3610. <http://dx.doi.org/10.1016/j.apm.2011.01.038>. – DOI 10.1016/j.apm.2011.01.038. – ISSN 0307904X
- [Karimi-Nasab und Konstantaras 2012] KARIMI-NASAB, Mehdi ; KONSTANTARAS, Ioannis: A random search heuristic for a multi-objective production planning. In: *Computers & Industrial Engineering* 62 (2012), Nr. 2, S. 479–490. <http://dx.doi.org/10.1016/j.cie.2011.10.015>. – DOI 10.1016/j.cie.2011.10.015. – ISSN 03608352
- [Karner 2019] KARNER, Matthias: *Phasenmodell zur Entwicklung einer werkzeug- und maschinenzustandsbedingten Produktionsreihenfolgeoptimierung*. Wien, Technische Universität Wien, Dissertation, 2019
- [Keller 2013] KELLER, Daniela: *Der Boxplot*. <https://statistik-und-beratung.de/2013/02/der-boxplot/>. Version: 2013. Zuletzt geprüft am: 21.12.2020
- [Kelner et al. 2008] KELNER, Vincent ; CAPITANESCU, Florin ; LÉONARD, Olivier ; WEHENKEL, Louis: A hybrid optimization technique coupling an evolutionary and a local search algorithm. In: *Journal of Computational and Applied Mathematics* 215 (2008), Nr. 2, S. 448–456. <http://dx.doi.org/10.1016/j.cam.2006.03.048>. – DOI 10.1016/j.cam.2006.03.048. – ISSN 03770427
- [Kennedy und Eberhart 1995] KENNEDY, J. ; EBERHART, R.: Particle swarm optimization. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, 1995. – ISBN 0-7803-2768-3, S. 1942–1948
- [Knowles 2006] KNOWLES, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. In: *IEEE Transactions on Evolutionary Computation* 10 (2006), Nr. 1, S. 50–66. <http://dx.doi.org/10.1109/TEVC.2005.851274>. – DOI 10.1109/TEVC.2005.851274
- [Koch 2015] KOCH, Susanne: *Einführung in das Management von Geschäftsprozessen: Six Sigma, Kaizen und TQM*. 2. Aufl. Berlin : Springer Vieweg, 2015. <http://dx.doi.org/10.1007/978-3-662-44450-4>. <http://dx.doi.org/10.1007/978-3-662-44450-4>. – ISBN 978-3-662-44450-4
- [Kogan und Tell 2009] KOGAN, Konstantin ; TELL, Hanan: Production smoothing by balancing capacity utilization and advance orders. In: *IIE Transactions* 41 (2009), Nr. 3, S. 223–231. <http://dx.doi.org/10.1080/07408170802116305>. – DOI 10.1080/07408170802116305
- [Konak et al. 2006] KONAK, Abdullah ; COIT, David W. ; SMITH, Alice E.: Multi-objective optimization using genetic algorithms: A tutorial. In: *Reliability Engineering & System Safety* 91 (2006), Nr. 9, S. 992–1007. <http://dx.doi.org/10.1016/j.res.2005.11.018>. – DOI 10.1016/j.res.2005.11.018. – ISSN 09518320
- [Koohestani 2020] KOOHESTANI, Behrooz: A crossover operator for improving the efficiency of permutation-based genetic algorithms. In: *Expert Systems with Applications* 151 (2020), Nr. 113381.

- <http://dx.doi.org/10.1016/j.eswa.2020.113381>. – DOI 10.1016/j.eswa.2020.113381. – ISSN 09574174
- [Korte und Vygen 2012] KORTE, Bernhard ; VYGEN, Jens: *Kombinatorische Optimierung: Theorie und Algorithmen*. 2. Aufl. 2012. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012 (SpringerLink Bücher). <http://dx.doi.org/10.1007/978-3-642-25401-7>. – ISBN 978-3-642-25400-0
- [Korytkowski et al. 2013] KORYTKOWSKI, Przemyslaw ; WISNIEWSKI, Tomasz ; RYMASZEWSKI, Szymon: Multivariate simulation analysis of production leveling (heijunka) - a case study. In: *IFAC Proceedings Volumes* 46 (2013), Nr. 9, S. 1554–1559. <http://dx.doi.org/10.3182/20130619-3-RU-3018.00285>. – DOI 10.3182/20130619-3-RU-3018.00285. – ISSN 14746670
- [Kritzinger et al. 2018] KRITZINGER, Werner ; KARNER, Matthias ; TRAR, Georg ; HENJES, Jan ; SIHN, Wilfried: Digital Twin in manufacturing: A categorical literature review and classification: IFAC-PapersOnLine, 51(11), 1016-1022. In: *IFAC-PapersOnLine* 51 (2018), Nr. 11, S. 1016–1022. <http://dx.doi.org/10.1016/J.IFACOL.2018.08.474>. – DOI 10.1016/J.IFACOL.2018.08.474. – ISSN 24058963
- [Kugi 2011] KUGI, Andreas: *Optimierung: Skriptum zur Volesung*. Wien, TU Wien, Skriptum, 2011
- [Kuhn et al. 2016] KUHN, K. ; RATH, A. ; SCHMIDT, M. ; SCHÖBEL, A.: Bi-objective robust optimisation. In: *European Journal of Operational Research* 252 (2016), Nr. 2, S. 418–431. <http://dx.doi.org/10.1016/j.ejor.2016.01.015>. – DOI 10.1016/j.ejor.2016.01.015. – ISSN 03772217
- [Kumar und Aouam 2019] KUMAR, Kunal ; AOUAM, Tarik: Extending the strategic safety stock placement model to consider tactical production smoothing. In: *European Journal of Operational Research* 279 (2019), Nr. 2, 429–448. <https://ideas.repec.org/a/eee/ejores/v279y2019i2p429-448.html>
- [Kurbel 2016] KURBEL, Karl: *Enterprise Resource Planning und Supply Chain Management in der Industrie: Von MRP bis Industrie 4.0*. 8., vollst. überarb. und erw. Auflage. Berlin/Boston : De Gruyter, 2016 (De Gruyter Studium). <http://dx.doi.org/10.1515/9783110441697>. <http://dx.doi.org/10.1515/9783110441697>. – ISBN 978-3-11-044168-0
