

Simulation und Optimierung logistischer Prozesse in der Wärmebehandlung von Stahl

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)
von der Fakultät für
Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie
genehmigte
DISSERTATION

von
Dipl.-Kfm. Eric Ebermann

Tag der mündlichen Prüfung: 2013/01/03
Referent: Herr Prof. Dr. Stefan Nickel
Korreferent: Herr Dr. Magnus Fröhling

Karlsruhe, 2013

Für Anne

Die europäische Stahlindustrie steht gegenwärtig und wohl auch in naher Zukunft vor großen Herausforderungen, um sich auf dem Weltmarkt gegenüber Niedriglohnländern zu behaupten. Neben der Strategie durch Spezialisierung und höchsten Qualitätsanspruch eine Technologieführerschaft einzunehmen, wachsen in den letzten Jahren auch die Anforderungen an eine möglichst schnelle und kostengünstige Produktion. Diese Forderung bietet im Zusammenhang mit der Existenz eines hochkomplexen sowie anlagenintensiven Produktionsprozesses ein breites Feld an Einsatzmöglichkeiten für Instrumente des *Operations Research* in der Stahlindustrie.

Die vorliegende Arbeit wurde in Kooperation mit der AG der Dillinger Hüttenwerke, Europas führendem Hersteller von Qualitätsgrobblechen, angefertigt. Gegenstand der Arbeit sind die in Verbindung mit der Wärmebehandlung der Bleche stehenden logistischen Prozesse. Es konnte festgestellt werden, dass aus der betrachteten, realen Problemstellung ein in der Literatur bislang nicht vergleichbares Maschinenbelegungsproblem resultiert. Neben der mathematischen Beschreibung und Modellierung des grundlegenden Problems zeichnet sich die vorliegende Arbeit vor allem durch die Entwicklung einer speziellen Materialflussimulation und deren Integration in ein gesamtheitliches Optimierungsverfahren aus. Ein weiterer nennenswerter Aspekt ist der hohe Praxisbezug, welcher insbesondere durch die Begleitung der kompletten Implementierung der entwickelten Lösungsmethode in den Realbetrieb der Dillinger Hüttenwerke belegt wird.

Vorwort

Im Rahmen der Erstellung der vorliegenden Arbeit gab es einige Personen, die mir mit Rat und Tat zur Seite standen und denen ich an dieser Stelle dafür meinen Dank aussprechen möchte.

Zunächst gilt mein Dank meinem Doktorvater Prof. Dr. Stefan Nickel, da er mir die Promotion schließlich erst ermöglicht hat. In diesem Zusammenhang möchte ich mich insbesondere für seine Unterstützung und Förderung sowie die Freiheiten, die mir eingeräumt wurden, herzlich bedanken.

Des Weiteren bedanke ich mich bei allen Kolleginnen und Kollegen der Dillinger Hüttenwerke, die mich bei der Promotion unterstützt und sich für mich eingesetzt haben. Sehr verbunden bin ich diesbezüglich meinen beiden Betreuern Frau Dr. Heike Busch und Herrn Christian Hentges. Die interessanten Diskussionen und das hervorragende Arbeitsklima haben mich stets positiv beeinflusst. Zudem gilt mein Dank den vielen Befürwortern meiner Arbeit, insbesondere Herrn Dr. Neis, Herrn Maffert und Herrn Waller. Auch allen Mitarbeitern aus dem Betrieb, mit denen ich im Rahmen meiner Arbeit zu tun hatte, danke ich herzlich für die gute Zusammenarbeit.

Besonderer Dank für die Durchsicht und das Korrekturlesen meines Manuskripts gelten Frau Dr. Ursula-Anna Schmidt sowie Herrn Felix Brandt und Herrn Fabian Dunke.

Von familiärer Seite gebührt meinen Eltern Karlheinz und Madeleine ein herzliches Dankeschön, da sie mir eine sehr gute Ausbildung ermöglicht und mich stets unterstützt haben. Der größte Dank gilt meiner Frau Anne, die mir mit viel Geduld und Liebe immer zur Seite stand, weshalb ich ihr diese Arbeit in tiefster Dankbarkeit und Zuneigung widme. Zuletzt danke ich auch meiner Tochter Amelie, dir mir des Öfteren ein angenehmer Ausgleich zur Arbeit war.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 2 | Materialfluss und Logistik in integrierten Hüttenwerken | 5 |
| 2.1 | Aufbau eines integrierten Hüttenwerks | 5 |
| 2.2 | Die Kokerei | 7 |
| 2.3 | Der Hochofen | 8 |
| 2.4 | Das Stahlwerk | 12 |
| 2.5 | Das Walzwerk | 19 |
| 2.6 | Herausforderungen für die Logistik | 26 |
| 3 | Grundlagen der Wärmebehandlung von Stahl | 35 |
| 3.1 | Zweck und Ziele des Wärmebehandelns | 35 |
| 3.2 | Ablauf des Wärmebehandelns | 37 |
| 3.3 | Wärmebehandlungsverfahren | 40 |
| 3.3.1 | Glühen | 41 |
| 3.3.2 | Härten | 44 |
| 3.3.3 | Vergüten | 46 |

| | | |
|----------|---|-----------|
| 3.4 | Wärmebehandlungsbegriffe, -verfahren und -einrichtungen der Dillinger Hüttenwerke | 48 |
| 4 | Maschinenbelegungsprobleme | 57 |
| 4.1 | Beschreibung und Einordnung | 57 |
| 4.2 | Notationen und Klassifizierung | 61 |
| 4.2.1 | Maschinencharakteristik | 62 |
| 4.2.2 | Auftragscharakteristik | 64 |
| 4.2.3 | Zielcharakteristik | 66 |
| 4.3 | Komplexität und Lösungsverfahren | 68 |
| 4.3.1 | Exakte Verfahren | 71 |
| 4.3.2 | Heuristische Verfahren | 74 |
| 4.4 | Explizite Betrachtung beschränkter Ressourcenkapazitäten . . . | 80 |
| 4.4.1 | <i>Batching</i> -Probleme | 81 |
| 4.4.2 | Das <i>Resource-Constrained Project Scheduling Problem</i> . . | 84 |
| 5 | Beschreibung, Modellierung und Analyse eines realen Ofenbelegungsproblems | 89 |
| 5.1 | Der Rollenherdofen RHO3 | 90 |
| 5.1.1 | Daten und Eigenschaften | 90 |
| 5.1.2 | Fertigungsablauf und Ofensteuerung | 92 |
| 5.2 | Mathematische Problemformulierung | 97 |
| 5.2.1 | Ein zeitdiskretes Modell | 100 |
| 5.2.2 | Ein ereignisbasiertes Modell | 103 |
| 5.3 | Ein intuitives Lösungsverfahren | 107 |

| | | |
|----------|--|------------|
| 5.4 | Analyse des Optimierungspotentials | 110 |
| 5.4.1 | Eine Materialflusssimulation des Ofens | 112 |
| 5.4.2 | Ein <i>Branch&Bound</i> -Verfahren | 116 |
| 6 | Problemerweiterungen und -varianten | 131 |
| 6.1 | Die Verband-Fahrweise | 131 |
| 6.1.1 | Ein Optimierungsmodell für die Verband-Fahrweise . . . | 133 |
| 6.1.2 | Eine Heuristik für die Verband-Fahrweise | 137 |
| 6.2 | Die <i>Hollomon</i> -Fahrweise | 140 |
| 6.2.1 | Analyse der Temperatur-Zeit-Kombinationen | 143 |
| 6.2.2 | Ein Optimierungsverfahren für die <i>Hollomon</i> -Fahrweise . | 146 |
| 6.3 | Einbeziehung des gesamtlogistischen Prozesses | 153 |
| 6.3.1 | Die Stapelproblematik als vorgelagerter Prozess | 154 |
| 6.3.2 | Der Einfluss nachgelagerter Prozesse | 165 |
| 6.3.3 | Anpassung der bestehenden Simulation und Analyse der Auswirkungen | 172 |
| 7 | Konzeption eines gesamtheitlichen Lösungsverfahrens | 183 |
| 7.1 | Idee, Aufbau und Struktur | 184 |
| 7.2 | Der hybride genetische Algorithmus für das HTFSPD (HTFSPD-HGA) | 187 |
| 7.2.1 | Der genetische Algorithmus (HTFSPD-GA) | 188 |
| 7.2.2 | Der <i>Tabu Search</i> (HTFSPD-TS) | 200 |
| 7.2.3 | Rechenergebnisse | 207 |
| 7.3 | Der <i>Online</i> -Aspekt | 210 |

| | | |
|----------|--|------------|
| 7.4 | Temperaturlückenoptimierung | 222 |
| 7.4.1 | Eine neue Ofenzonenregelung | 224 |
| 7.4.2 | Ein Ansatz zur Temperaturlückenvorgabe aus der Wärmelehre | 227 |
| 7.5 | <i>Simulation-3</i> | 237 |
| 8 | Implementierung und Bewertung | 243 |
| 8.1 | Realisierung des entwickelten Konzepts und Einbindung in den Realbetrieb | 243 |
| 8.1.1 | Das Alt-System | 244 |
| 8.1.2 | Das neue System | 246 |
| 8.2 | Quantitative und qualitative Bewertung | 253 |
| 8.2.1 | Numerische Ergebnisse | 254 |
| 8.2.2 | Allgemeine Verbesserungen im Planungs- und Produktionsablauf | 256 |
| 9 | Zusammenfassung | 261 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 2.1 | Ein integriertes Hüttenwerk am Beispiel des Stahlstandorts Dillingen/Saar | 6 |
| 2.2 | Schematische Darstellung einer Koksofenanlage | 7 |
| 2.3 | Materialfluss eines Hochofenwerks | 9 |
| 2.4 | Aufbau einer Hochofenanlage | 10 |
| 2.5 | Stahlwerksprozess | 14 |
| 2.6 | Konverterprozess | 15 |
| 2.7 | Blockguss | 17 |
| 2.8 | Strangguss | 18 |
| 2.9 | Walzwerksprozess | 20 |
| 2.10 | Walzverfahren | 21 |
| 2.11 | Walzenanordnung in einem Walzgerüst | 22 |
| 2.12 | Walztafel | 23 |
| 2.13 | Produktionslogistische Herausforderungen in integrierten Hüttenwerken | 27 |
| 3.1 | Schematischer Zeit-Temperaturverlauf bei der Wärmebehandlung von Stahl | 38 |
| 3.2 | Überblick und Einteilung verschiedener Wärmebehandlungsverfahren | 41 |

| | | |
|------|--|-----|
| 4.1 | Aufgabensicht des Aachener PPS-Modells | 59 |
| 4.2 | Das Modell der Fertigungssteuerung nach Lödding | 60 |
| 4.3 | Ein Analyseschema für <i>Scheduling</i> -Probleme | 69 |
| 4.4 | Eine Klassifizierung von Optimierungsmethoden mit speziellem Hinblick auf Metaheuristiken | 70 |
| 4.5 | Lösung des Beispiels bei Modellierung als <i>Batching</i> -Problem . . | 86 |
| 4.6 | Lösung des Beispiels bei Modellierung als RCPSP | 87 |
| 5.1 | Der Rollenherdofen RHO3 im Walzwerk der Dillinger Hüttenwerke | 90 |
| 5.2 | Eine schematische Darstellung des RHO3 | 93 |
| 5.3 | Beispiel für Wartezeit durch nicht ausreichend freie Ofenkapazität | 94 |
| 5.4 | Beispiel für Wartezeit durch abweichende Ofenverweilzeiten . . . | 95 |
| 5.5 | Temperaturregelung | 96 |
| 5.6 | Beispiel für Wartezeit aufgrund einer Temperaturlücke | 97 |
| 5.7 | Lösung des HTFSPD-Beispiels nach Anwendung der <i>Rule-based</i> <i>Sorting</i> -Heuristik | 109 |
| 5.8 | Bessere Lösung des HTFSPD-Beispiels gegenüber der Lösung der <i>Rule-based Sorting</i> -Heuristik | 110 |
| 5.9 | Ein beispielhafter Knoten des <i>HTFSPD-Branch&Bound</i> - Verfahrens | 117 |
| 5.10 | Beispiel des <i>HTFSPD-Branch&Bound</i> -Verfahrens | 125 |
| 6.1 | Die Verband-Fahrweise | 132 |
| 6.2 | Unerlaubte Verband-Fahrweise | 133 |
| 6.3 | Veranschaulichung der Berechnung des Hollomon-Jaffe- Parameters | 141 |

| | | |
|------|--|-----|
| 6.4 | Beispielhafte Ausgangssituation zur Veranschaulichung der Idee der <i>Hollomon</i> -Fahrweise | 143 |
| 6.5 | Beispielhafte Situation bei Berücksichtigung der <i>Hollomon</i> -Fahrweise | 143 |
| 6.6 | Temperatur-Zeit-Entwicklung für unterschiedliche Ausgangsparameter (bei konstantem <i>Hollomon</i> -Parameter) | 145 |
| 6.7 | Verzweigungsmöglichkeiten eines Knotens aufgrund verschiedener Temperatur-Zeit-Kombinationen | 147 |
| 6.8 | Ein beispielhafter Stapel mit drei umzustapelnden Waben | 155 |
| 6.9 | Beispielhafte Veranschaulichung der <i>Max-Min</i> -Heuristik | 159 |
| 6.10 | Veranschaulichung der Stapelkoordinaten anhand eines Beispielstapels | 162 |
| 6.11 | Vorgehensweise zur Durchsuchung des Lagerbereichs auf freie Stapelplätze | 162 |
| 6.12 | Beispiel für Wartezeit durch nicht ausreichend freie Kapazität auf dem Ofenauslaufrollgang | 166 |
| 6.13 | Das Abkühlverhalten unterschiedlicher Wabendicken | 167 |
| 6.14 | Beispiel für Wartezeit aufgrund eines Engpasses an den Brennanlagen | 169 |
| 7.1 | Gesamtheitliches Lösungskonzept | 185 |
| 7.2 | Der HTFSPD-HGA | 188 |
| 7.3 | Der <i>Order-based Crossover Operator</i> | 195 |
| 7.4 | Der <i>Position-based Crossover Operator</i> | 195 |
| 7.5 | Vier klassische Mutationsoperatoren | 197 |
| 7.6 | Der <i>Hollomon Mutation Operator</i> | 199 |
| 7.7 | Nachbarschaftsbeziehungen im HTFSPD-TS (Beispiel) | 202 |

| | | |
|------|---|-----|
| 7.8 | Die Belegungsoptimierung des RHO3 als <i>Online</i> -Problem | 211 |
| 7.9 | Die Belegungsoptimierung bei vollständiger Information | 212 |
| 7.10 | Die Belegungsoptimierung zum Zeitpunkt 30 | 214 |
| 7.11 | Die Belegungsoptimierung zum Zeitpunkt 90 | 214 |
| 7.12 | Die Belegungsoptimierung zum Zeitpunkt 310 | 215 |
| 7.13 | Die Ofenbelegung zum Zeitpunkt 550 des aus statischer Sicht optimalen Produktionsprogramms | 217 |
| 7.14 | Die Ofenbelegung zum Zeitpunkt 550 des aus dynamischer Sicht vorgeschlagenen Produktionsprogramms | 217 |
| 7.15 | Der <i>Online</i> -Aspekt im Zusammenhang mit der WBH-Blockbildung | 220 |
| 7.16 | Auswirkung einer zu kleinen Temperaturlücke | 223 |
| 7.17 | Vergleich des neuen mit dem ursprünglichen Temperaturregelungsansatzes für zwei beispielhafte Situationen | 226 |
| 7.18 | Zwei parallele Rechteckflächen, wobei eine mit einer Ecke in der Mittelsenkrechten der anderen Fläche liegt | 229 |
| 7.19 | Berechnung der auf Wabenkopf und -fuß bezogenen Einstrahlzahlen bei einem Temperatursprung | 231 |
| 7.20 | Auswirkungen unterschiedlicher Einflussfaktoren auf die Temperaturlückenvorgabe | 236 |
| 8.1 | Startoberfläche des Alt-Systems | 244 |
| 8.2 | Programmerstellung des Alt-Systems | 245 |
| 8.3 | Startoberfläche des neuen Systems | 247 |
| 8.4 | Die neue Programmerstellungsansicht (1) | 248 |
| 8.5 | Die neue Programmerstellungsansicht (2) | 249 |
| 8.6 | Screenshot der Visualisierung | 252 |

| | | |
|-----|---|-----|
| A.1 | Beispiel des <i>HTFSPD-Branch&Bound</i> -Verfahrens | 270 |
| A.2 | Programmerstellung des Alt-Systems | 271 |
| A.3 | Die neue Programmerstellungsansicht (1) | 272 |
| A.4 | Die neue Programmerstellungsansicht (2) | 273 |
| A.5 | Screenshot der Visualisierung | 274 |