- [Kux 2011] KUX, Steffen: *Hybride Optimierungsstrategien für komplexe technische Aufgabenstellungen*. Mittweida, Hochschule Mittweida, Masterarbeit, 2011
- [Laha und Sarin 2009] LAHA, Dipak ; SARIN, Subhash C.: A heuristic to minimize total flow time in permutation flow shop. In: *Omega* 37 (2009), Nr. 3, S. 734–739. <http://dx.doi.org/10.1016/j.omega.2008.05.002>. – DOI 10.1016/j.omega.2008.05.002. – ISSN 03050483
- [Ławrynowicz 2011] ŁAWRYNOWICZ, Anna: Genetic Algorithms for Solving Scheduling Problems in Manufacturing Systems. In: *Foundations of Management* 3 (2011), Nr. 2, S. 7–26. <http://dx.doi.org/10.2478/v10238-012-0039-2>. – DOI 10.2478/v10238-012-0039-2. – ISSN 2080-7279
- [Lenort et al. 2012] LENORT, Radim (Hrsg.) ; KLEPEK, Roman (Hrsg.) ; SAMOLEJOVÁ, Andrea (Hrsg.): *Heuristic algorithm for planning and scheduling of forged pieces heat treatment*. Hrvatsko Metalurško Društvo, 2012
- [Lingitz et al. 2018] LINGITZ, Lukas ; GALLINA, Viola ; ANSARI, Fazel ; GYULAI, Dávid ; PFEIFFER, András ; SIHN, Wilfried ; MONOSTORI, László: Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. In: *Procedia CIRP* 72 (2018), 1051–1056. <http://dx.doi.org/10.1016/j.procir.2018.03.148>. – DOI 10.1016/j.procir.2018.03.148. – ISSN 22128271
- [Lingitz et al. 2019] LINGITZ, Lukas ; GALLINA, Viola ; KARDOS, Csaba ; KOLTAI, Tamás ; SIHN, Wilfried: Balancing non-bottleneck stations using simple assembly line balancing models. In: *IFAC-PapersOnLine* 52 (2019), Nr. 13, S. 1432–1437. <http://dx.doi.org/10.1016/j.ifacol.2019.11.400>. – DOI 10.1016/j.ifacol.2019.11.400. – ISSN 24058963

- [Lödding 2008] LÖDDING, Hermann: *Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration*. 2., erw. Aufl. Berlin, Heidelberg : Springer-Verlag, 2008 (VDI-Buch). <http://dx.doi.org/10.1007/978-3-540-76860-9>. <http://dx.doi.org/10.1007/978-3-540-76860-9>. – ISBN 978-3-540-76859-3
- [Low et al. 2004] LOW, Chinyao ; HSU, Chih-Ming ; HUANG, Kai-I: Benefits of lot splitting in job-shop scheduling. In: *The International Journal of Advanced Manufacturing Technology* 24 (2004), Nr. 9-10, S. 773–780. <http://dx.doi.org/10.1007/s00170-003-1785-9>. – DOI 10.1007/s00170-003-1785-9. – ISSN 0268-3768
- [Marinelli et al. 2007] MARINELLI, Fabrizio ; NENNI, Maria E. ; SFORZA, Antonio: Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company. In: *Annals of Operations Research* 150 (2007), Nr. 1, S. 177–192. <http://dx.doi.org/10.1007/s10479-006-0157-x>. – DOI 10.1007/s10479-006-0157-x. – ISSN 0254-5330
- [Marketagent.com 2017] MARKETAGENT.COM ; MARKETAGENT.COM ONLINE RESEARCH GMBH (Hrsg.): *10 Customer Journeys, 10 Produkte*. https://www.handelsverband.at/fileadmin/content/images/publikationen/Studien/STUDIE_ExpeditionKunde_CustomerJourney_2017.pdf. Version: 2017. Zuletzt geprüft am: 14.01.2018
- [Marler und Arora 2004] MARLER, R. T. ; ARORA, J. S.: Survey of multi-objective optimization methods for engineering. In: *Structural and Multidisciplinary Optimization* 26 (2004), Nr. 6, S. 369–395. <http://dx.doi.org/10.1007/s00158-003-0368-6>. – DOI 10.1007/s00158-003-0368-6
- [Marler und Arora 2010] MARLER, R. T. ; ARORA, Jasbir S.: The weighted sum method for multi-objective optimization: New insights. In: *Structural and Multidisciplinary Optimization* 41 (2010), Nr. 6, S. 853–862. <http://dx.doi.org/10.1007/s00158-009-0460-7>. – DOI 10.1007/s00158-009-0460-7
- [Martin Stöcker 2007] MARTIN STÖCKER: *Untersuchung von Optimierungsverfahren für rechenzeitaufwändige technische Anwendungen in der Motorenentwicklung*. Chemnitz, TU Chemnitz, Diplomarbeit, 2007
- [März et al. 2011] MÄRZ, Lothar ; KRUG, Wilfried ; ROSE, Oliver ; WEIGERT, Gerald: *VDI-Buch. Bd. 130: Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2011. <http://dx.doi.org/10.1007/978-3-642-14536-0>. <http://dx.doi.org/10.1007/978-3-642-14536-0>. – ISBN 978-3-642-14535-3
- [Mashwani 2011] MASHWANI, Wali K.: Hybrid Multiobjective Evolutionary Algorithms: A Survey of the State-of -the-art. In: *International Journal of Computer Science Issues* 2011 (2011), Nr. Vol. 8, Issue 6, No 3, S. 374–392
- [Masmoudi et al. 2016] MASMOUDI, Oussama ; YALAOUI, Alice ; OUAZENE, Yassine ; CHEHADE, Hicham: Multi-item capacitated lot-sizing problem in a flow-shop system with energy consideration. In: *IFAC-PapersOnLine* 49 (2016), Nr. 12, S. 301–306. <http://dx.doi.org/10.1016/j.ifacol.2016.07.621>. – DOI 10.1016/j.ifacol.2016.07.621. – ISSN 24058963
- [Mercé und Fontan 2003] MERCÉ, C. ; FONTAN, G.: MIP-based heuristics for capacitated lotsizing problems. In: *International Journal of Production Economics* 85 (2003), Nr. 1, S. 97–111. [http://dx.doi.org/10.1016/S0925-5273\(03\)00090-2](http://dx.doi.org/10.1016/S0925-5273(03)00090-2). – DOI 10.1016/S0925-5273(03)00090-2. – ISSN 09255273
- [Mertens und Bauke 2004] MERTENS, Stephan ; BAUKE, Heiko: Entropy of pseudo-random-number generators. In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 69 (2004), Nr. 5 Pt 2, S. 055702. <http://dx.doi.org/10.1103/PhysRevE.69.055702>. – DOI 10.1103/PhysRevE.69.055702. – ISSN 1539-3755
- [Merz 2003] MERZ, Peter: *Moderne heuristische Optimierungsverfahren: Meta-Heuristiken*. Tübingen, Universität Tübingen, Skriptum, 2003

- [Metropolis et al. 1953] METROPOLIS, Nicholas ; ROSENBLUTH, Arianna W. ; ROSENBLUTH, Marshall N. ; TELLER, Augusta H. ; TELLER, Edward: Equation of State Calculations by Fast Computing Machines. In: *The Journal of Chemical Physics* 21 (1953), Nr. 6, S. 1087–1092. <http://dx.doi.org/10.1063/1.1699114>. – DOI 10.1063/1.1699114. – ISSN 0021–9606
- [Meywerk 2007] MEYWERK, Martin: *CAE-Methoden in der Fahrzeugtechnik*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2007. <http://dx.doi.org/10.1007/978-3-540-49867-4>. <http://dx.doi.org/10.1007/978-3-540-49867-4>. – ISBN 978–3–540–49867–4
- [Michalewicz 1999] MICHALEWICZ, Zbigniew: *Genetic algorithms + data structures: Evolution programs ; with 36 tables*. 3., rev. and extended ed., 1. corrected printing. Berlin : Springer, 1999. – ISBN 3–540–60676–9
- [Michalewicz und Fogel 2004] MICHALEWICZ, Zbigniew ; FOGEL, David B.: *How to solve it: Modern heuristics ; with 7 tab.* 2., rev. and extended ed. Berlin : Springer, 2004 <http://www.loc.gov/catdir/enhancements/fy0818/2004094686-d.html>. – ISBN 978–3540224945
- [Miettinen 1998] MIETTINEN, Kaisa: *International Series in Operations Research & Management Science*. Bd. 12: *Nonlinear Multiobjective Optimization*. Boston, MA : Springer, 1998. <http://dx.doi.org/10.1007/978-1-4615-5563-6>. <http://dx.doi.org/10.1007/978-1-4615-5563-6>. – ISBN 978–1–4615–5563–6
- [Miettinen Kaisa 2001] MIETTINEN KAISA (Hrsg.): *Lecture Notes in Computer Science*. Bd. 1993: *Some Methods for Nonlinear Multi-objective Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7-9, 2001 Proceedings*: In: Zitzler E., Thiele L., Deb K., Coello Coello C.A., Corne D. (eds) *Evolutionary Multi-Criterion Optimization*. Berlin and Heidelberg : Springer, 2001. <http://dx.doi.org/10.1007/3-540-44719-9>. <http://dx.doi.org/10.1007/3-540-44719-9>. – ISBN 3–540–44719–9
- [Ming 2008] MING, Cher: *Simulated Annealing: Edited by Cher Ming Tan*. InTech, 2008. <http://dx.doi.org/10.5772/67>. <http://dx.doi.org/10.5772/67>. – ISBN 978–953–7619–07–7
- [Minner 2009] MINNER, Stefan: A comparison of simple heuristics for multi-product dynamic demand lot-sizing with limited warehouse capacity. In: *International Journal of Production Economics* 118 (2009), Nr. 1, 305–310. <http://dx.doi.org/10.1016/j.ijpe.2008.08.034>. – DOI 10.1016/j.ijpe.2008.08.034. – ISSN 09255273
- [Mokhtari und Hasani 2017] MOKHTARI, Hadi ; HASANI, Aliakbar: An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. In: *Computers & Chemical Engineering* 104 (2017), 339–352. <http://dx.doi.org/10.1016/j.compchemeng.2017.05.004>. – DOI 10.1016/j.compchemeng.2017.05.004. – ISSN 0098–1354
- [Morawetz 2015] MORAWETZ, Christian: *Vorgehensmodell zur Entwicklung eines Entscheidungsunterstützungssystems zur kostenoptimalen mittelfristigen Kapazitätsanpassung*. Wien, Technische Universität Wien, Dissertation, 2015
- [Müller-Merbach 1992] MÜLLER-MERBACH, Heiner: *Operations Research: Methoden und Modelle der Optimalplanung*. 10. Nachdr. der 3., durchges. Aufl. München : Vahlen, 1992 (Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften). – ISBN 978–3800603886
- [Nickel et al. 2020] NICKEL, Stefan ; STEINHARDT, Claudius ; SCHLENKER, Hans: *Angewandte Optimierung mit IBM ILOG CPLEX Optimization Studio: Modellierung von Planungs- und Entscheidungsproblemen des Operations Research mit OPL*. 2020 (Lehrbuch). – ISBN 9783662541708
- [Ohno et al. 2013] OHNO, Taiichi ; HOF, Wilfried ; STOTKO, Eberhard C. ; ROTHER, Mike: *Das Toyota-Produktionssystem: [das Standardwerk zur Lean Production]*. 3., erw. und aktualisierte Aufl. Frankfurt am Main : Campus-Verl., 2013 (Produktion). <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=831372>. – ISBN 978–3593399294
- [Olson 2004] OLSON, D. L.: Comparison of weights in TOPSIS models. In: *Mathematical and Computer Modelling* 40 (2004), Nr. 7-8, S. 721–727. <http://dx.doi.org/10.1016/j.mcm.2004.10.003>. – DOI 10.1016/j.mcm.2004.10.003. – ISSN 08957177