Tabellenverzeichnis

| | | |
|-----|---|-----|
| 2.1 | Ausgewählte produktionslogistische Arbeiten aus der Stahlindustrie | 32 |
| 3.1 | Wärmebehandlungsöfen der Dillinger Hüttenwerke | 51 |
| 4.1 | Beispieldaten für ein Maschinenbelegungsproblem mit Kapazitätsbeschränkung | 86 |
| 5.1 | Anlagen- und Produktgeometrie des RHO3 | 91 |
| 5.2 | Technische Daten und Eigenschaften des RHO3 | 92 |
| 5.3 | Ein Beispieldatensatz für das HTFSPD | 109 |
| 5.4 | Vergleich der Rechenergebnisse der (RBS)-Heuristik mit denen des (ZDF)- und des (EBF)-Modells | 111 |
| 5.5 | Vergleich der Rechenergebnisse der (RBS)-Heuristik mit denen des <i>HTFSPD-Branch&Bound</i> -Verfahrens (mit $f_{templ} = 1000$) . . . | 126 |
| 5.6 | Vergleich der Rechenergebnisse der (RBS)-Heuristik mit denen des <i>HTFSPD-Branch&Bound</i> -Verfahrens (mit $f_{templ} = 600$) . . . | 128 |
| 6.1 | Rechenergebnisse des (VF)-Modells | 135 |
| 6.2 | Vergleich der Rechenergebnisse bei Verband-Fahrweise (mit $f_{templ} = 1000$) | 136 |
| 6.3 | Vergleich der Rechenergebnisse bei Verband-Fahrweise (mit $f_{templ} = 600$) | 137 |

| | | |
|------|---|-----|
| 6.4 | Vergleich der Rechenergebnisse der (VF)-Heuristik mit denen des (VF)-Modells | 140 |
| 6.5 | Ein Beispieldatensatz für die <i>Hollomon</i> -Fahrweise | 142 |
| 6.6 | Vergleich der Rechenergebnisse bei <i>Hollomon</i> -Fahrweise (mit $f_{templ} = 1000$) | 151 |
| 6.7 | Vergleich der Rechenergebnisse bei <i>Hollomon</i> -Fahrweise (mit $f_{templ} = 600$) | 152 |
| 6.8 | Vergleich der Rechenergebnisse für <i>Simulation-2</i> (mit $f_{templ} = 1000$ und $AnzBR = 2$) | 177 |
| 6.9 | Vergleich der Rechenergebnisse für <i>Simulation-2</i> (mit $f_{templ} = 1000$ und $AnzBR = 2$) | 178 |
| 6.10 | Vergleich der Rechenergebnisse für Vorwärmer (mit $f_{templ} = 1000$) | 179 |
| 6.11 | Vergleich der Rechenergebnisse für Vorwärmer (mit $f_{templ} = 600$) | 180 |
| 7.1 | Vier Sortierverfahren zur Initialisierung des HTFSPD-GA | 191 |
| 7.2 | Vergleich der Rechenergebnisse des HTFSPD-HGA mit denen des <i>HTFSPD-Branch&Bound</i> -Verfahrens | 208 |
| 7.3 | Vergleich der Rechenergebnisse des HTFSPD-HGA mit denen des <i>HTFSPD-HF-Branch&Bound</i> -Verfahrens | 210 |
| 8.1 | Vergleich der Rechenergebnisse zwischen Alt- und Neu-System | 255 |

Kapitel 1

Einleitung

Die Stahlindustrie stellt eine weltweit stark wachsende Wirtschaftsbranche dar. Im Jahr 2011 wurde mit einer Weltrohstahlproduktion von 1515 Millionen Tonnen ein neuer Höchststand erreicht, woran China mit rund 45% einen sehr großen Anteil hatte. Der Anteil der EU, innerhalb der Deutschland den größten Rohstahlproduzenten bildet, betrug 11,7%. Weltweit liegt Deutschland derzeit auf Platz 7 (Stahl-Zentrum [82]). Die Stahlindustrie nimmt in Deutschland folglich eine bedeutende Rolle ein. Die globale Entwicklung auf dem Stahlmarkt ist in den letzten Jahren allerdings größtenteils durch die Expansion der chinesischen und indischen Stahlindustrie geprägt. Hiervon sind hauptsächlich die Segmente der einfach zu erzeugenden Stahlprodukte betroffen. Für die europäische bzw. deutsche Stahlindustrie galt es daher, sich frühzeitig an die veränderte Marktsituation anzupassen. Um dem hohen Preis- und Wettbewerbsdruck standzuhalten, verfolgen europäische Stahlunternehmen heute daher i. d. R. eine Nischenstrategie, indem sie sich überwiegend auf qualitativ hochwertige und anspruchsvolle Produkte spezialisieren. Zudem liegt der Fokus mehr auf einer kundenspezifischen Produktion, dem *Make-to-Order*-Prinzip, als auf Massenproduktion.

Die vorliegende Arbeit wurde in Kooperation mit der AG der Dillinger Hüttenwerke angefertigt. Das Unternehmen stellt den fünftgrößten Stahlerzeuger Deutschlands (Stahl-Zentrum [81]) dar und hat sich auf den Grobblechmarkt spezialisiert, auf dem es die Technologieführerschaft einnimmt. Bei Grobblech

handelt es sich nach DIN EN 10029 [29] um „ein Flacherzeugnis mit einer Dicke von 3 mm und mehr“.

Auch die Dillinger Hüttenwerke produzieren nach dem *Make-to-Order*-Prinzip. Die Folgen einer derartigen Auftragsfertigung sind, neben den hohen Qualitätsansprüchen, insbesondere stark volatile Auftragseingänge. Dies erfordert i. d. R. einen großen und variablen Produktmix. Zudem sind die Kundenaufträge meist nicht über aufgebaute Lagerbestände zu decken. Aus diesen Umständen heraus entstehen hohe und komplexe Anforderungen an die Produktion sowie an deren Planung und Steuerung. Der allgemein komplexe Herstellungsprozess von Stahlerzeugnissen trägt hierzu ebenfalls bei.

Um ihre Technologieführerschaft und Wettbewerbsfähigkeit zu wahren, verfolgen die Dillinger Hüttenwerke das Ziel der Automatisierung und stetigen Optimierung ihrer Fertigungsprozesse. Aufgrund der anlagen- und transportintensiven Produktion betrifft dies vor allem den Bereich der innerbetrieblichen Logistik. Angestrebt wird hier insbesondere die Erhöhung des Durchsatzes und der Auslastung der bestehenden Anlagen durch geeignete Methoden und Werkzeuge des *Operations Research*.

Diese Zielsetzung wird auch in der vorliegenden Arbeit verfolgt. Sie befasst sich mit einer konkreten Problemstellung, welche den Dillinger Hüttenwerken in der Realität bei der Wärmebehandlung ihrer Stahlerzeugnisse gegenübersteht. Das Hauptaugenmerk liegt hierbei auf einem speziellen Wärmebehandlungssofen mit dem Ziel der Optimierung des Produktionsprogramms sowie dessen Erstellungsprozesses. Aus wissenschaftlicher Sicht ist diese Problemstellung von Interesse, da aus ihr ein spezielles Maschinenbelegungsproblem resultiert, welches in der Literatur bislang noch nicht bekannt ist. In der vorliegenden Arbeit werden sowohl exakte, als auch heuristische Ansätze zur Problemlösung erarbeitet und mit der ursprünglich bei den Dillinger Hüttenwerken angewandten Verfahrensweise verglichen. Für diesen Vergleich wird eine eigens programmierte Materialflusssimulation herangezogen, welche im Rahmen der Entwicklung der verschiedenen Lösungsverfahren zudem als Bewertungsfunktion dient. Die Kopplung von Simulation und Optimierung stellt somit einen zentralen Punkt der vorliegenden Arbeit dar. Hervorzuheben sei an dieser Stelle, dass das entwickelte Lösungsverfahren nicht nur auf Konzeptebene vorgestellt wird, sondern

auch die darüber hinausgehende Implementierung in den Realbetrieb einen wichtigen Bestandteil dieser Arbeit bildet.

Im Folgenden sei der Aufbau der Arbeit kurz dargestellt. Zunächst widmet sich Kapitel 2 der Beschreibung des allgemeinen Materialflusses in integrierten Hüttenwerken und den daraus resultierenden, speziellen, produktionslogistischen Herausforderungen. Kapitel 3 liefert anschließend einen kurzen Überblick über die Grundlagen der Wärmebehandlung von Stahl sowie über die diesbezüglich bei den Dillinger Hüttenwerken zur Verfügung stehenden Anlagen. Mit Kapitel 4 folgt daraufhin eine Einführung in die Theorie der Maschinenbelegungsprobleme. Da es sich bei der in dieser Arbeit vorliegenden Problemstellung um eine spezielle Form eines kapazitätsbeschränkten Maschinenbelegungsproblems handelt, kommt dieser Klasse gegen Ende des Kapitels eine explizite Betrachtung zu. Im Anschluss befasst sich Kapitel 5 mit der Beschreibung der konkreten, realen Problemstellung. Dies beinhaltet sowohl die Abbildung der Ist-Situation, als auch die mathematische Problemformulierung sowie die Erarbeitung einer ersten Simulation und eines exakten Lösungsverfahrens. Schließlich werden die beiden letztgenannten Entwicklungen dazu herangezogen, das Optimierungspotential der aktuellen Verfahrensweise zu analysieren. In Kapitel 6 erfolgt die Einführung und Erläuterung einiger Aspekte, welche jeweils zu einer komplexeren aber auch detaillierteren Betrachtung der Problemstellung führen. Kapitel 7 stellt mit der Entwicklung eines gesamtheitlichen Lösungskonzepts schließlich den Kern der vorliegenden Arbeit dar. Der Beschreibung der betrieblichen Implementierung dieses Konzepts sowie einer abschließenden Gesamtbewertung widmet sich daraufhin Kapitel 8. Den Abschluss der Arbeit bildet Kapitel 9 mit der Zusammenfassung der wichtigsten Punkte, einem Ausblick auf mögliche Weiterentwicklungen für die Zukunft sowie dem Ziehen eines Fazits.

Kapitel 2

Materialfluss und Logistik in integrierten Hüttenwerken

In diesem Kapitel stellt Unterkapitel 2.1 zunächst den Aufbau eines integrierten Hüttenwerks vor. Zudem wird der grobe Materialfluss beschrieben. Die folgenden Unterkapitel 2.2 - 2.5 beschäftigen sich dann intensiver mit den Prozessen innerhalb und zwischen den wichtigsten Betrieben eines derartigen Werks. Neben den eher allgemeingültigen Erläuterungen, wird hier an einigen Stellen bereits Bezug auf die bei den Dillinger Hüttenwerken vorherrschenden Gegebenheiten und Ablaufmechanismen genommen. Den Abschluss bildet Unterkapitel 2.6 mit einem Überblick über die speziellen Herausforderungen für die Logistik in der Stahlindustrie.

2.1 Aufbau eines integrierten Hüttenwerks

Ein klassisches, integriertes Hüttenwerk vereinigt die für die Stahlproduktion notwendigen Betriebe Kokerei, Hochofen, Stahlwerk und Walzwerk an einem gemeinsamen Standort. Abbildung 2.1 skizziert den Aufbau und den Materialfluss eines derartigen Hüttenwerks am Beispiel des Stahlstandorts Dillingen/Saar.

Am Standort Dillingen/Saar produziert die AG der Dillinger Hüttenwerke zu-

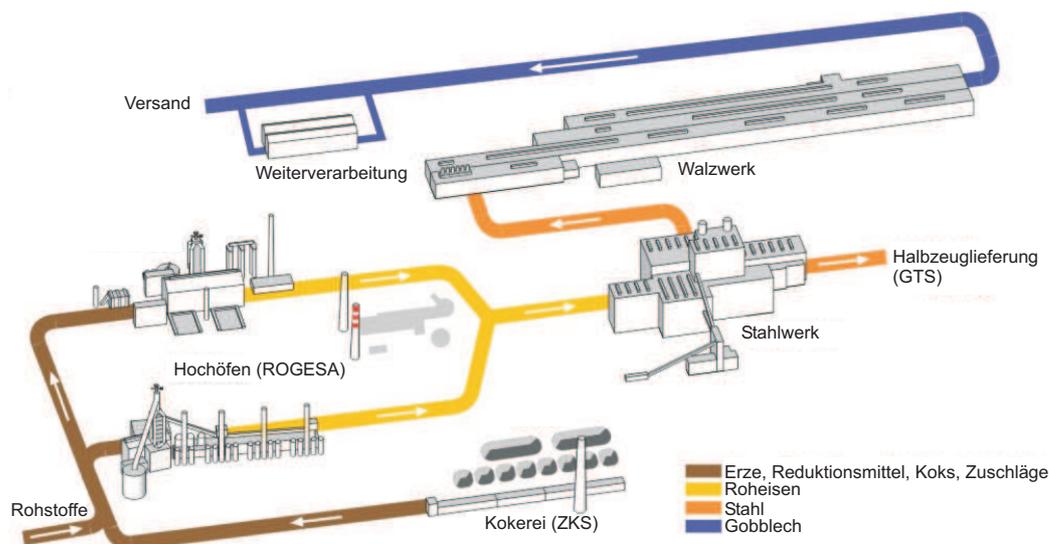


Abbildung 2.1: Ein integriertes Hüttenwerk am Beispiel des Stahlstandorts Dillingen/Saar
(Quelle: AG der Dillinger Hüttenwerke)

sammen mit ihren Tochtergesellschaften, der Roheisengesellschaft Saar (ROGESA) und der Zentralkokerei Saar (ZKS), Grobbleche für den weltweiten Markt. Zu der Gruppe zählt auch ein in Frankreich angesiedeltes Tochterwalzwerk, GTS Industries.

In der Kokerei wird Koks hergestellt, der zusammen mit Erzen sowie weiteren Reduktionsmitteln (z. B. Öl) und Zuschlagstoffen (z. B. Kalk) in die Hochöfen eingebracht wird. Dort findet eine Reduktion der in den Erzen enthaltenen Eisenoxide statt, wodurch flüssiges Roheisen gewonnen wird. Es enthält noch hohe Anteile an unerwünschten Begleitelementen (z. B. Kohlenstoff). Diese werden im Konverterprozess des Stahlwerks verbrannt. Darüber hinaus werden die gewünschten Stahlanalysen durch verschiedene sekundärmetallurgische Maßnahmen eingestellt. Im Anschluss wird der flüssige Stahl vergossen und so in bestimmte Formen, Gewichte und Abmessungen gebracht. Die weitere Formgebung erfolgt durch das Walzen im Walzwerk, wodurch sich das Gefüge und damit die mechanischen Eigenschaften ändern. Da der erforderliche, endgültige Gefügestand hierdurch meist noch nicht erzielt wird, schließt sich häufig eine (oft auch mehrere) nachträgliche Wärmebehandlung an. Je nach Endprodukt folgen abschließend weitere Umformungsprozesse, Schneidvorgänge oder sonstige Weiterbehandlungen. Nicht selten sind weiterverarbeitende Betriebe direkt auf dem Hüttengelände angesiedelt.