- [Özdamar und Bozyel 2000] ÖZDAMAR, Linet ; BOZYL, Mehmet A.: The capacitated lot sizing problem with overtime decisions and setup times. In: *IIE Transactions* 32 (2000), Nr. 11, S. 1043–1057. <http://dx.doi.org/10.1080/07408170008967460>. – DOI 10.1080/07408170008967460
- [Petri 2006] PETRI, Christian: *Ablaufplanung bei Reihenfertigung mit mehrfacher Zielsetzung auf der Basis von Ameisenalgorithmen*. Passau, Universität Passau, Dissertation, 2006. http://www.opus-bayern.de/uni-passau/volltexte/2007/85/pdf/Petri_Christian.pdf
- [Pohlmann 2007] POHLMANN, Werner: *Algorithmen und Datenstrukturen: Skriptum zur Vorlesung*. 2007
- [Pomberger und Dobler 2008] POMBERGER, Gustav ; DOBLER, Heinz: *Algorithmen und Datenstrukturen: Eine systematische Einführung in die Programmierung*. München : Pearson Studium, 2008 (informatik). http://deposit.d-nb.de/cgi-bin/dokserv?id=3075212&prov=M&dok_var=1&dok_ext=htm. – ISBN 978-3-8273-7268-0
- [Ponsich et al. 2013] PONSICH, Antonin ; JAIMES, Antonio L. ; COELLO, Carlos A. C.: A Survey on Multiobjective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and Other Finance and Economics Applications. In: *IEEE Transactions on Evolutionary Computation* 17 (2013), Nr. 3, S. 321–344. <http://dx.doi.org/10.1109/TEVC.2012.2196800>. – DOI 10.1109/TEVC.2012.2196800
- [Press 2002] PRESS, William H.: *Numerical recipes in C: The art of scientific computing*. 2. ed., reprinted with corr. to software version 2.10. Cambridge : Univ. Press, 2002. – ISBN 0-521-43108-5
- [Prokop 2014] PROKOP, Lukas: *Grundlagen der Informatik: Skriptum zur Vorlesung "Grundlagen der Informatik" (LV-Nr. 716.232)*. Graz, 2014
- [Prügel-Bennett 2004] PRÜGEL-BENNETT, Adam: When a genetic algorithm outperforms hill-climbing. In: *Theoretical Computer Science* 320 (2004), Nr. 1, S. 135–153. <http://dx.doi.org/10.1016/j.tcs.2004.03.038>. – DOI 10.1016/j.tcs.2004.03.038. – ISSN 03043975
- [Rager 2008] RAGER, Markus: *Energieorientierte Produktionsplanung: Analyse, Konzeption und Umsetzung*. Wiesbaden : Betriebswirtschaftlicher Verlag Dr. Th. Gabler / GWV Fachverlage GmbH Wiesbaden, 2008 (Gabler Edition Wissenschaft). <http://dx.doi.org/10.1007/978-3-8350-5569-8>. – ISBN 978-3-8350-0987-5
- [Raidl und Chwatal 2008] RAIDL, Günther ; CHWATAL, Andreas: *Heuristische Optimierungsverfahren: Skriptum zur Vorlesung "Heuristische Optimierungsverfahren" (LV-Nr. 186.112)*. Wien, 2008
- [Ramadhan et al. 2017] RAMADHAN, Muhammad M. ; SITANGGANG, Imas S. ; NASUTION, Fahrendi R. ; GHIFARI, Abdullah: Parameter Tuning in Random Forest Based on Grid Search Method for Gender Classification Based on Voice Frequency. In: *DEStech Transactions on Computer Science and Engineering* (2017), Nr. cece. <http://dx.doi.org/10.12783/dtcse/cece2017/14611>. – DOI 10.12783/dtcse/cece2017/14611. – ISSN 2475-8841
- [Ramos et al. 2018] RAMOS, Bruna ; ALVES, Cláudio ; VALÉRIO DE CARVALHO, José: Column Generation Based Approaches for Combined Routing and Scheduling. In: *Electronic Notes in Discrete Mathematics* 64 (2018), S. 155–164. <http://dx.doi.org/10.1016/j.endm.2018.01.017>. – DOI 10.1016/j.endm.2018.01.017. – ISSN 15710653
- [Rewers et al. 2019] REWERS, Paulina ; BOŻEK, Mariusz ; KULUS, Wojciech: Increasing the efficiency of the production process by production levelling. In: *Management and Production Engineering Review* 10 (2019), Nr. 2, S. 93–100. <http://dx.doi.org/10.24425/MPER.2019.129572>. – DOI 10.24425/MPER.2019.129572. – ISSN 2082-1344
- [Rewers et al. 2017] REWERS, Paulina ; HAMROL, Adam ; ŻYWICKI, Krzysztof ; BOŻEK, Mariusz ; KULUS, Wojciech: Production Leveling as an Effective Method for Production Flow Control – Experience of Polish Enterprises. In: *Procedia Engineering* 182 (2017), S. 619–626. <http://dx.doi.org/10.1016/j.proeng.2017.03.167>. – DOI 10.1016/j.proeng.2017.03.167. – ISSN 18777058

- [Ricondo et al. 2021] RICONDO, Itziar ; PORTO, Alain ; UGARTE, Miriam: A digital twin framework for the simulation and optimization of production systems. In: *Procedia CIRP* 104 (2021), 762–767. <http://dx.doi.org/10.1016/j.procir.2021.11.128>. – DOI 10.1016/j.procir.2021.11.128. – ISSN 2212–8271
- [Rogalski 2009] ROGALSKI, Sven: *Reihe Informationsmanagement im Engineering Karlsruhe*. Bd. 2009,2: *Entwicklung einer Methodik zur Flexibilitätsbewertung von Produktionssystemen: Messung von Mengen-, Mix- und Erweiterungsflexibilität zur Bewältigung von Planungsunsicherheiten in der Produktion: @Karlsruhe, Univ., Diss., 2009*. Karlsruhe : Univ.-Bibl and Univ.-Verl. Karlsruhe, 2009 <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000011810>. – ISBN 9783866443839
- [Rossit et al. 2021] ROSSIT, Daniel A. ; VÁSQUEZ, Óscar C. ; TOHMÉ, Fernando ; FRUTOS, Mariano ; SAFE, Martín D.: A combinatorial analysis of the permutation and non-permutation flow shop scheduling problems. In: *European Journal of Operational Research* 289 (2021), Nr. 3, S. 841–854. <http://dx.doi.org/10.1016/j.ejor.2019.07.055>. – DOI 10.1016/j.ejor.2019.07.055. – ISSN 03772217
- [Rozenberg et al. 2012] ROZENBERG, Grzegorz (Hrsg.) ; BÄCK, Thomas (Hrsg.) ; KOK, Joost N. (Hrsg.): *Handbook of natural computing*. Berlin : Springer, 2012 (Springer reference). <http://dx.doi.org/10.1007/978-3-540-92910-9>. <http://dx.doi.org/10.1007/978-3-540-92910-9>. – ISBN 978–3–540–92910–9
- [Sampaio et al. 2017] SAMPAIO, Raimundo J. ; WOLLMANN, Rafael R. ; VIEIRA, Paula F.: A flexible production planning for rolling-horizons. In: *International Journal of Production Economics* 190 (2017), S. 31–36. <http://dx.doi.org/10.1016/j.ijpe.2017.01.003>. – DOI 10.1016/j.ijpe.2017.01.003. – ISSN 09255273
- [Sand 2013] SAND, Bastian: *Parallele Algorithmen zur Lösung des Capacitated-Vehicle-Routing-Problems*. Kaiserslautern, Technische Universität Kaiserslautern, Dissertation, 2013. <http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-36098>
- [SAP Help Portal 2021a] SAP HELP PORTAL ; SAP SE (Hrsg.): *Heuristiken*. https://help.sap.com/saphelp_scm700_ehp04/helpdata/de/46/7f3051ea67fa50e10000000a441470/frameset.htm. Version: 2021
- [SAP Help Portal 2021b] SAP HELP PORTAL ; SAP SE (Hrsg.): *Rückwärtsterminierung bei Eigenfertigung*. <https://help.sap.com/viewer/3aab1d460bbd459a9ac3ff2724bc819c/6.00.32/de-DE/91abce5314894208e10000000a174cb4.html>. Version: 2021
- [Saracoglu et al. 2014] SARACOGLU, Ilkay ; TOPALOGLU, Seyda ; KESKINTURK, Timur: A genetic algorithm approach for multi-product multi-period continuous review inventory models. In: *Expert Systems with Applications* 41 (2014), Nr. 18, S. 8189–8202. <http://dx.doi.org/10.1016/j.eswa.2014.07.003>. – DOI 10.1016/j.eswa.2014.07.003. – ISSN 09574174
- [Sathyan et al. 2015] SATHYAN, Anoop ; BOONE, Nathan ; COHEN, Kelly: Comparison of Approximate Approaches to Solving the Travelling Salesman Problem and its Application to UAV Swarming. In: *International Journal of Unmanned Systems Engineering* 3 (2015), Nr. 1, S. 1–16. <http://dx.doi.org/10.14323/ijuseng.2015.1>. – DOI 10.14323/ijuseng.2015.1
- [Sazvar et al. 2016] SAZVAR, Z. ; MIRZAPOUR AL-E-HASHEM, S.M.J. ; GOVINDAN, K. ; BAHLI, B.: A novel mathematical model for a multi-period, multi-product optimal ordering problem considering expiry dates in a FEFO system. In: *Transportation Research Part E: Logistics and Transportation Review* 93 (2016), S. 232–261. <http://dx.doi.org/10.1016/j.tre.2016.04.011>. – DOI 10.1016/j.tre.2016.04.011. – ISSN 13665545
- [Schmickl 2009] SCHMICKL, Thomas: *Schwarmintelligenz am Beispiel der Ameisenstraßen*. https://www.zobodat.at/pdf/DENISIA_0025_0141-0155.pdf. Version: 2009
- [Schoner 2008] SCHONER, Peter: *Operative Produktionsplanung in der verfahrenstechnischen Industrie: Zugl.: Kassel, Univ., Diss., 2007*. Kassel : Kassel Univ. Press, 2008. – ISBN Schoner.2008

- [Schröder 2010] SCHRÖDER, Dierk: *Intelligente Verfahren: Identifikation und Regelung nichtlinearer Systeme*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2010. <http://dx.doi.org/10.1007/978-3-642-11398-7>. <http://dx.doi.org/10.1007/978-3-642-11398-7>. – ISBN 9783642113987
- [Schuh und Stich 2012] SCHUH, Günther ; STICH, Volker: *Produktionsplanung und -steuerung*. 4., überarbeitete Auflage. Berlin and Heidelberg : Springer Vieweg, 2012 (VDI-Buch). <http://www.springer.com/de/book/978-3-642-25422-2>. – ISBN 978-3-642-25423-9
- [Schüller 2009] SCHÜLLER, A.: *Adaptive Optimierung fokussierter Substanzbibliotheken*. Frankfurt am Main, Johann Wolfgang Goethe-Universität, Dissertation, 2009. <http://publikationen.uni-frankfurt.de/opus4/frontdoor/index/index/year/2009/docId/6874>
- [Sel und Bilgen 2014] SEL, Çağrı ; BILGEN, Bilge: Hybrid simulation and MIP based heuristic algorithm for the production and distribution planning in the soft drink industry. In: *Journal of Manufacturing Systems* 33 (2014), Nr. 3, S. 385–399. <http://dx.doi.org/10.1016/j.jmsy.2014.01.002>. – DOI 10.1016/j.jmsy.2014.01.002. – ISSN 02786125
- [Selçuk et al. 2008] SELÇUK, B. ; FRANSOO, J. C. ; KOK, A. G.: Work-in-process clearing in supply chain operations planning. In: *IIE Transactions* 40 (2008), Nr. 3, S. 206–220. <http://dx.doi.org/10.1080/07408170701487997>. – DOI 10.1080/07408170701487997