2.2 Die Kokerei

In einer Kokerei entsteht das wichtigste Reduktionsmittel für den Hochofenprozess, der Koks. Lueger [63] definiert Koks als „den beim Erhitzen gewisser Arten von Steinkohlen unter annäherndem oder vollständigem Luftabschluß verbleibenden festen, mehr oder weniger dichten Rückstand“. Die Erhitzung dient dem Zweck der Austreibung unerwünschter, flüchtiger Bestandteile. Zum Einsatz kommen hauptsächlich schwefelarme Steinkohlen, die zudem über bestimmte Backfähigkeiten verfügen.

Der Prozess des Verkokens erfolgt in Koksöfen, die gewöhnlich zu einer sogenannten Koksofenbatterie zusammengefasst sind. Lueger [63] unterscheidet hierbei Koksflammöfen und Destillationsöfen. Während die bei der Verkokung entstehenden Nebenprodukte in Koksflammöfen entweichen, werden sie in Destillationsöfen gezielt abgefangen. So können beispielsweise Teer, Schwefelwasserstoff, Ammoniak und Benzol, welche hauptsächlich in der chemischen Industrie Verwendung finden, gewonnen werden. Das zudem anfallende Koksofengas kann zur Unterfeuerung der Kokerei sowie zur Stromerzeugung in einem Kraftwerk genutzt werden.

Das Stahl-Institut VDEh [80] beschreibt den Betriebsablauf einer Kokerei anhand einer schematischen Darstellung einer Koksofenanlage (siehe Abbildung 2.2). Die angelieferte Rohkohle wird zunächst in Bunkern gelagert und für den

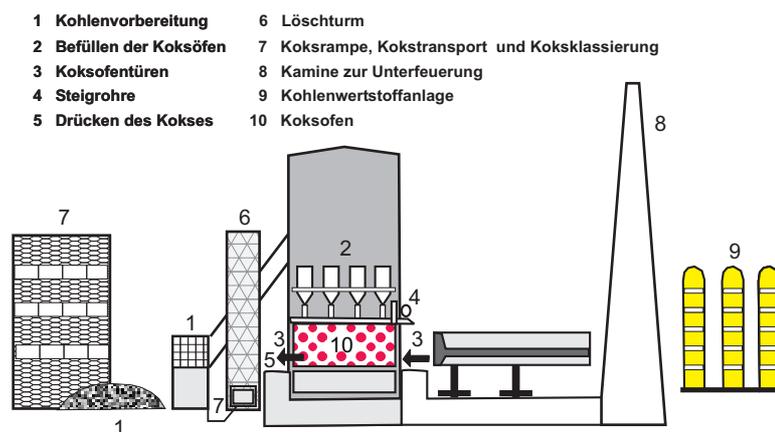


Abbildung 2.2: Schematische Darstellung einer Koksofenanlage (in Anlehnung an Stahl-Institut VDEh [80])

Prozess auf- und vorbereitet. In regelmäßigen Abständen wird ein Füllwagen befüllt, der die Kohle an die Decke des Horizontalkammerofens befördert. Dort erfolgt der Schüttvorgang in die einzelnen Koksöfen und die Erhitzung unter Luftabschluss beginnt. Nach Abschluss der Verkokung (dauert etwa 16 bis 20 Stunden) werden die Koksofentüren geöffnet. Danach stoßen Ausdrückmaschinen den ausgegarten Koks wieder auf einen Füllwagen. Schließlich wird der Koks in einem Löschurm mit Wasser abgekühlt und gelöscht. Häufig befindet sich auf Kokereigeländen zusätzlich eine Kohlenwertstoffanlage, deren Aufgabe die möglichst effektive Gewinnung der bei der Verkokung anfallenden Nebenprodukte ist.

2.3 Der Hochofen

Der Weg zum Rohstahl kann grundsätzlich über zwei Verfahrensrouten realisiert werden. Auf der einen Seite gibt es den zweistufigen Weg über die Erzeugung von flüssigem Roheisen im Hochofen mit anschließender Umwandlung in speziellen Konvertergefäßen oder Herdöfen. Auf der anderen Seite kann Stahl auch durch Einschmelzen von Stahlschrott bzw. direkt reduziertem Eisen im Elektrolichtbogenofen hergestellt werden. Diese Arbeit konzentriert sich ausschließlich auf die Route über den Hochofen und den Konverter.

Bevor die Rohstoffe und Reduktionsmittel in den Hochofen eingebracht werden können, sind teilweise sehr umfangreiche Vorbehandlungen nötig. Abbildung 2.3 veranschaulicht den komplexen Materialfluss rund um den Hochofen. Der Koks aus der Kokerei sowie die Zuschlagstoffe (z. B. Kalk und Quarzsand) und zusätzliche Kreislaufstoffe werden i. d. R. auf Güterwaggons angeliefert. Feinerze, Stückerze und Pellets nehmen eher den Weg über das Wasser. Hochofenwerke verfügen daher meist über einen eigenen oder zumindest nahegelegenen Hafen.

Nach der Herstellung des Kokses in der Kokerei wird dieser zunächst gesiebt. Grob- und Kleinkoks können anschließend direkt in den Bunkeranlagen des Hochofens eingelagert werden. Der feinkörnige Koksgrus hingegen wird zunächst mit den Feinerzen sowie mit den Zuschlag- und Kreislaufstoffen auf

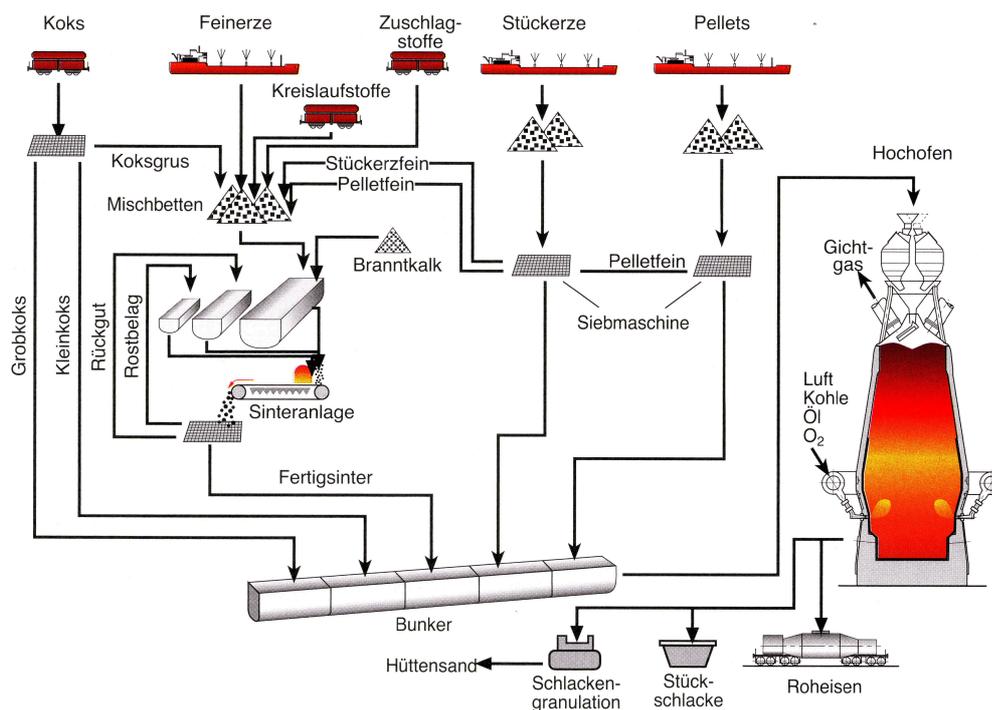


Abbildung 2.3: Materialfluss eines Hochofenwerks
(Stahl-Institut VDEh [80])

Mischbetten lagenweise übereinander geschichtet. Durch diese Lagerungsmethode und das anschließende Abgraben mit großen Schaufelbaggern wird versucht, einen möglichst hohen Homogenisierungsgrad zu erzielen. Ähnlich wie der Koks werden auch Stückerze und Pellets zunächst gesiebt und anschließend teilweise direkt zu den Bunkeranlagen transportiert oder den Mischbetten zugeführt.

Der nächste Prozessschritt liegt darin, das Mischgut von den Mischbetten zusammen mit Branntkalk in eine Sinteranlage zu befördern. Dort werden die feinkörnigen Rohstoffe auf einem Sinterband zusammengebacken und somit stückig gemacht. Anschließend wird der so entstandene Sinter gesiebt, um möglichst gleich große Stücke zu erhalten. Sind die Stücke zu groß, müssen sie zerkleinert und wieder gesiebt werden. Zu kleine Stücke hingegen müssen rückgeführt werden und das Sinterband wieder in Form von Rückgut durchlaufen. Ist die gewünschte Stückigkeit erreicht, gelangt auch der Fertigsinter in die Bunkeranlagen des Hochofens.

In den Bunkeranlagen sind somit alle für den Hochofenprozess benötigten Ein-

satzstoffe jeweils getrennt gelagert. Als nächster Schritt erfolgt die Beschickung des Hochofens. Über Förderbänder gelangen die Einsatzstoffe in genau dosierten Mengen mit Hilfe von Begichtungsanlagen in den Ofen. Das Haupterzeugnis eines Hochofens ist flüssiges Roheisen, welches zur Herstellung von Stahl dient. Prozessbedingt fallen jedoch noch weitere Nebenerzeugnisse wie Gichtgas und Stückschlacke an, die anderweitige Verwertungsmöglichkeiten bieten. So kann Gichtgas beispielsweise für verschiedene Heizzwecke und Stückschlacke beim Straßenbau verwendet werden. Die Hochofenschlacke kann jedoch auch zu Hüttensand granuliert werden, welcher überwiegend in der Zementindustrie zum Einsatz kommt.

An den Überblick über den Materialfluss und die wichtigsten Prozesse rund um den Hochofen schließt sich nun eine detailliertere Beschreibung des Ofens und der Vorgänge innerhalb des Ofens an. Abbildung 2.4 veranschaulicht den Aufbau einer Hochofenanlage. Winderhitzer dienen der Vorwärmung der Reaktionsluft vor der Einspeisung in den Hochofen. Eine Hochofenanlage ist im Allgemeinen mit drei Winderhitzern ausgestattet, die diskontinuierlich zwischen Aufheiz- und Blasbetrieb wechseln. Während immer zwei Winderhitzer im Aufheizbetrieb das Gichtgas aus dem Hochofen unter Luftzufuhr verbrennen, wird der dritte durch Einblasen von Kaltluft abgekühlt. Dieser gibt somit

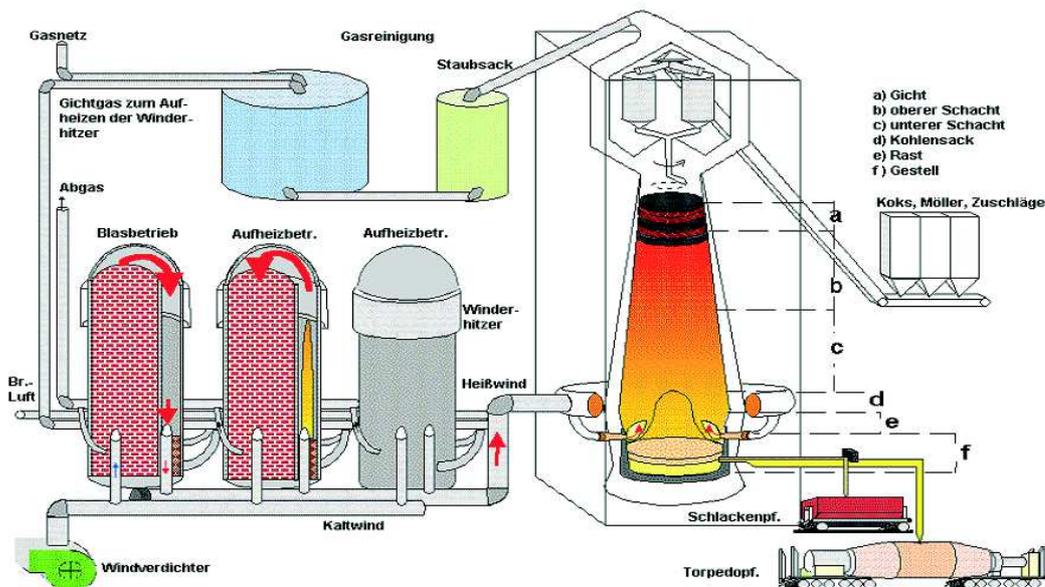


Abbildung 2.4: Aufbau einer Hochofenanlage
(Quelle: AG der Dillinger Hüttenwerke)

seine Wärme ab, die als Heißwind in den Hochofen geleitet wird.

Der Hochofen selbst ist ein 30 bis 50 Meter hoher Schachtofen, der von einem Stahlgerüst umgeben ist. Der Ofen arbeitet nach dem Gegenstromprinzip, d. h. das Füllgut durchläuft den Ofen von oben nach unten während ihm das Hochofengas von unten nach oben entgegenströmt.

Das Innenprofil des Ofens kann in verschiedene Zonen unterteilt werden. Den oberen Teil des Ofens, wo das Beschickungsgut eingebracht wird, bezeichnet man auch als Gicht. Um ein Entweichen des im Ofen befindlichen Gichtgases zu vermeiden, ist dieser Teil mit einem Doppelverschluss versehen. Beim Einfüllen wird zunächst nur der obere Verschluss geöffnet und anschließend wieder geschlossen. Das aufkommende Gichtgas wird über Gasreinigungseinrichtungen abgeführt und in das allgemeine Gasnetz eingespeist bzw. zur Erhitzung der Winderhitzer verwendet. Erst nach erfolgtem Druckausgleich wird der untere Gichtverschluss geöffnet, wodurch das Füllgut in den Schacht gelangt. Vom oberen bis zum unteren Schacht steigt die Temperatur von ca. 200 °C auf ca. 1000 °C an. Zudem erweitert sich der Durchmesser des Ofens, da sich das Füllgut infolge der Erwärmung ausdehnt. Es folgt der breiteste Teil des Hochofens, der Kohlensack, mit Temperaturen von ca. 1500 °C. Im darunter liegenden Teil, der Rast, verengt sich der Ofen wieder. Hier erreicht der Ofen Temperaturen von über 2000 °C und der Schmelzvorgang beginnt. Den untersten Teil des Ofens bildet das sogenannte Gestell, wo sich das flüssige Roheisen und die Schlacke sammeln.

Hiersig [46] fasst die wichtigsten Vorgänge im Ofen phasenweise zusammen:

1. Trocknen, Vorwärmen sowie Austreiben des Hydratwassers,
2. indirekte Reduktion,
3. direkte Reduktion,
4. Schmelzen.

Die erste Phase spielt sich im oberen Schacht ab, wo das Gas seine Wärme abgibt, die Trocknung und Vorwärmung des Füllguts stattfindet und das chemisch gebundene Wasser (Hydratwasser) ausgetrieben wird. Im unteren Schacht vollzieht sich dann die zweite Phase, die indirekte Reduktion der Ei-

senoxide durch Kohlenmonoxid und Wasserstoff. In der dritten Phase beginnen die bis dahin noch nicht zu Eisen reduzierten Eisenoxide im Kohlensack bei über 1000 °C direkt zu metallischem Eisen zu reduzieren. Nach Abschluss der Schmelzphase ist die Reduktion beendet.

Der Roheisen- und Schlackeabstich erfolgt in regelmäßigen Abständen von etwa 2 bis 4 Stunden. Mit Hilfe von speziellen Waggonen werden Roheisen und Schlacke an ihren jeweiligen Zielort abtransportiert. Für die Schlacke stehen hierfür z. B. sogenannte Schlackenpfannenwaggonen zur Verfügung. Der Transport des Roheisens zur Weiterverarbeitung im nahegelegenen Stahlwerk erfolgt in Torpedopfannenwaggonen.