- [Sihn et al. 2018] SIHN, Wilfried ; SOBOTTKA, Thomas ; HEINZL, Bernhard ; KAMHUBER, Felix: Interdisciplinary multi-criteria optimization using hybrid simulation to pursue energy efficiency through production planning. In: *CIRP Annals* 67 (2018), Nr. 1, S. 447–450. <http://dx.doi.org/10.1016/j.cirp.2018.04.059>. – DOI 10.1016/j.cirp.2018.04.059. – ISSN 00078506
- [Silberholz und Golden 2010] SILBERHOLZ, John ; GOLDEN, Bruce: Comparison of Metaheuristics. Version: 2010. http://dx.doi.org/10.1007/978-1-4419-1665-5_{_}21. In: GENDREAU, Michel (Hrsg.) ; POTVIN, Jean-Yves (Hrsg.): *Handbook of Metaheuristics* Bd. 146. s.l. : Springer Science + Business Media, 2010. – DOI 10.1007/978-1-4419-1665-5_21. – ISBN 978-1-4419-1663-1, S. 625–640
- [Silver 2004] SILVER, E. A.: An overview of heuristic solution methods. In: *Journal of the Operational Research Society* 55 (2004), Nr. 9, S. 936–956. <http://dx.doi.org/10.1057/palgrave.jors.2601758>. – DOI 10.1057/palgrave.jors.2601758. – ISSN 0160-5682
- [Sivanandam und Deepa 2008] SIVANANDAM, S. N. (Hrsg.) ; DEEPA, S. N. (Hrsg.): *Introduction to Genetic Algorithms*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2008. <http://dx.doi.org/10.1007/978-3-540-73190-0>. <http://dx.doi.org/10.1007/978-3-540-73190-0>. – ISBN 978-3-540-73189-4
- [Smyl 2020] SMYL, Slawek: A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. In: *International Journal of Forecasting* 36 (2020), Nr. 1, S. 75–85. <http://dx.doi.org/10.1016/j.ijforecast.2019.03.017>. – DOI 10.1016/j.ijforecast.2019.03.017. – ISSN 01692070
- [Sobottka 2017] SOBOTTKA, Thomas: *Eine anwendungsorientierte simulationsbasierte Methode, unter Berücksichtigung von Energieeffizienz, in der optimierenden Planung von Produktion und Logistik*. Wien, TU Wien, Dissertation, 2017
- [Sobottka et al. 2019] SOBOTTKA, Thomas ; KAMHUBER, Felix ; FAEZIRAD, Mohammadali ; SIHN, Wilfried: Potential for Machine Learning in Optimized Production Planning with Hybrid Simulation. In: *Procedia Manufacturing* 39 (2019), S. 1844–1853. <http://dx.doi.org/10.1016/j.promfg.2020.01.254>. – DOI 10.1016/j.promfg.2020.01.254. – ISSN 23519789
- [Sobottka et al. 0312] SOBOTTKA, Thomas ; KAMHUBER, Felix ; HENJES, Jan ; SIHN, Wilfried: A case study for simulation and optimization based planning of production and logistics systems. In: *2017 Winter Simulation Conference (WSC)*, IEEE, 03.12.2017 - 06.12.2017. – ISBN 978-1-5386-3428-8, S. 3495–3506
- [Sobottka et al. 2018] SOBOTTKA, Thomas ; KAMHUBER, Felix ; RÖSSLER, Matthias ; SIHN, Wilfried: Hybrid simulation-based optimization of discrete parts manufacturing to increase energy efficiency

- and productivity. In: *Procedia Manufacturing* 21 (2018), S. 413–420. <http://dx.doi.org/10.1016/j.promfg.2018.02.139>. – DOI 10.1016/j.promfg.2018.02.139. – ISSN 23519789
- [Sobottka et al. 2017] SOBOTTKA, Thomas ; KAMHUBER, Felix ; SIHN, Wilfried: Increasing Energy Efficiency in Production Environments Through an Optimized, Hybrid Simulation-based Planning of Production and Its Periphery. In: *Procedia CIRP* 61 (2017), S. 440–445. <http://dx.doi.org/10.1016/j.procir.2016.11.151>. – DOI 10.1016/j.procir.2016.11.151. – ISSN 22128271
- [Spath 2013] SPATH, Dieter (Hrsg.): *Produktionsarbeit der Zukunft - Industrie 4.0: [Studie]*. Stuttgart : Fraunhofer-Verl., 2013. – ISBN 978-3839605707
- [Stadtler 2000] STADTLER, Hartmut: Improved Rolling Schedules for the Dynamic Single-Level Lot-Sizing Problem: Management Science, 46(2), 318-326. In: *Management Science* 46 (2000), Nr. 2, S. 318–326. <http://dx.doi.org/10.1287/MNSC.46.2.318.11924>. – DOI 10.1287/MNSC.46.2.318.11924. – ISSN 0025-1909
- [Stadtler und Kilger 2005] STADTLER, Hartmut (Hrsg.) ; KILGER, Christoph (Hrsg.): *Supply Chain Management and Advanced Planning: Concepts, Models, Software and Case Studies*. Third Edition. Berlin, Heidelberg : Springer Berlin & Heidelberg, 2005. <http://dx.doi.org/10.1007/b106298>. <http://dx.doi.org/10.1007/b106298>. – ISBN 3-540-22065-8
- [Stein 2018] STEIN, Oliver: *Grundzüge der Globalen Optimierung*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2018. <http://dx.doi.org/10.1007/978-3-662-55360-2>. <http://dx.doi.org/10.1007/978-3-662-55360-2>. – ISBN 978-3-662-55359-6
- [Stöcker 2007] STÖCKER, M.: *Untersuchung von Optimierungsverfahren für rechenzeitaufwändige technische Anwendungen in der Motorenentwicklung*. Chemnitz, TU Chemnitz, Diplomarbeit, 2007
- [Sven Grundmann 2007] SVEN GRUNDMANN: *Planung flexibler Produktionskapazitäten im Spannungsfeld logistischer und monetärer Ziele*. Hannover, Universität Hannover, Dissertation, 2007. <https://core.ac.uk/download/pdf/237446165.pdf>
- [Synáková 2017] SYNÁKOVÁ, Lucie: Production Smoothing and Cost Performance in a Production-inventory System. In: *Journal of Competitiveness* 9 (2017), Nr. 1, S. 117–133. <http://dx.doi.org/10.7441/joc.2017.01.08>. – DOI 10.7441/joc.2017.01.08
- [Talbi 2016] TALBI, El-Ghazali: Combining metaheuristics with mathematical programming, constraint programming and machine learning. In: *Annals of Operations Research* 240 (2016), Nr. 1, S. 171–215. <http://dx.doi.org/10.1007/s10479-015-2034-y>. – DOI 10.1007/s10479-015-2034-y. – ISSN 0254-5330
- [Tautrim 2014] TAUTRIM, Jörg: *Lean Administration Taschenbuch: Taschenbuch/Beraterleitfaden: Wesentliche Konzepte und Werkzeuge für mehr Effizienz in der Verwaltung*. Berlin : epubli GmbH, 2014. – ISBN 978-3-7375-2446-9
- [Teerasoponpong und Sopadang 2021] TEERASOPONPONG, Siravat ; SOPADANG, Apichat: A simulation-optimization approach for adaptive manufacturing capacity planning in small and medium-sized enterprises. In: *Expert Systems with Applications* 168 (2021), S. 114451. <http://dx.doi.org/10.1016/j.eswa.2020.114451>. – DOI 10.1016/j.eswa.2020.114451. – ISSN 0957-4174
- [Tegel 2012] TEGEL, Andreas: *Analyse und Optimierung der Produktionsglättung für Mehrprodukt-Fließlinien: Eine Studie zum Lean-Production-Konzept: @Augsburg, Univ., Diss., 2010*. Wiesbaden : Gabler Verlag, 2012 (Produktion und Logistik). <http://dx.doi.org/10.1007/978-3-8349-3587-8>. <http://dx.doi.org/10.1007/978-3-8349-3587-8>. – ISBN 978-3-8349-3587-8
- [Tempelmeier 2005] TEMPELMEIER, Horst: *Produktion und Logistik*. 6., verb. Aufl. Berlin und Köln : Springer, 2005. – ISBN 3-540-23246-X
- [The R Foundation 2021] THE R FOUNDATION: *The R Project for Statistical Computing*. <https://www.r-project.org/>. Version: 2021. Zuletzt geprüft am: 24.05.2021
- [Tratar et al. 2016] TRATAR, L. F. ; MOJŠKERC, B. ; TOMAN, A.: Demand forecasting with four-parameter exponential smoothing. In: *International Journal of Production Economics* 181 (2016), S.

- 162–173. <http://dx.doi.org/10.1016/j.ijpe.2016.08.004>. – DOI 10.1016/j.ijpe.2016.08.004. – ISSN 09255273
- [Tseng und Lin 2009] TSENG, Lin-Yu ; LIN, Ya-Tai: A hybrid genetic local search algorithm for the permutation flowshop scheduling problem. In: *European Journal of Operational Research* 198 (2009), Nr. 1, S. 84–92. <http://dx.doi.org/10.1016/j.ejor.2008.08.023>. – DOI 10.1016/j.ejor.2008.08.023. – ISSN 03772217
- [Ulbrich und Ulbrich 2012] ULBRICH, Michael ; ULBRICH, Stefan: *Nichtlineare Optimierung*. Basel : Imprint Birkhäuser, 2012 (Mathematik Kompakt). <http://dx.doi.org/10.1007/978-3-0346-0654-7>. <http://dx.doi.org/10.1007/978-3-0346-0654-7>. – ISBN 978–3–0346–0142–9
- [Vaillant 2015] VAILLANT, Moritz: *Design Space Exploration zur multikriteriellen Optimierung elektrischer Sportwagenantriebsstränge*. Karlsruhe, Karlsruher Institut für Technologie (KIT), Dissertation, 2015. <http://nbn-resolving.de/urn:nbn:de:0072-506189>
- [van Hop und Tabucanon 2005] VAN HOP, Nguyen ; TABUCANON, Mario T.: Adaptive genetic algorithm for lot-sizing problem with self-adjustment operation rate. In: *International Journal of Production Economics* 98 (2005), Nr. 2, S. 129–135. <http://dx.doi.org/10.1016/j.ijpe.2004.05.016>. – DOI 10.1016/j.ijpe.2004.05.016. – ISSN 09255273
- [Veit 2010] VEIT, Martin: *Wissenschaftliche Berichte des Institutes für Fördertechnik und Logistiksysteme des Karlsruher Instituts für Technologie*. Bd. 74: *Modelle und Methoden für die Bestandsauslegung in Heijunka-nivellierten Supply Chains: Zugl.: Karlsruhe, KIT, Diss., 2010*. Print on demand. Hannover and Karlsruhe : Technische Informationsbibliothek u. Universitätsbibliothek and KIT Scientific Publ, 2010 <http://nbn-resolving.de/urn:nbn:de:0072-187726>. – ISBN 9783866445338
- [Vijini Mallawaarachchi 2020] VIJINI MALLAWAARACHCHI: *8 Useful Tree Data Structures Worth Knowing*. <https://towardsdatascience.com/8-useful-tree-data-structures-worth-knowing-8532c7231e8c>. Version: 2020. Zuletzt geprüft am: 27.02.2022
- [Wagner und Whitin 2004] WAGNER, Harvey M. ; WHITIN, Thomson M.: Dynamic Version of the Economic Lot Size Model. In: *Management Science* 50 (2004), Nr. 12_supplement, S. 1770–1774. <http://dx.doi.org/10.1287/mnsc.1040.0262>. – DOI 10.1287/mnsc.1040.0262. – ISSN 0025–1909
- [Wang und Wu 2003] WANG, Hsiao-fang ; WU, Kuang-yao: Modeling and analysis for multi-period, multi-product and multi-resource production scheduling. In: *Journal of Intelligent Manufacturing* 14 (2003), Nr. 3/4, S. 297–309. <http://dx.doi.org/10.1023/A:1024645608673>. – DOI 10.1023/A:1024645608673. – ISSN 1572–8145