2.4 Das Stahlwerk

Die Hauptaufgabe eines jeden Stahlwerks liegt in der Produktion von Stahl. Nach DIN EN 10020 [28] bezeichnet man als Stahl „alle Werkstoffe, deren Massenanteil an Eisen größer ist als der jedes anderen Elements und die im Allgemeinen weniger als 2,0 % Kohlenstoff aufweisen und noch andere Elemente enthalten“. Demgegenüber steht die Definition von Roheisen nach DIN EN 10001 [27] als „Legierung aus Eisen mit mehr als 2% Kohlenstoff“. Um aus dem eher spröden Roheisen schmiedbaren Stahl herzustellen, muss demnach der Kohlenstoffgehalt gesenkt werden. Zudem müssen störende Begleitelemente (wie Phosphor, Schwefel, Silizium oder Sauerstoff) auf niedrigere Restgehalte reduziert bzw. weitestgehend entfernt werden. Dieser Prozess, der auch als Frischen bezeichnet wird, kommt einer Verbrennung (Oxidation) gleich.

Gobrecht und Rumpler [41] unterscheiden die verschiedenen Frischverfahren nach zwei Kriterien, zum einen nach der Ofenart und zum anderen nach eingesetztem Oxidationsmittel. Nach der Ofenart werden demnach

- Herdfrischverfahren,
- Konverterverfahren und
- Elektrolichtbogenverfahren

unterschieden. Beim Herdfrischverfahren wird ein wannenförmiger Herdofen

mit großer Schmelzoberfläche eingesetzt. Das Verfahren wird auch als Siemens-Martin-Verfahren bezeichnet. Konverterverfahren verwenden dagegen feuerfest ausgekleidete, birnenförmige Gefäße. Zu diesen Verfahren zählen beispielsweise das Bessemer-Verfahren, das Thomas-Verfahren oder das Sauerstoffblasverfahren, welche sich hauptsächlich bezüglich Konvertergefäßgröße und Verfahrensprinzip voneinander unterscheiden. Bei der dritten Methode, den Elektrolichtbogenverfahren, wird die Wärme dagegen in sogenannten Lichtbogenöfen durch elektrische Energie erzeugt.

Bezüglich des eingesetzten Oxidationsmittels lassen sich zudem

- Windfrischverfahren und
- Sauerstoffblasverfahren

unterscheiden. Während Windfrischverfahren Luft verwenden, wird bei den Sauerstoffblasverfahren reiner Sauerstoff auf die Schmelze (Sauerstoffaufblas- bzw. Linz-Donawitz (LD)-Verfahren) oder von unten durch die Schmelze (Sauerstoffbodenblas bzw. Oxygen-Bottom-Maxhütte (OBM)-Verfahren) geblasen.

Aufgrund steigender Qualitätsanforderungen sowie zunehmender Konkurrenz ist eine Nachbehandlung des Stahls (nach dem Frischen) in den meisten Stahlwerken heutzutage nahezu unumgänglich. Die dabei eingesetzten metallurgischen Maßnahmen können unter dem Begriff Sekundärmetallurgie zusammengefasst werden. Im Anschluss an das Frischen und die sekundärmetallurgische Behandlung muss der flüssige Rohstahl noch vergossen und somit in einen erstarrten Zustand überführt werden. Abbildung 2.5 veranschaulicht die wichtigsten Prozessschritte am Beispiel des Stahlwerks der AG der Dillinger Hüttenwerke.

Wie heutzutage bei den meisten Stahlproduzenten (ca. 65 % der Weltrohstahlproduktion, vgl. Stahl-Institut VDEh [80]) wird auch im Stahlwerk der Dillinger Hüttenwerke zum Frischen das Sauerstoffblasverfahren (bzw. das Sauerstoffaufblasverfahren) angewandt. Da der Schwefelabbau im Konverter nur begrenzt möglich oder mit sehr hohem Aufwand verbunden ist, wird das Roheisen häufig zunächst einer Vorentschwefelung in der Roheisenpfanne unterzogen. Hierbei werden Elemente, die sich leicht mit Schwefel verbinden, mittels einer Lanze in das Roheisen eingeblasen. In der sich absetzenden Schlacke lässt

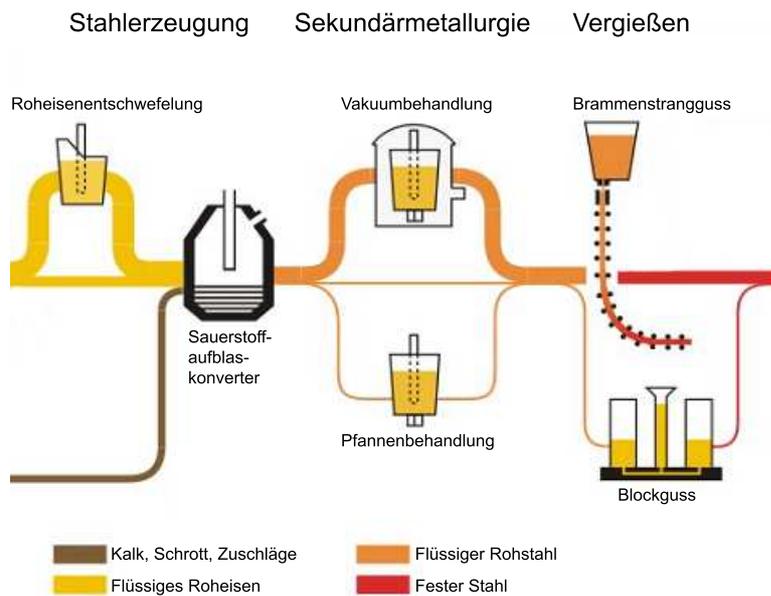


Abbildung 2.5: Stahlwerksprozess
(Quelle: AG der Dillinger Hüttenwerke)

sich somit ein Großteil des Schwefels als Sulfid binden. Diese Schlacke kann nun entfernt und das Roheisen anschließend direkt in den Konverter gefüllt werden.

Das Stahlwerk der Dillinger Hüttenwerke verfügt über zwei Konverter, in denen jeweils Chargen von ca. 185 t Rohstahl produziert werden. Neben dem Roheisen mit ca. 140 - 160 t besteht eine Charge noch aus ca. 40 - 50 t Schrott. Als Schlackenbildner werden zudem ca. 10 t gebrannter Stückkalk hinzugegeben. Das Stahl-Institut VDEh [80] beschreibt Aufbau und Prozess innerhalb eines Sauerstoffaufblaskonverters anhand von Abbildung 2.6.

Mit einem Druck von bis zu 12 bar wird der reine Sauerstoff durch eine wassergekühlte Lanze auf das Metallbad geblasen. Infolgedessen oxidiert das Eisen zu Eisenoxid und der Kohlenstoff zu Kohlenmonoxid. Durch den hohen Druck und die anhaltende Entwicklung von aufsteigendem Kohlenmonoxid wird das Roheisen ständig bewegt und durchmischt. Zudem bildet sich aus den oxidierenden Begleitelementen eine Schlackenschicht, die auf dem Metallbad schwimmt. Nach Beendigung des Blasens werden zunächst Proben entnommen und geprüft, ob ein Nachblasen nötig ist. Ist dies nicht der Fall, ist der Frischprozess beendet und der Abstich kann erfolgen. Hierzu wird der Konverter gekippt und

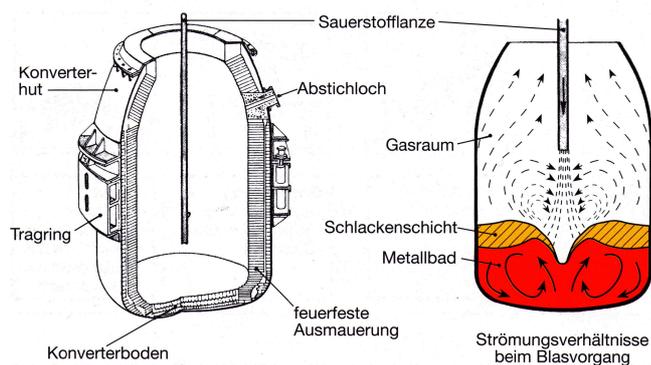


Abbildung 2.6: Konverterprozess
(Stahl-Institut VDEh [80])

der Stahl kann durch das Abstichloch in eine Gießpfanne fließen. Aufgrund seiner geringeren Dichte bleibt die Schlacke zu Beginn des Abstichs zunächst im Konverter. Um ein Mitlaufen der Schlacke gegen Ende des Abstichs zu vermeiden, muss der Konverter rechtzeitig wieder zurückgeschwenkt werden. Das Abkippen der Schlacke erfolgt dann meist über den Konverterrund auf der dem Abstichloch gegenüberliegenden Seite in eine Schlackenpfanne bzw. in einen Schlackenkübel.

Im Laufe der letzten ca. 40 Jahre sind mehrere Varianten des Sauerstoffblasverfahrens entwickelt worden, die zumeist in Mischformen zwischen dem LD- und dem OBM-Verfahren resultieren. So werden auch bei der Dillinger Hütte heute neben dem Aufblasen des Sauerstoffs außerdem Inertgase (z. B. Stickstoff und Argon) durch Bodendüsen von unten in die Schmelze eingeblasen. Mit Hilfe dieses Verfahrens, das auch als *Lance-Bubbling-Equilibrium* (LBE)-Verfahren bezeichnet wird, lässt sich die Geschwindigkeit der metallurgischen Reaktionen im Konverter steigern.

Wie bereits erwähnt, werden zwischen dem Frischprozess und dem Vergießen häufig noch sekundärmetallurgische Maßnahmen an der Schmelze durchgeführt. Bolbrinker et al. [11] beschreiben die Sekundärmetallurgie als „Oberbegriff für alle Verfahren zur Desoxidation, Entgasung, Legierungsfeineinstellung, Einschlusseinformung und Reinheitsgradverbesserung von Stahl“. Bei einem Großteil dieser Verfahren wird die Schmelze direkt nach dem Abstich in der Pfanne behandelt. Daher wird die Sekundärmetallurgie häufig auch als Pfannenmetallurgie bezeichnet. Bargel et al. [5] unterteilen die wesentlichen

Verfahren in

- Injektionsverfahren,
- Vakuumverfahren und
- Umschmelzverfahren.

Bei den Injektionsverfahren werden Reaktionsmittel (wie Kalk, Calcium-Silizium oder Magnesium) mittels einer Lanze eingeblasen. Das Ziel besteht darin, vor allem den für hochqualitative Stähle immer noch zu hohen Schwefelgehalt des Stahls wiederholt zu senken. Auch das Einblasen von Legierungselementen, um bestimmte Stahleigenschaften zu erzielen, ist möglich.

Eine sehr gebräuchliche, weil auch effektive, Methode zur Nachbehandlung des Stahls bildet die Gruppe der Vakuumverfahren. Ihr Vorteil besteht darin, dass gelöste Gase unter Vakuum wesentlich besser entweichen können als unter Atmosphärendruck. Neben der Entgasung werden häufig weitere sekundärmetallurgische Maßnahmen wie die Feinentkohlung, das Feinlegieren oder Methoden zur Verbesserung des Reinheitsgrades im Vakuum durchgeführt.

Das Prinzip der Umschmelzverfahren liegt darin, bereits erstarrten Stahl wieder aufzuschmelzen und anschließend wieder zur Erstarrung zu bringen. Dabei findet das Umschmelzen entweder in einer flüssigen Schlackenschicht oder unter Vakuum statt. Umschmelzverfahren sind meist sehr aufwändig, können aber einen sehr hohen Reinheitsgrad erzielen.

Im Stahlwerk der Dillinger Hüttenwerke erfolgt heute im Anschluss an den Konverter schon während des Abstichs eine Zugabe von Legierungselementen (Groblegierung) in der Gießpfanne. Darauf folgt eine Lanzenspülung mit Argon zur Homogenisierung der Schmelze. Der Großteil der Produktion wird anschließend in einer von drei Vakuumanlagen zur Pfannenentgasung nachbehandelt. Zudem erfolgt hier eine Tiefstentschwefelung und das Feinlegieren der Schmelze. Nach weiteren Behandlungen unter Normaldruck (wie das Legieren und Drahteinspulen oder das Reinheitsgradspülen) gelangt der noch immer flüssige Stahl in den Gießbetrieb, wo er durch Vergießen seine erste feste Form erhält.

Bis in die 1970er Jahre war es üblich, den Stahl portionsweise in gusseisernen

Dauerformen (Standkokillen) zu vergießen. Dieses als Blockgießen bezeichnete Gießverfahren wird heute nur noch für kleinere Chargen spezieller Stähle durchgeführt. Weitestgehend abgelöst wurde das Blockgießen durch das Stranggießverfahren, bei dem der Stahl in einem kontinuierlichen Strang abgegossen wird. Die Dillinger Hüttenwerke vergießen heute ca. 6 % ihres Stahls mit dem Blockgieß- und ca. 94 % mit dem Stranggießverfahren. Die beiden Verfahrensweisen werden im Folgenden näher erläutert.

Gobrecht und Rumpler [41] unterscheiden beim Blockgießverfahren zwei Methoden, den steigenden und den fallenden Guss. Abbildung 2.7 zeigt eine schematische Darstellung der beiden Verfahren.

Beim fallenden Guss (in Abbildung 2.7 rechts) fällt der flüssige Stahl aus der Pfanne unmittelbar in die auf einer Bodenplatte stehende Kokille. Im Gegensatz dazu fließt der Stahl beim steigenden Gussverfahren (in Abbildung 2.7 links) zunächst durch einen Gießtrichter und verteilt sich dann durch Kanäle in der Gespannplatte auf mehrere Kokillen, die sich schließlich aufsteigend füllen. Die Dillinger Hüttenwerke nutzen, wie die meisten Stahlwerke heutzutage, das steigende Blockgussverfahren. Das Abgießen des Stahls erfolgt in Dillingen jedoch nicht mittels einer Stopfenstange, sondern durch eine Schieberplatte am Pfannenboden.

Nach beendeter Erstarrung des Stahls wird die leicht konische, nach unten

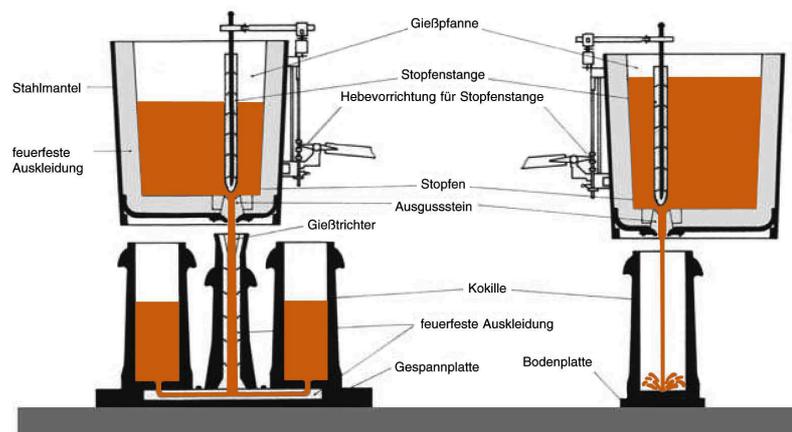


Abbildung 2.7: Blockgießen
(in Anlehnung an Gobrecht und Rumpler [41])

offene Kokille mittels eines Krans abgezogen. Dieser Prozess wird auch als Strippen bezeichnet. Der Abtransport des noch glühenden Stahlblocks erfolgt wiederum per Kran.

Das Stranggießverfahren hat gegenüber dem Blockgießverfahren eindeutige Vorteile, insbesondere im Hinblick auf Qualität und Ausbringungsmenge. Anhand von Abbildung 2.8 lässt sich der Ablauf dieses Verfahrens näher erläutern. Auf der rechten Seite ist der grobe Aufbau einer Stranggießanlage schematisch dargestellt. Dabei sei erwähnt, dass solche Anlagen eine Gesamthöhe von über 40 Metern besitzen können. Auf der linken Seite von Abbildung 2.8 sieht man einen vergrößerten Ausschnitt des Verteilergefäßes und einer Kokille.

Zunächst gelangt der flüssige Stahl aus der Gießpfanne unter Luftabschluss in das Verteilergefäß, auch *Tundish* genannt. Von dort aus fließt er über Tauchrohr in die wassergekühlten Kupferkokillen der einzelnen Stränge. Dabei lässt sich die Durchflussmenge über eine Stopfenstange regulieren. Weiter unten erfolgt eine intensive Kühlung des Stranges, wodurch er schließlich von außen nach innen erstarrt. Während dieses Erstarrungsvorgangs muss der Strang durch ein Rollengerüst von allen Seiten abgestützt werden. Im unteren Bereich der Anlage wird der erstarrte, aber noch weiche Strang mit Hilfe von Biegerollen in die Horizontale abgebogen und schließlich auf Brennschneidanlagen in beliebige Längen geschnitten.

Die hier beschriebene, in Abbildung 2.8 (rechts) aufgezeigte Stranggießanlage

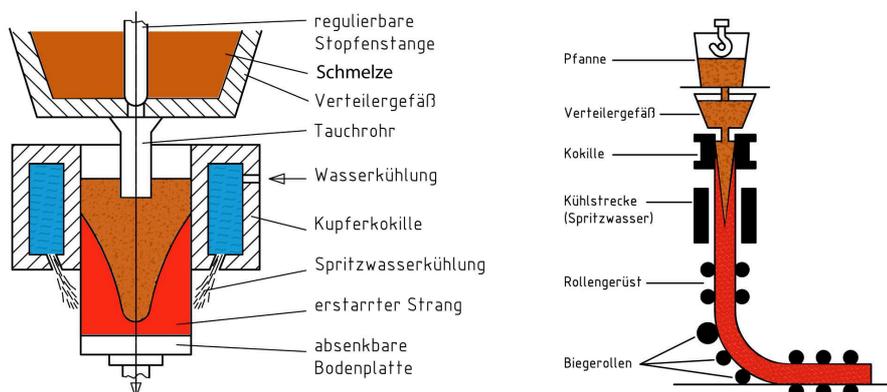


Abbildung 2.8: Stranggießen
(in Anlehnung an Gobrecht und Rumpler [41])

ge wird auch als Senkrecht-Abbiege-Anlage bezeichnet. Es gibt jedoch auch andere Bauarten wie die reine Senkrecht-Anlage, die Anlage mit progressiver Abbiegung oder auch die Kreisbogen-Anlage (siehe Stahl-Institut VDEh [80]). Die Dillinger Hüttenwerke verfügen für das Vergießen von Stahl über drei Senkrecht-Abbiege-Anlagen. Der Standort Dillingen hat im Bereich des Stranggießens schon immer eine weltweite Vorrangrolle eingenommen. 1961 wurde hier die erste (Brammen-)Stranggießanlage der Welt in Betrieb genommen. Heute produziert die Dillinger Hütte auf der modernsten Anlage mit 450 mm die weltweit dicksten Formate.

Der nach dem Stranggießverfahren gegossene Stahl wird als Bramme oder Rohbramme, der nach dem Blockgießverfahren gegossene Stahl als Block oder Rohblock bezeichnet. Bolbrinker et al. [11] definieren eine Bramme als „urformten Stahlquader, dessen Breite mindestens das Zweifache der Dicke beträgt“. In Dillingen gelangen die Brammen über den Ausförderrollgang der Stranggießanlagen per Kran in die nahegelegene, dem Walzwerk vorgelagerte Brammenadjustage. Mit gleichem Ziel, jedoch auf Tiefladern, erfolgt der Abtransport der Rohblöcke. In der Brammenadjustage werden die Vormaterialien für das Walzwerk verschiedenen Kontrollmechanismen unterzogen und für die spätere Walzung aufbereitet (z. B. geflämmt, geschnitten und geschliffen). Etwa zwei Drittel der Brammen werden im Dillinger Walzwerk, die restlichen im in Nordfrankreich gelegenen Tochterwalzwerk gewalzt. Der Transport nach Nordfrankreich erfolgt per Bahn.

2.5 Das Walzwerk

Nach dem Verlassen des Stahlwerks ist der Urformprozess und damit die erste Formgebung des Stahls beendet. Im Walzwerk schließen sich nun das Umformen durch Walzen sowie weitere Zurichtungs- und Nachbehandlungsverfahren an. Dabei unterscheiden sich Anlagen und Prozesse innerhalb von Walzwerken je nach produziertem Stahlerzeugnis sehr stark voneinander. Eine Übersicht über verschiedene Walzstahlerzeugnisse bietet das Stahl-Institut VDEh [80]. Eine Beschreibung aller Produktionsprozesse ist aufgrund der hohen Anzahl an unterschiedlichen Erzeugnissen in dieser Arbeit nicht vorgesehen. Da sich diese

Arbeit hauptsächlich mit einer konkreten, im Walzwerk der Dillinger Hüttenwerke auftretenden Problemstellung befasst, konzentrieren sich die folgenden Erläuterungen auf die Grobblechproduktion. Materialfluss und Produktionsprozess werden durch Abbildung 2.9 veranschaulicht.

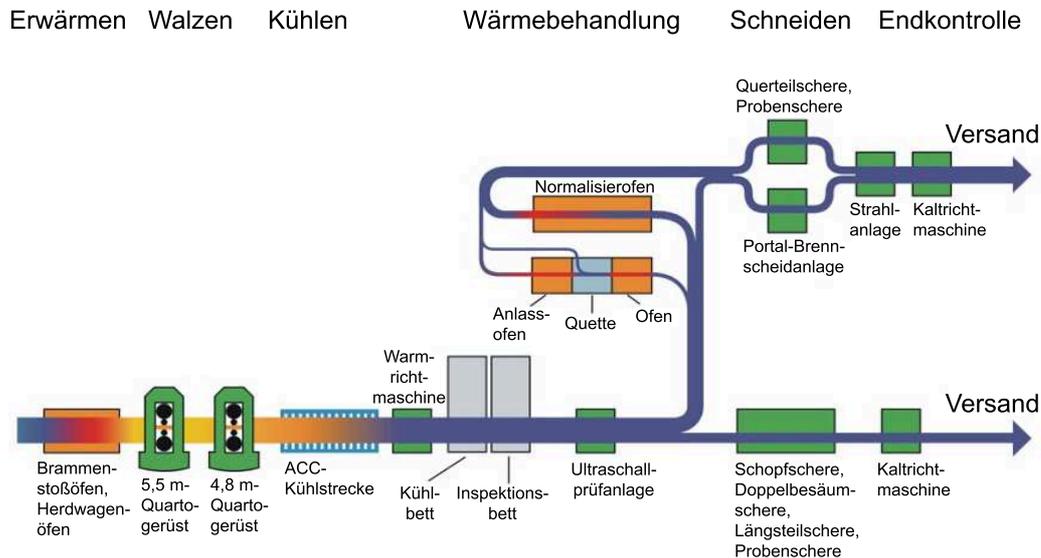


Abbildung 2.9: Walzwerksprozess
(Quelle: AG der Dillinger Hüttenwerke)

Stahl kann grundsätzlich warm- oder kaltgewalzt werden. Der Kraftaufwand und Arbeitsbedarf ist beim Umformen unter hohen Temperaturen jedoch wesentlich geringer. In Grobblechwalzwerken erfolgt die Brammenerwärmung gewöhnlich in sogenannten Stoßöfen, die Blockerwärmung in sogenannten Herdwagenöfen. In einem Stoßofen liegen mehrere Brammen berührungsdicht, meist in mehreren Reihen, auf wassergekühlten Schienen über dem Ofenherd. Der Ofeneintrag erfolgt durch Stoßen mit Hilfe von hydraulischen Drückvorrichtungen. Bei jedem Stoßvorgang verschiebt sich der gesamte Ofeninhalt einen Schritt weiter in Richtung Ofenausgang. Die Ofenbeschickung ist somit stetig. Demgegenüber erfolgt die Beschickung bei Herdwagenöfen satzweise, d. h. eine Ofenbelegung fährt auf einem sogenannten Herdwagen komplett in den Ofen ein und nach abgeschlossener Erwärmung wieder komplett aus ihm aus. Die Dillinger Hüttenwerke verfügen für die Brammen- und Blockerwärmung über jeweils drei Stoß- und Herdwagenöfen. Infolge der Erwärmung auf bis zu 1300 °C hat sich auf dem Walzgut eine Oxidschicht (Zunder) gebildet, die vor

der Walzung entfernt werden muss. Nach abgeschlossener Hochdruckentzunderung in einem Wäscher gelangt das Walzgut schließlich über einen Rollgang zum ersten Walzgerüst.

Witt [102] definiert Walzen als „fortschreitendes Druckumformen mit sich drehenden Werkzeugen (Walzen). Dabei können die Walzen angetrieben oder durch das Werkstück geschleppt sein.“ Durch den Druck ändern sich neben der Geometrie des Walzguts auch dessen Gefüge und mechanisch-technologische Eigenschaften. Das Walzen trägt somit in hohem Maße zur Qualität der Stahlerzeugnisse bei. Je nach Druckumformverfahren kann zwischen Längs-, Quer- und Schrägwalzen unterschieden werden (siehe Abbildung 2.10).

Beim Längswalzen stehen das Walzgut und die Walzenachsen senkrecht zueinander. Dabei bewegt sich das Walzgut i. d. R. ohne Drehung in Richtung seiner Längsachse zwischen den Walzen hindurch. Mit diesem Walzverfahren werden hauptsächlich Flacherzeugnisse (z. B. Bleche oder Bänder) und Profilerzeugnisse (z. B. Stäbe oder Drähte) hergestellt. Beim Querwalzen stehen das Walzgut und die Walzenachsen hingegen parallel zueinander. Während des Walzvorgangs rotiert das Walzgut zwischen den Walzen um seine eigene Längsachse. So lassen sich beispielsweise abgestufte, rotationssymmetrische Stahlerzeugnisse (wie Achsen und Wellen) herstellen. Beim Schrägwalzen bewegt sich das Walzgut unter Drehung um seine Längsachse durch die gegenseitig schräg zueinander geneigten Walzen. Diese Methode wird beispielsweise bei der Produktion nahtloser Rohre angewandt.

Walzgerüste lassen sich je nach Anordnung der Walzen im Gerüst in verschiede-

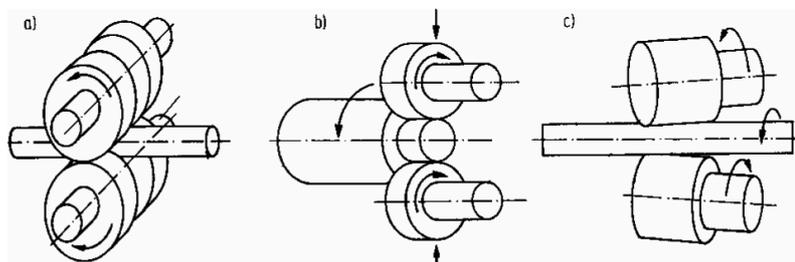


Abbildung 2.10: Walzverfahren

a) Längswalzen, b) Querwalzen, c) Schrägwalzen
(Witt [102])

ne Baugruppen unterteilen (siehe Abbildung 2.11), wobei jede Grundbauform wiederum zusätzliche unterschiedliche Bauarten aufweist. Im Laufe der letzten Jahrzehnte wurden bei der Produktion von Flacherzeugnissen die Zwei- und Dreiwalzengerüste (Abbildung 2.11 b)-e)) mehr und mehr von Vierwalzengerüsten (Abbildung 2.11 f)) abgelöst. Heute werden Zwei- und Dreiwalzengerüste hauptsächlich zur Produktion von Profilerzeugnissen verwendet. Vierwalzengerüste bestehen jeweils aus zwei Stütz- und zwei Arbeitswalzen, wobei die Arbeitswalzen die eigentliche Arbeit verrichten und von den Stützwalzen größeren Durchmessers lediglich abgestützt werden. Die Abstützung verhindert das Durchbiegen der Arbeitswalzen und ermöglicht somit eine gleichmäßige Dickenabnahme. Vielwalzengerüste (Abbildung 2.11 g)-i)) arbeiten nach dem gleichen Prinzip, werden aber ausschließlich zum Kaltwalzen von Stählen verwendet.

Das Walzwerk der Dillinger Hüttenwerke verfügt über zwei Vierwalzengerüste (Quartogerüste), ein 5,5 m breites Gerüst (Vorgerüst) und ein 4,8 m breites

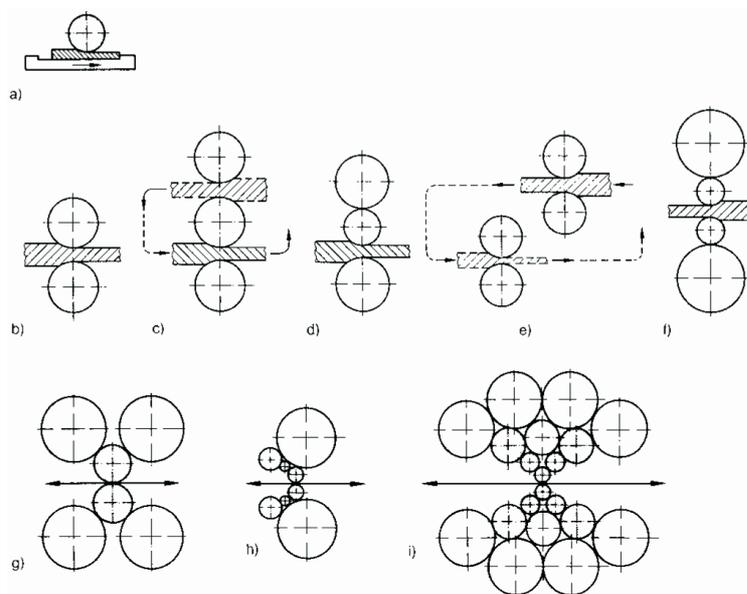


Abbildung 2.11: Walzenanordnung in einem Walzgerüst (Witt [102])

- a) Walzen gegen eine ebene oder profilierte Platte,
- b) Zweiwalzen/Duo-Gerüst, c) Dreiwalzen/Trio-Gerüst,
- d) Lauth'sches Trio-Gerüst, e) Doppel-Zweiwalzengerüst,
- f) Vierwalzen/Quarto-Gerüst, g) Sechswalzengerüst,
- h) Mehrzweck-Kalt-Walzwerk-Gerüst, i) 20-Walzengerüst

Gerüst (Fertiggerüst). Der Walzprozess gliedert sich je nach Anforderung in mehrere Phasen, wobei eine Phase aus einem bzw. mehreren Walzstichen (Umformvorgängen) und aus einer Halte- bzw. Wartezeit besteht. Am Vorgerüst werden die ersten Phasen durchgeführt, am Fertiggerüst, wenn erforderlich, die letzten. Grundsätzlich können die angewandten Walzverfahren in drei Gruppen eingeteilt werden. Zur Gruppe der *Normalwalzung* zählen alle reinen Formgebungsverfahren ohne spezifische Temperaturvorgaben. Mit dem sogenannten *Normalisierersatz-Walzen* wird hingegen das Ziel verfolgt, durch eine geregelte Temperaturführung die spätere Wärmebehandlung des Normalisierens zu ersetzen. Die dritte Gruppe bilden die *thermomechanischen Walzverfahren*, bei denen Temperatur und Umformung in ihrem zeitlichen Ablauf sehr gezielt gesteuert werden, um bestimmte mechanisch-technologische Eigenschaften des Stahls zu erzielen. Bei diesen Verfahren gilt es, die Vorgaben bezüglich Walz- und Wartezeiten genauestens einzuhalten.

Infolge des Walzprozesses entstehen aus den Brammen und Blöcken in Dillingen bis zu 5,4 m breite und bis zu 50 m lange *Walztafeln*. Auf einer Walztafel sind i. d. R. mehrere Bleche verplant. Sofern Probenentnahmen vorgesehen sind, muss zusätzlicher Platz für die meist 400×400 mm großen Probencoupons eingeplant werden. Abbildung 2.12 zeigt eine beispielhafte Darstellung einer Walztafel, bestehend aus vier Blechen und zwölf Proben.