- [Wari und Zhu 2016] WARI, Ezra ; ZHU, Weihang: A survey on metaheuristics for optimization in food manufacturing industry. In: *Applied Soft Computing* 46 (2016), S. 328–343. <http://dx.doi.org/10.1016/j.asoc.2016.04.034>. – DOI 10.1016/j.asoc.2016.04.034. – ISSN 15684946
- [Whitley 2007] WHITLEY, Darrell: *A Genetic Algorithm Tutorial*. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.1052&rep=rep1&type=pdf>. Version: 2007. Zuletzt geprüft am: 27.07.2022
- [Whitley und Sutton 2012] WHITLEY, Darrell ; SUTTON, Andrew M.: Genetic Algorithms — A Survey of Models and Methods. Version: 2012. http://dx.doi.org/10.1007/978-3-540-92910-9_{ }21. In: ROZENBERG, Grzegorz (Hrsg.) ; BÄCK, Thomas (Hrsg.) ; KOK, Joost N. (Hrsg.): *Handbook of natural computing*. Berlin : Springer, 2012 (Springer reference). – DOI 10.1007/978-3-540-92910-9_21. – ISBN 978–3–540–92910–9, S. 637–671
- [Woolsey und Swanson 1975] WOOLSEY, Robert E. ; SWANSON, Huntington S.: *Operations research for immediate application: A quick & dirty manual*. New York : Harper & Row, 1975. – ISBN 978–0060472337
- [Yang 2012] YANG, Xin-She: Nature-Inspired Metaheuristic Algorithms: Success and New Challenges. In: *Journal of Computer Engineering and Information Technology* 01 (2012), Nr. 01, S. 1–3. <http://dx.doi.org/10.4172/2324-9307.1000e101>. – DOI 10.4172/2324–9307.1000e101

- [Yang 2015] YANG, Xin-She: *Studies in computational intelligence*. Bd. 585: *Recent advances in swarm intelligence and evolutionary computation*. Cham : Springer, 2015. – ISBN 978-3-319-13825-1
- [Yannibelli und Amandi 2013] YANNIBELLI, Virginia ; AMANDI, Analía: Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. In: *Expert Systems with Applications* 40 (2013), Nr. 7, S. 2421–2434. <http://dx.doi.org/10.1016/j.eswa.2012.10.058>. – DOI 10.1016/j.eswa.2012.10.058. – ISSN 09574174
- [Yavuz und Tufekci 2007] YAVUZ, Mesut ; TUFEKCI, Suleyman: Analysis and solution to the single-level batch production smoothing problem. In: *International Journal of Production Research* 45 (2007), Nr. 17, S. 3893–3916. <http://dx.doi.org/10.1080/00207540600917542>. – DOI 10.1080/00207540600917542. – ISSN 0020-7543
- [Yi et al. 2018] YI, Jiao-Hong ; DEB, Suash ; DONG, Junyu ; ALAVI, Amir H. ; WANG, Gai-Ge: An improved NSGA-III algorithm with adaptive mutation operator for Big Data optimization problems. In: *Future Generation Computer Systems* 88 (2018), S. 571–585. <http://dx.doi.org/10.1016/j.future.2018.06.008>. – DOI 10.1016/j.future.2018.06.008
- [Yu und Gen 2010] YU, Xinjie ; GEN, Mitsuo: *Decision Engineering*. Bd. 0: *Introduction to Evolutionary Algorithms*. London : Springer-Verlag London, 2010. <http://dx.doi.org/10.1007/978-1-84996-129-5>. <http://dx.doi.org/10.1007/978-1-84996-129-5>. – ISBN 9781849961288
- [Yusoff et al. 2011] YUSOFF, Yusliza ; NGADIMAN, Mohd S. ; ZAIN, Azlan M.: Overview of NSGA-II for Optimizing Machining Process Parameters. In: *Procedia Engineering* 15 (2011), S. 3978–3983. <http://dx.doi.org/10.1016/j.proeng.2011.08.745>. – DOI 10.1016/j.proeng.2011.08.745. – ISSN 18777058
- [Zhou et al. 2011] ZHOU, Aimin ; QU, Bo-Yang ; LI, Hui ; ZHAO, Shi-Zheng ; SUGANTHAN, Ponnuthurai N. ; ZHANG, Qingfu: Multiobjective evolutionary algorithms: A survey of the state of the art. In: *Swarm and Evolutionary Computation* 1 (2011), Nr. 1, S. 32–49. <http://dx.doi.org/10.1016/j.swevo.2011.03.001>. – DOI 10.1016/j.swevo.2011.03.001. – ISSN 22106502
- [Zimmermann 2008] ZIMMERMANN, Hans-Jürgen: *Operations Research: Methoden und Modelle ; für Wirtschaftsingenieure, Betriebswirte, Informatiker*. 2., aktualisierte Aufl. Wiesbaden : Vieweg, 2008 (Aus dem Programm Mathematik für Wirtschaftswissenschaftler). <http://dx.doi.org/10.1007/978-3-8348-9461-8>. <http://dx.doi.org/10.1007/978-3-8348-9461-8>. – ISBN 9783834894618
- [Zitzler et al. 2001] ZITZLER, Eckart (Hrsg.) ; THIELE, Lothar (Hrsg.) ; DEB, Kalyanmoy (Hrsg.) ; COELLO COELLO, Carlos A. (Hrsg.) ; CORNE, David (Hrsg.): *Lecture Notes in Computer Science*. Bd. 1993: *Evolutionary Multi-Criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7-9, 2001 Proceedings*. Berlin and Heidelberg : Springer, 2001. <http://dx.doi.org/10.1007/3-540-44719-9>. <http://dx.doi.org/10.1007/3-540-44719-9>. – ISBN 3-540-41745-1
- [Zupan und Herakovic 2015] ZUPAN, H. ; HERAKOVIC, N.: *Production line balancing with discrete event simulation: A case study*. Bd. 48. 2015. <http://dx.doi.org/10.1016/j.ifacol.2015.06.431>. <http://dx.doi.org/10.1016/j.ifacol.2015.06.431>