Umfang und Reihenfolge der im Anschluss an die Walzung folgenden Zurichtungs- und Nachbehandlungsverfahren (im sogenannten *Adjustagebetrieb*) hängen von vielen Faktoren ab. Insbesondere die Walztafelabmessungen und Stahlqualitäten sind hierfür maßgeblich entscheidend. Zu den wichtigsten Zurichtungs- und Nachbehandlungsverfahren zählen Maßnahmen

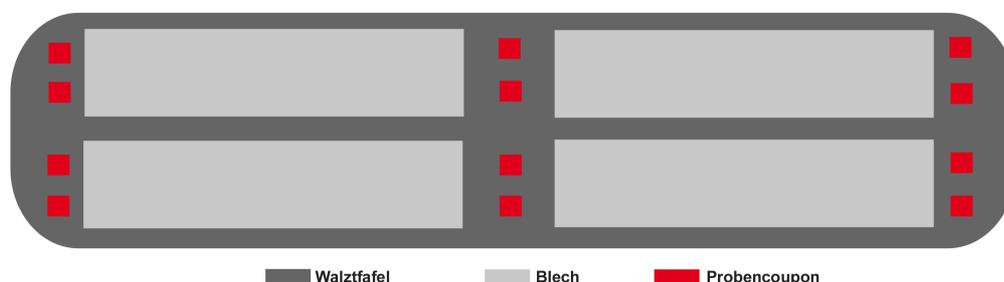


Abbildung 2.12: Walztafel
(Quelle: AG der Dillinger Hüttenwerke)

- zur Abkühlung,
- zum Zuschneiden,
- zum Richten,
- zur Qualitätsprüfung,
- zur Wärmebehandlung und
- zur Oberflächenbearbeitung.

Unterschiedliche Stahlqualitäten können unterschiedliche Anforderungen an den Abkühlungsprozess stellen. Im Walzwerk der Dillinger Hüttenwerke steht hierfür z. B. eine *Accelerated Controlled Cooling* (ACC)-Anlage zur Verfügung. Die Anlage befindet sich auf dem gleichen Rollgang wie die Walzgerüste, wenige Meter hinter dem Fertiggerüst. In ihr können die Walztafeln direkt im Anschluss an die Walzung mit exakt regelbaren Wassermengen von oben und unten kontrolliert gekühlt werden, wodurch sich bestimmte Gefügeeinstellungen und Materialeigenschaften realisieren lassen. Die unterschiedlichen Abkühlprozesse sind in sogenannten *Kühlplänen* zusammengefasst. Eine Walztafel kann die Anlage entweder pendelnd oder kontinuierlich durchfahren. Neben der Art der Fahrweise stellen vor allem die Fahrgeschwindigkeit und die Intensität der beaufschlagten Wassermenge wichtige Stellparameter eines Kühlplans dar. Insbesondere bei thermomechanischer Walzung müssen diese Parameter sehr genau gesteuert werden, da hier der Abkühlprozess von sehr hoher Bedeutung ist und nur geringe Toleranzen erlaubt.

Neben dem kontrollierten Abkühlen in der ACC-Anlage gibt es noch Möglichkeiten zur gleichmäßigen Abkühlung der Walztafeln. Das Ziel dabei besteht darin, eine Wasserstoffeffusion herbeizuführen, die bei zu schneller Abkühlung an der Luft nicht möglich wäre. So lässt sich die Gefahr der Verursachung von Kaltrissen oder gar Sprödbrüchen mindern. Für die gleichmäßige Abkühlung der Walztafeln stehen der Dillinger Hütte sechs Bunker zur Verfügung. Die Mindestliegezeit einer Walztafel im Bunker schwankt zwischen 12 und 48 Stunden. Die Belegung der Bunker erfolgt kontinuierlich, wobei sich die Belegungsreihenfolge stark nach der Walzreihenfolge richtet. Die Ausförderung erfolgt hingegen satzweise, d. h. der komplette Bunker wird entleert, sobald alle sich im Bunker befindlichen Walztafeln ihre Mindestliegezeit erreicht haben. Da die Bunkerkapazitäten begrenzt sind, wird die gleichmäßige Walztafelabkühlung häufig auch einfach durch kompaktes Übereinanderstapeln der Walz-

tafeln (zu sogenannten *Warmstapeln*) herbeigeführt. Auch mit dieser Methode kann der Wasserstoff effundieren. Allerdings sind die Mindestliegezeiten der Walztafeln in einem Warmstapel höher als in einem Bunker.

Zu den Hauptaufgaben des Adjustagebetriebs zählt das Zuschneiden. Walztafeln bzw. Bleche müssen beispielsweise abgelängt, längsgeteilt, quergeteilt oder besäumt (Kantenschneiden) werden. Auch das Herausschneiden der Proben-coupons erfordert Zeit und Aufwand. Die zahlreichen Schneidvorgänge können während des kompletten Prozesses vom Anschluss an die Walzung bis hin zur Vorbereitung für die Verladung mehrmals erforderlich sein. Im Walzwerk der Dillinger Hüttenwerke stehen hierfür neben verschiedenen Scheren- und Brennschneidanlagen noch eine Wasserstrahlschneidanlage sowie eine Plasmaschneidanlage zur Verfügung.

Eine weitere wichtige Zurichtungsmaßnahme ist das Richten. Dieser Vorgang ist erforderlich, um eine möglichst optimale Ebenheit der Endprodukte zu gewährleisten. Das Richten der Walztafeln bzw. Bleche wird im Walzwerk der Dillinger Hüttenwerke auf einer Warm- bzw. auf Kaltrichtmaschinen realisiert. Bei größeren Unebenheiten müssen Richtpressen aushelfen.

Die Qualitätsprüfung bildet ein weiteres Aufgabengebiet des Adjustagebetriebs. Neben den Probenentnahmen, die anschließend von der metallurgischen Abteilung untersucht werden, zählen hierzu vor allem Ebenheits- und Abmessungskontrollen sowie optische Qualitätskontrollen (Prüfung auf Risse). Die Kontrollen werden hauptsächlich auf Inspektionsbetten, die mit einem Blechwender ausgestattet sind, oder auf verschiedenen Besichtigungsböcken oder Ebenheitsmesstischen durchgeführt. Zudem verfügt das Walzwerk der Dillinger Hüttenwerke über eine Ultraschallprüfanlage zur Überprüfung auf innere Fehler.

Da die Gefügestände der Walztafeln im Anschluss an die Walzung noch nicht immer den endgültigen Sollvorgaben entsprechen, erfolgen häufig qualitative Nachbehandlungen. Wichtigstes Instrument hierfür ist die Wärmebehandlung in speziell konstruierten Öfen. Da sich diese Arbeit ab Kapitel 5 vornehmlich einer konkreten Problemstellung in Zusammenhang mit der Wärmebehandlung widmet, werden die wichtigsten Grundlagen hierzu gesondert in Kapitel 3 erläutert. Um den in Abbildung 2.9 abgebildeten Walzwerksprozess an dieser

Stelle dennoch ausreichend zu beschreiben, sei bereits darauf hingewiesen, dass im Walzwerk der Dillinger Hüttenwerke zur Wärmebehandlung hauptsächlich Normalisier-, Anlass- und Quettöfen im Einsatz sind, wobei *Normalisieren*, *Anlassen* und *Quetten* jeweils bestimmte Wärmebehandlungsverfahren darstellen.

Im Laufe des gesamten Zurichtungs- und Nachbehandlungsprozesses werden die Oberflächen der Bleche teilweise stark beansprucht. Auch das Lagern in Stapeln sowie das Transportieren über Rollgänge tragen hierzu bei. In einigen Fällen ist es daher erforderlich, eine nachträgliche Oberflächenbearbeitung der Bleche vorzunehmen. Diese lässt sich z. B. mit Hilfe einer Strahlanlage oder durch Schleifen durchführen.

Nach Abschluss der zahlreichen Zurichtungs- und Nachbehandlungsverfahren gelangen die fertigen Bleche schließlich in den Verladebetrieb. Hier werden sie auf Güterwaggons oder LKW verladen und dann je nach Zielbestimmung direkt zum Kunden, zum nahegelegenen Hafen oder zu einem der auf dem Hüttengelände gelegenen Weiterverarbeitungsbetriebe transportiert.

2.6 Herausforderungen für die Logistik

Anhand der Erläuterungen aus den Unterkapiteln 2.2 - 2.5 lässt sich bereits feststellen, dass die Stahlindustrie bzw. ein integriertes Hüttenwerk durch einen sehr ausgeprägten Materialfluss und hochkomplexe Produktionsprozesse gekennzeichnet ist. Im Folgenden sollen die speziellen Herausforderungen dieser Branche bezüglich der Logistik hervorgehoben werden, wobei der Fokus auf innerbetrieblichen Problemstellungen liegt.

Allgemein lassen sich produzierende Unternehmen je nach Prinzip der Leistungserstellung einem von zwei grundsätzlichen Industriezweigen zuordnen. Vahrenkamp [92] spricht in diesem Zusammenhang von Fertigungs- und von Prozessindustrie. Während die Fertigungsindustrie durch mechanische Arbeiten bestimmt ist, überwiegen in der Prozessindustrie chemische und physikalische Reaktionen. Anders ausgedrückt, die Fertigungsindustrie dient der Formgebung eingebrachter Stoffe und die Prozessindustrie der Veränderung derer

stofflichen Eigenschaften.

Eine Besonderheit der Stahlindustrie besteht darin, beiden Industriezweigen zugeordnet werden zu können. So ist der Produktionsweg von der Kokerei bis einschließlich zum Gießbetrieb ein prozessindustrieller Produktionsabschnitt. Einige der anschließenden Produktionsschritte sind jedoch teilweise beiden Produktionssystemen (wie das Walzen), der Fertigungsindustrie (wie das Richten und Schneiden) oder wiederum der Prozessindustrie (wie das Wärmebehandeln) zuzuordnen. Die Stahlindustrie sieht sich somit mit für beide Industriezweige typischen logistischen Problemstellungen konfrontiert. Abbildung 2.13 fasst produktionslogistische Herausforderungen von integrierten Hüttenwerken, wie dem der Dillinger Hüttenwerke, zusammen.

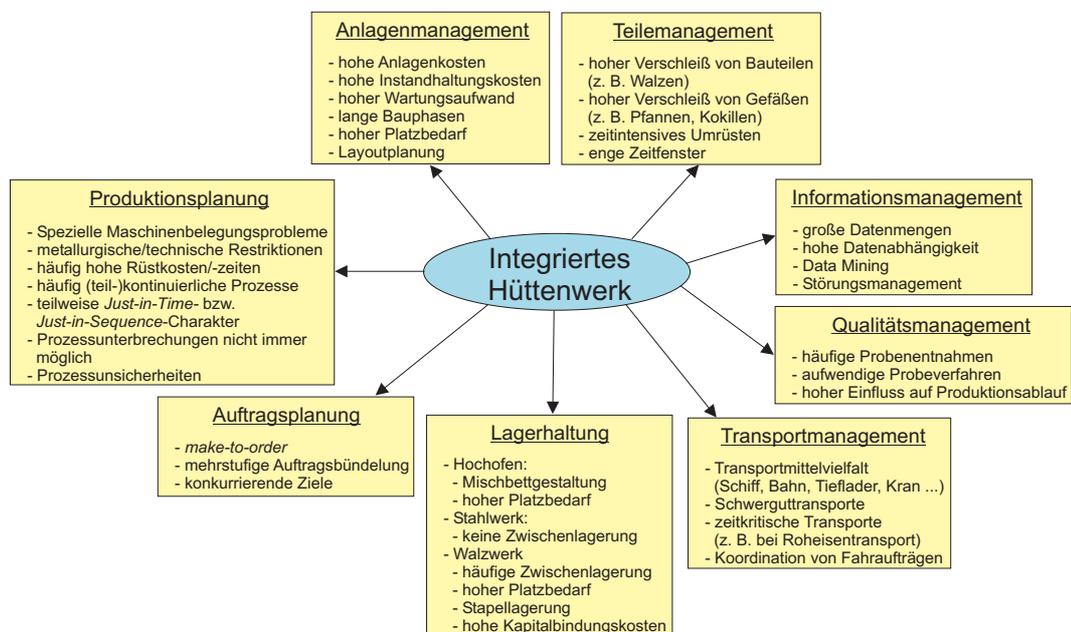


Abbildung 2.13: Produktionslogistische Herausforderungen in integrierten Hüttenwerken

Ein Merkmal der Stahlindustrie allgemein sind große und teure Anlagen. Ein aktuelles Beispiel der Dillinger Hüttenwerke ist die Investition von 300 Millionen Euro in eine neue Stranggießanlage [68]. Neben den hohen Anschaffungskosten sind auch die Betriebskosten (z.B. Energiekosten) derartiger Anlagen extrem hoch. Die zudem hohen Instandhaltungskosten, die in Verbindung mit hohem Wartungsaufwand stehen, stellen die Produktionslogistik immer wieder vor schwierige Planungsaufgaben. Diese bestehen beispielsweise

se darin, die Reparatur- und Wartungsarbeiten möglichst zeit- und kostenoptimal in den Produktionsprozess einzubinden. Noch schwieriger wird es bei Neubauprojekten, deren Bauphasen über mehrere Jahre andauern können. Damit diese den Produktionsprozess nicht zu stark behindern, müssen frühzeitig geeignete Vorkehrungen getroffen werden. Der hohe Platzbedarf der meisten Anlagen, in Verbindung mit den begrenzten Kapazitäten der Betriebshallen, stellt hohe Anforderungen an eine systematische Layoutplanung. Bei der Wahl des Standorts einer neuen Anlage geht es insbesondere auch darum, die Folgen für die innerbetrieblichen Transportwege abzuschätzen und diese in den Entscheidungsprozess einzubeziehen.

Das Teilemanagement ist zwar prinzipiell auch dem Anlagenmanagement zuzuordnen, soll hier jedoch aufgrund seiner Bedeutung gesondert aufgeführt werden. Verschiedene Bauteile, wie z. B. die Walzen des Walzgerüsts sind hohem Verschleiß ausgesetzt und müssen regelmäßig geschliffen oder ausgetauscht werden. Das Schleifen der Arbeitswalzen kann beispielsweise bis zu vier Stunden, das der Stützwalzen bis zu 24 Stunden andauern. Diese Rüstvorgänge müssen bei der Planung des Walzprogramms bzw. der Belegung der Erwärmmöfen berücksichtigt werden. Aufgabe des Teilemanagements ist neben der Terminierung der Walzenwechsel auch das rechtzeitige Bestellen und Bereitstellen der Walzen. Ähnliche Herausforderungen sind im Stahlwerk gegeben. Hier stehen Gefäße, wie Pfannen, Blockgusskokillen oder die Verteilergefäße der Stranggießanlagen unter hohem Verschleiß. Auch diese müssen regelmäßig ausgetauscht, vorbereitet und rechtzeitig bereitgestellt werden, wobei die Zeitfenster hier prozessbedingt sehr eng sind.

Ein oft vergessener bzw. vernachlässigter Bestandteil der Logistik ist das Informationsmanagement, dem insbesondere in der Stahlindustrie große Bedeutung zukommt. Hier herrscht nicht nur eine enorm große Datenvielfalt, sondern auch eine hohe Datenabhängigkeit. Aufgrund des hochkomplexen sowie anlagenintensiven Produktionsablaufs sind Störungen oder Fehlberechnungen keine Seltenheit. Um aus Fehlberechnungen entsprechend lernen bzw. Störungen vorbeugen zu können, gilt es insbesondere, die physikalischen und chemischen Prozesse genauestens zu dokumentieren, die Daten systematisch aufzubereiten sowie Rückschlüsse aus den Ergebnissen zu ziehen. Ein häufig in diesem Zusammenhang verwendeter Begriff ist das *Data Mining*, das Arnold et al.

[1] als „automatisches Detektieren von Zusammenhängen in Datenbeständen durch Anwendung von Algorithmen“ bezeichnen. Zudem muss ein funktionierendes Informationsmanagement sicherstellen, dass die gewonnenen Informationen zielgerichtet und rechtzeitig ver- bzw. übermittelt werden.

Das Qualitätsmanagement hat insbesondere in der europäischen Stahlindustrie eine Vorrangstellung, da sich europäische Unternehmen nur über das Angebot von hochqualitativen Stählen vom asiatischen Markt abheben können. Bei den Dillinger Hüttenwerken gilt das Motto: „Stillstand gleich Rückschritt“, d. h. es wird kontinuierlich an Qualitätsverbesserungen gearbeitet und auch geforscht. Die hohen Qualitätsanforderungen resultieren in häufigen Probenentnahmen sowie aufwändigen Probeverfahren entlang der gesamten Produktionskette. Dies hat einen dementsprechend hohen Einfluss auf den Produktionsablauf, was sich in längeren bzw. zusätzlichen Bearbeitungszeiten und damit höheren Produktionskosten und längeren Lieferzeiten niederschlägt.

Integrierte Hüttenwerke haben einen großen Flächenbedarf. So erstreckt sich das Werksgelände der Dillinger Hüttenwerke über ca. 500 Hektar. Für den inner- und zwischenbetrieblichen Transport der Rohstoffe und Zwischenprodukte ergibt sich daraus die Notwendigkeit, relativ große Distanzen zu überbrücken. Hinzu kommt, dass mit feinkörnigen, stückigen, flüssigen und festen sowie teilweise sehr heißen und schweren Transportmaterialien hohe Anforderungen an die Transportmittel gestellt werden. Aus dieser Materialvielfalt resultiert demnach auch eine hohe Transportmittelvielfalt. Während die Anlieferung der Rohstoffe und die Auslieferung der Fertigprodukte hauptsächlich auf Schiffen vollzogen werden, erfolgt der Transport auf dem Werksgelände beispielsweise über spezielle Waggons, Tieflader oder Krane. Viele dieser Transporte sind mit einem engen Zeitfenster behaftet. So ist insbesondere die Versorgung des Stahlwerks mit Roheisen vom Hochofen sehr zeitkritisch, da das Roheisen nicht auskühlen darf. Aus den genannten Gründen bedarf es eines Transportmanagements, das einen ausreichenden Bestand an Transportmitteln sicherstellt und verwaltet, sowie die einzelnen Fahraufträge optimal koordiniert.

Die Herausforderungen an die Lagerhaltung unterscheiden sich zwischen den einzelnen Betrieben innerhalb eines integrierten Hüttenwerks sehr stark von-

einander. Wie bereits in Unterkapitel 2.3 erläutert, werden die meisten Einsatzstoffe des Hochofens auf Mischbetten lagenweise übereinander geschichtet. Dabei ist die Gestaltung der Mischbetten hinsichtlich des Aufbaus der einzelnen Lagen von großer Bedeutung, um den angestrebten Mischeffekt beim Abtragen zu erzielen. Kennzeichnend für das Stahlwerk ist, dass während des Prozesses von der Roheisenanlieferung bis zum Vergießen keine Zwischenlagerung erfolgt. Nach dem Vergießen ändern sich Lagerart und -notwendigkeit jedoch drastisch. Aufgrund begrenzter Lagerkapazitäten müssen die Brammen sowie die späteren Walztafeln und Bleche von nun an in Stapeln übereinander gelagert werden. Zudem müssen diese Stapellager nahezu nach jedem der folgenden Arbeitsschritte gebildet werden, da die Bearbeitungsfolgen und -zeiten der Anlagen sowohl von der Produktionsreihenfolge und Taktzeit des Gießbetriebs als auch untereinander abweichen. Hieraus resultieren nicht nur hohe Lager- und Kapitalbindungskosten, sondern auch lange Zugriffszeiten. So ist es nahezu unmöglich, die Stapel immer so zu bilden, dass sie von oben herab abgearbeitet werden können. Infolgedessen müssen beispielsweise in einem Blechlager häufig einige im Stapel oben liegende Bleche zunächst umgestapelt werden, bevor das als nächstes benötigte Blech dem Lager entnommen werden kann. Sowohl dieses Umstapeln und Entnehmen der Bleche als auch die Stapelbildung erfolgt dabei mittels mit Magneten ausgestatteten Kranen. Somit kann immer nur ein Blech gleichzeitig aufgenommen und transportiert werden. Bei Stapeln mit über 100 Blechen ist ersichtlich, wie zeitintensiv derartige Kranbewegungen sind. Um diese Problematik zu bewältigen, bedarf es einer auf die Produktion optimal abgestimmten Lagerhaltung.

Neben der bzw. durch die Strategie auf die Produktion hochqualitativer Stähle zu setzen, zeichnen sich europäische Stahlunternehmen durch eine hohe Variantenvielfalt aus. Bei den Dillinger Hüttenwerken resultiert dies in einer auftragsbasierten Produktion (*make-to-order*). Da es sich bei dem Produktionsprozess ab dem Konverter um eine divergierende Produktion handelt, muss die Auftragsplanung den Produktionsweg in umgekehrter Richtung betrachten. So müssen Bleche auf Walztafeln verplant, diese wiederum zu Brammen umgerechnet und diese wiederum zu Gießsequenzen und entsprechenden Schmelzen verplant werden. Die Abstimmung der Aufträge auf die Roheisenproduktion ist nicht zwingend notwendig, da die kundenspezifischen Stahleigenschaften erst im Stahlwerk eingestellt werden und eine kurzfristige Leistungsänderung

des Hochofens zudem nur sehr begrenzt möglich ist. Die mehrstufige Auftragsbündelung ist eine sehr komplexe Aufgabe und mit zahlreichen Restriktionen und teilweise konkurrierenden Zielen verbunden. Beim Konverter z. B. sind die Chargengrößen durch Mindestfüllmenge und Fassungsvermögen vorgegeben. Da die Kunden i. d. R. nicht genau dieser Menge entsprechend bestellen, muss häufig mehr produziert werden. Die zu viel gegossenen Brammen können dann entweder eingelagert oder wieder eingeschmolzen werden. Bezüglich der Formate ist die Leistung der Stranggießanlagen bei dickeren Formaten vergleichsweise schlecht, wohingegen die Leistung der Walzung bei dickeren Formaten hoch ist. Im Großen und Ganzen besteht die Aufgabe der Auftragsplanung darin, die einzelnen Kundenaufträge derart zu verplanen bzw. zu bündeln, dass sie den gewünschten Anforderungen entsprechen und möglichst schnell und kostengünstig zu realisieren sind.

Als letzter Punkt der produktionslogistischen Herausforderungen folgt nun mit der Produktionsplanung das eigentliche Kerngebiet der Produktionslogistik. Besonderes Augenmerk sei hier auf die zahlreichen und häufig sehr speziellen Maschinenbelegungsprobleme gelegt. Das Thema wird in Kapitel 4 ausführlich behandelt. Der Unterschied zu den klassischen Problemstellungen folgt meist aus den speziellen Gegebenheiten des Stahlherstellungs- und verarbeitungsprozesses. So sind die Maschinenbelegungsprobleme meist mit zahlreichen metallurgischen und technischen Restriktionen und hohen Rüstkosten bzw. langen Rüstzeiten behaftet (Beispiel: Stranggießanlage). Zudem liegen häufig (teil-)kontinuierliche Prozesse vor, die z. B. dazu führen, dass einmal begonnene Bearbeitungen nicht unterbrochen werden können und verbundene Anlagen genau aufeinander abgestimmt werden müssen. Demnach besitzen Abläufe, bei denen die Temperaturführung der Materialien genauestens einzuhalten ist, häufig sogar *Just-in-Time*- bzw. *Just-in-Sequence*-Charakter (Beispiel: thermomechanische Walzung). Insbesondere diese Problemstellungen verschärfen sich durch die Existenz von Prozessunsicherheiten. So sind viele chemische und physikalische Prozesse noch heute mit Unsicherheiten und damit starken Schwankungen behaftet. Davon betroffen sind prinzipiell die Prozesse im Hochofen allgemein, aber auch beispielsweise Legierungseinstellungen im Stahlwerk sowie Erwärm-, Bearbeitungs- und Abkühlzeiten im Walzwerk.

An dieser Stelle soll erwähnt sein, dass die Darstellung der produktionslogisti-

schen Herausforderungen in integrierten Hüttenwerken (Abbildung 2.13) und die anschließenden Erläuterungen keinen Anspruch auf Vollständigkeit erheben und lediglich einer beispielhaften Auswahl einiger Aspekte genügen. Zudem überschneiden sich die Grenzen der einzelnen Aufgabengebiete teilweise. Diese Arbeit befasst sich im weiteren Verlauf hauptsächlich mit den Teilgebieten der Produktionsplanung und der Lagerhaltung, speziell im Bereich der Wärmebehandlung.

In der Literatur finden sich bis heute nur wenige produktionslogistische Arbeiten im Zusammenhang mit der Stahlindustrie. Der Großteil stammt aus dem asiatischen Raum und beschäftigt sich z. B. mit dem Stahlwerksprozess, der Stapelproblematik oder dem Walzprozess (siehe Tabelle 2.1).

Tabelle 2.1: Ausgewählte produktionslogistische Arbeiten aus der Stahlindustrie

| Stahlwerksprozess | Stapelproblematik | Walzprozess |
|-----------------------------|--------------------------|--------------------|
| Pacciarelli und Pranzo [69] | Hansen [44] | Cowling [23] |
| Bellabdaoui und Teghem [6] | König [57] | Busch [14] |
| Tang und Wang [87] | Cheng und Tang [21] | Wang und Tang [97] |
| Atighehchian et al. [3] | Tang und Ren [86] | Pan und Yang [70] |
| Zhu [104] | | Zhao et al. [103] |
| | | Tang et al. [88] |

Maschinenbelegungsprobleme im Zusammenhang mit Wärmebehandlungsöfen und der damit verbundenen Lagerhaltungsproblematik wurden nach derzeitigem Kenntnisstand bislang noch nicht in der Literatur betrachtet. Genau diese Lücke wird mit der vorliegenden Arbeit gefüllt. Bevor jedoch im Detail auf die spezielle Problemstellung eingegangen wird, vermitteln Kapitel 3 und 4 zunächst die erforderlichen Grundlagen.

Kapitel 3

Grundlagen der Wärmebehandlung von Stahl

Dieses Kapitel dient mit Unterkapitel 3.1 zunächst dazu, Zweck und Ziele des Wärmebehandelns zu beschreiben. Anschließend wird der grundlegende Ablauf einer Wärmebehandlung in Unterkapitel 3.2 erläutert. Die Vorstellung und Einteilung der wichtigsten Wärmebehandlungsverfahren schließt sich in Unterkapitel 3.3 an. Mit Unterkapitel 3.4 wird die Beschreibung der Grundlagen um eine Erläuterung der gängigen Wärmebehandlungsbegriffe, -verfahren und -einrichtungen, welche im Walzwerk der Dillinger Hüttenwerke ver- bzw. angewendet werden, ergänzt.

3.1 Zweck und Ziele des Wärmebehandelns

Eisenwerkstoffe sind im festen Zustand i. d. R. kristallin aufgebaut, d. h. die räumliche Anordnung der Atome ist festgelegt. Eisen hat jedoch die Eigenschaft, dass die Atome sich zu unterschiedlichen Gitterstrukturen aufbauen können (*Polymorphie*). Je nach Kohlenstoffgehalt und Temperatur resultieren demnach unterschiedliche Gefüge. Da zudem die verschiedenen Werkstoffeigenschaften hauptsächlich durch das Gefüge bestimmt werden, lassen sich durch Wärmebehandeln bestimmte Eigenschaften gezielt einstellen.

Nach DIN EN 10052 [30] bedeutet Wärmebehandeln „ein Werkstück ganz oder teilweise Zeit-Temperatur-Folgen zu unterwerfen, um eine Änderung seiner Eigenschaften und/oder seines Gefüges herbeizuführen. Gegebenenfalls kann während der Behandlung die chemische Zusammensetzung des Werkstoffs geändert werden.“

Die Notwendigkeit des Wärmebehandelns ergibt sich aus der Tatsache, dass der Werkstoffzustand nach dem Umformprozess oft noch nicht den Anforderungen des späteren Verwendungszwecks gerecht wird. So können je nach Verwendungszweck beispielsweise unterschiedliche Anforderungen bezüglich

- Härte,
- Festigkeit,
- Zähigkeit,
- Verschleißwiderstand,
- Schweißbarkeit oder
- Korrosionsbeständigkeit

bestehen.

Grobbleche für den Brückenbau müssen z.B. sowohl über eine sehr hohe Zähigkeit als auch über eine sehr hohe Festigkeit verfügen. Mit sehr zähen Stahlkonstruktionen lassen sich beispielsweise Schwingungen der Brücke bei Sturm ausgleichen. Eine hohe Festigkeit ist notwendig, um auch der Belastung der überquerenden Fahrzeuge standzuhalten. Um die Bleche miteinander zu verbinden, ist zudem eine hohe Schweißbarkeit erforderlich. Liegt die Brücke am (bzw. die Träger im) Wasser, muss der Stahl zudem korrosionsbeständig sein. Neben den Anforderungen an die Sicherheit derartiger Konstruktionen spielen auch Kostenaspekte eine wichtige Rolle. So erlaubt eine hohe Festigkeit, die Blechdicke der Stahlkonstruktionen, verglichen mit konventionellen Stählen, erheblich zu verringern, wodurch eventuell Materialkosten eingespart werden können. Aufgrund der besseren Verarbeitungseigenschaften sehr zäher Stähle, lassen sich zudem die Kosten der Verarbeitung senken.

Ein anderes Anwendungsgebiet, bei dem extrem hohe Anforderungen an die Härte und den Verschleißwiderstand der eingesetzten Stähle gestellt werden,

ist die Herstellung von großen Baumaschinen und Baugeräten. Beispiele hierfür sind Baggerschaufeln und Muldenkipper für den Einsatz im Bergbau, Steinbrüchen oder Kiesgruben. Den Kunden dieser Branche ist insbesondere eine möglichst lange Haltbarkeit der Stahlbauteile wichtig. Von ebenso großer Bedeutung ist aber auch hier das Einsparpotential bezüglich Material- und Verarbeitungskosten.

An dieser Stelle sei darauf hingewiesen, dass ein Werkstück nicht gleichzeitig allen Anforderungen in hohem Maße gerecht werden kann. Der Grund hierfür ist, dass manche Ziele untereinander konträr sind und somit Zielkonflikte bestehen. Insbesondere sehr gute Härte- und Zähigkeitseigenschaften lassen sich kaum miteinander vereinbaren. Allgemein lässt sich sagen, dass man immer einen Kompromiss zwischen hoher Verschleißfestigkeit und bestmöglicher Verarbeitbarkeit eingehen muss.

Aus den aufgeführten Beispielen und Erläuterungen lässt sich das Hauptziel des Wärmebehandelns ableiten. Demnach gilt es, für den jeweiligen Verwendungszweck das Verhältnis zwischen Beanspruchbarkeit, Werkstückgeometrie und Abmessungen so zu optimieren, dass sowohl eine hohe Lebensdauer mit ausreichender Sicherheit gewährleistet ist, als auch mit der späteren Konstruktion verbundene Material- und Verarbeitungskosten Berücksichtigung finden.

3.2 Ablauf des Wärmebehandelns

Zur Auslösung der gewünschten Vorgänge im Werkstoff ist die Zufuhr bzw. der Entzug einer bestimmten Wärmemenge erforderlich. Bei den meisten Wärmebehandlungsverfahren kommt zudem der Zeit, in der diese Wärmemenge dem Werkstoff zugeführt bzw. entzogen wird, eine wichtige Rolle zu. Häufig muss der Werkstoff auch eine gewisse Zeit bei einer bestimmten Temperatur gehalten werden. So läuft bei allen Verfahren die Wärmebehandlung prinzipiell in drei Schritten ab:

- Erwärmen auf eine bestimmte Behandlungstemperatur,
- Halten der Behandlungstemperatur für eine bestimmte Zeit,

- Abkühlen von Behandlungs- auf Raumtemperatur in bestimmter Geschwindigkeit.

Abbildung 3.1 veranschaulicht einen beispielhaften Zeit-Temperatur-Verlauf. Im Gegensatz zur Halte- und Abkühlphase, die je nach Verfahren kurz oder lang bzw. schnell oder langsam durchgeführt werden können, sollte die Erwärmphase grundsätzlich eher langsam ablaufen. Dies hat den Grund, dass bei zu schneller Erwärmung die Oberflächentemperatur wesentlich schneller steigen würde als die Kerntemperatur. Die Folgen wären unterschiedliche Spannungen im Werkstück und gegebenenfalls Rissbildung. Daher unterteilt man gewöhnlich die Erwärmphase noch einmal in eine Anwärm- und eine Durchwärmphase. Die Anwärmphase stellt die Zeitspanne dar, welche die Oberfläche vom Beginn des Erwärms bis zum Erreichen der Behandlungstemperatur benötigt. Die Durchwärmphase ist dementsprechend die Zeitspanne bis der Kern die Behandlungstemperatur erreicht. Je nach Verfahren spricht man auch von der Glüh- oder Härtetemperatur.

Sobald auch der Kern des Werkstücks die Behandlungstemperatur erreicht hat, beginnt die zweite Phase, auch Haltephase genannt. Hier wird das Werkstück nicht mehr weiter erwärmt, sondern stattdessen auf der Behandlungstemperatur gehalten. Die Dauer dieses Abschnitts wird demnach auch als Haltedauer bezeichnet. Bei manchen Wärmebehandlungsverfahren kann die Haltedauer

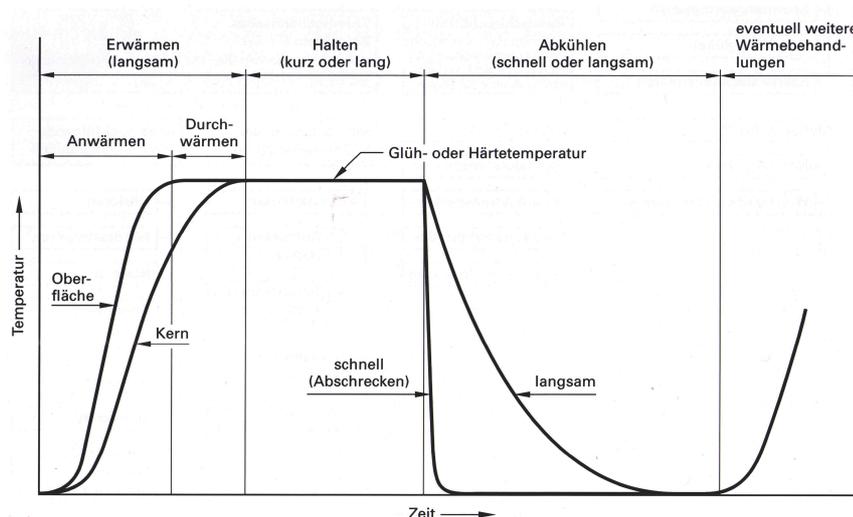


Abbildung 3.1: Schematischer Zeit-Temperaturverlauf bei der Wärmebehandlung von Stahl (Läpple [62])

sehr kurz sein bzw. die Haltephase sogar komplett übersprungen werden, bei anderen kann das Halten mehrere Stunden andauern.

Im Anschluss an die Haltephase, d. h. sobald die Haltedauer abgeschlossen ist, wird das Werkstück abgekühlt. Vor dem Hintergrund, dass die Erwärm- und die Haltephase in einem Ofen vollzogen werden, beginnt die Abkühlphase prinzipiell automatisch mit der Entnahme des Werkstücks aus dem Ofen. Neben der Abkühlung an der Luft gibt es aber noch andere Abkühlmethoden. Manche Verfahren sehen beispielsweise eine sehr langsame Abkühlung vor. In diesem Fall kühlt das Werkstück meist noch im Ofen oder in einer Art nachgelagerter Thermobox ab. Andere Verfahren erfordern eine extrem schnelle Abkühlung (Abschrecken). Hierbei wird das Werkstück direkt nach der Entnahme aus dem Ofen beispielsweise durch das Zuführen bestimmter Wassermengen von oben und unten abgekühlt. Nach erfolgter Abkühlung ist die Wärmebehandlung prinzipiell abgeschlossen. Je nach Anforderungsprofil und gewünschter Qualität des Werkstücks können aber auch mehrere aufeinanderfolgende Wärmebehandlungsverfahren erforderlich sein.

An dieser Stelle sei darauf hingewiesen, dass es von großer Wichtigkeit ist, die je nach Verfahren vorgegebenen Temperaturen und Zeiten in den einzelnen Phasen genauestens einzuhalten. In verschiedenen Bereichen gibt es zwar auch gewisse Toleranzen, doch wenn diese überschritten werden, muss die Wärmebehandlung i. d. R. wiederholt werden. Da Wärmebehandlungen häufig mit sehr langen Zeiten verbunden sind, gilt es, dies möglichst zu vermeiden. In manchen Fällen kann der Fehler der Wärmebehandlung auch irreversibel sein, sodass das entsprechende Werkstück Ausschuss ist. Wie bereits erwähnt, führt ein zu schnelles oder auch ungleichmäßiges Erwärmen beispielsweise zu unterschiedlichen Spannungen oder Rissbildungen. Das Problem der Spannungen kann durch nachgelagerte Behandlungen wieder behoben werden. Werkstücke mit Rissen sind hingegen Ausschuss.

3.3 Wärmebehandlungsverfahren

Heutzutage sind in der Stahlindustrie bzw. der stahlverarbeitenden Industrie zahlreiche Wärmebehandlungsverfahren im Einsatz. Ihr Ablauf entspricht im Wesentlichen dem in Kapitel 3.2 beschriebenen, grundsätzlichen Ablauf. Die einzelnen Verfahren lassen sich hauptsächlich durch unterschiedliche Ausprägungen der einzelnen Phasen voneinander abgrenzen. So kann ein Verfahren prinzipiell anhand der folgenden verfahrenscharakteristischen Wärmebehandlungsparameter beschrieben werden:

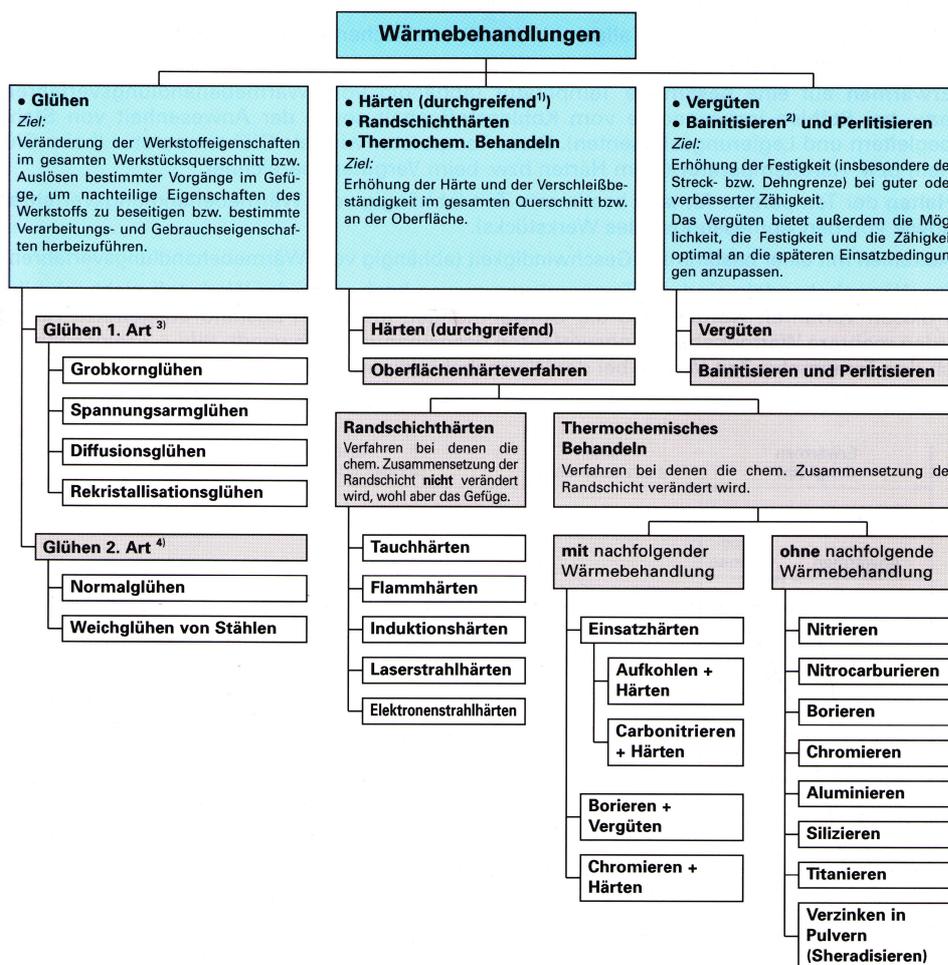
- Aufheizrate,
- Behandlungstemperatur,
- Haltedauer,
- Abkühlrate.

Läpple [62] bietet einen Überblick über verschiedene Verfahren zur Wärmebehandlung von Stahl. Die Verfahren sind in Abbildung 3.2 zusammengefasst und bestimmten Gruppen zugeordnet. Hinsichtlich der verfolgten Ziele einer Wärmebehandlung werden grundsätzlich die folgenden drei Hauptgruppen unterschieden:

- Glühen,
- Härten,
- Vergüten.

Ziel des Glühens ist, den Werkstoff in einen in seiner Gefügebesechaffenheit und inneren Spannung gleichmäßigeren und beständigeren Zustand zu bringen. Im Allgemeinen folgt daraus ein eher weniger fester Zustand. Im Gegensatz dazu wird beim Härten eine möglichst harte bzw. feste Struktur des Gefüges angestrebt. Das Vergüten verfolgt das Ziel, die schwer miteinander zu vereinbarenden Eigenschaften einer sowohl hohen Festigkeit als auch hohen Zähigkeit zu erreichen.

Läpple [62] unterteilt die drei Hauptgruppen noch in weitere Untergruppen. Die Erläuterung der verschiedenen Glüh-, Härte- und Vergütungsverfahren folgt in den Unterkapiteln 3.3.1 - 3.3.3.



¹⁾ Durchgreifendes Härten bedeutet, dass der Härtevorgang im gesamten Werkstücksquerschnitt abläuft.

²⁾ Bainitisieren wurde früher auch als **Zwischenstufenvergüten** bezeichnet.

³⁾ Glühen 1. Art bedeutet, dass es sich um ein **werkstoffunabhängiges Glühverfahren** handelt, d.h. unter den entsprechenden Voraussetzungen im Allgemeinen bei allen Metallen bzw. Metalllegierungen angewandt werden kann.

⁴⁾ Glühen 2. Art bedeutet, dass das Glühverfahren ganz bestimmte Eigenschaften eines Metalls bzw. einer Metalllegierung ausnützt und damit **werkstoffabhängig** ist.

Abbildung 3.2: Überblick und Einteilung verschiedener Wärmebehandlungsverfahren (Läpple [62])

3.3.1 Glühen

Wie bereits erwähnt, können Eisenwerkstoffe in verschiedenen Erscheinungsformen (bezüglich des Gefüges) vorkommen. Dabei entsprechen fehlerfreie Kristallgitter eher einer idealisierten Vorstellung. Tatsächlich enthalten alle Metalle Gitterbaufehler, die in einer Störung des Kristallaufbaus resultieren. Glühverfahren zielen darauf ab, die bestehenden Fehlernordnungen zu reduzieren und somit einen gleichmäßigeren Gitteraufbau herbeizuführen.

Nach DIN EN 10052 [30] ist Glühen „eine Wärmebehandlung, bestehend aus Erwärmen auf eine bestimmte Temperatur (= die Glühtemperatur), Halten und Abkühlen unter solchen Bedingungen, dass der Zustand des Werkstoffs bei Raumtemperatur dem Gleichgewichtszustand näher ist.“ Bezogen auf die in Kapitel 3.1 dargelegten Werkstoffeigenschaften resultiert ein solcher Zustand in einer höheren Zähigkeit sowie einer besseren Schweißbarkeit, jedoch in einer weniger hohen Festigkeit.

Läpple [62] unterteilt die Hauptgruppe der Glühverfahren noch einmal in ein Glühen 1. Art und ein Glühen 2. Art bzw. in werkstoffunabhängige und werkstoffabhängige Verfahren. Zu den werkstoffunabhängigen Verfahren zählen

- das Grobkornglühen,
- das Spannungsarmglühen,
- das Diffusionsglühen sowie
- das Rekristallisationsglühen.

Bei den werkstoffabhängigen Verfahren unterscheidet man zwischen

- dem Normalglühen und
- dem Weichglühen.

Das Grobkornglühen zielt auf eine zerspannungstechnisch günstige Gefügestruktur ab. Obwohl grobkörnige Metalle (im Vergleich zu feinkörnigen) eine geringere Festigkeit und Zähigkeit besitzen, kann Grobkörnigkeit dennoch für bestimmte spanabhebende Bearbeitungen (z.B. Drehen, Hobeln oder Fräsen) erwünscht sein. Die Glühtemperaturen sind beim Grobkornglühen mit $950\text{ }^{\circ}\text{C}$ - $1100\text{ }^{\circ}\text{C}$ sehr hoch, was das Verfahren sehr teuer macht. Die Haltedauer beträgt 1 bis 4 Stunden. Je nach Stahlsorte schließt sich entweder ein langsames oder schnelles Abkühlen an.

Ziel des Spannungsarmglühens ist der Abbau von Eigenspannungen im Werkstück, wobei die übrigen Werkstoffeigenschaften möglichst unverändert bleiben sollen. Eigenspannungen können bei äußeren Belastungen zu Rissbildung oder gar zum Bruch des Werkstücks führen. Beim Spannungsarmglühen wird üblicherweise sehr langsam auf Temperaturen zwischen $450\text{ }^{\circ}\text{C}$ und $650\text{ }^{\circ}\text{C}$ erwärmt. Das anschließende Halten dauert ca. 1 bis 2 Stunden. Das Abkühlen sollte wie

das Erwärmen mit möglichst geringer Geschwindigkeit erfolgen, sodass sich keine neuen Eigenspannungen bilden.

Das Diffusionsglühen verfolgt das Ziel, Konzentrationsunterschiede im Werkstück (z. B. Kristallseigerungen) zu verringern bzw. möglichst zu beseitigen. Dies ist mit einer Veränderung der chemischen Zusammensetzung des Werkstücks verbunden. Hierzu sind Glühtemperaturen im Bereich von 1050 °C und 1300 °C sowie ein bis zu 50 Stunden langes Halten erforderlich. Auch bei diesem Verfahren sollte die Abkühlung möglichst langsam erfolgen.

Durch das Rekristallisationsglühen wird versucht, die mit einer Kaltverformung häufig einhergehende Kaltverfestigung zu beseitigen. Das Ziel besteht darin, die plastische Verformbarkeit, die bei einer Kaltverfestigung stark vermindert ist, wiederherzustellen. Glühtemperatur und Haltedauer sind beim Rekristallisationsglühen stark von der Schwere der vorangegangenen Kaltverfestigung abhängig. So kann das Halten der zwischen 550 °C und 700 °C liegenden Glühtemperaturen sowohl wenige Minuten als auch mehrere Stunden andauern. Das Abkühlen muss auch hier wiederum langsam erfolgen.

Das Normalglühen ist im Vergleich zu den bisher beschriebenen Verfahren werkstoffabhängig. Es zählt zu den wichtigsten Wärmebehandlungsverfahren für Stähle und hat ein gleichmäßiges, möglichst feinkörniges Gefüge zum Ziel. So sollen insbesondere Stähle, die aufgrund vorangegangener Umformungen sehr fest, aber wenig zäh geworden sind, wieder ihr ursprüngliches („normales“) Gefüge einnehmen. Die Glühtemperaturen liegen je nach Stahlsorte zwischen 800 °C und 950 °C. Die Haltedauer ist sowohl von der Glühtemperatur als auch vom Format des Werkstücks abhängig, da das Gefüge über den ganzen Querschnitt des Werkstücks umgewandelt werden soll. So können je nach Werkstück Haltedauern von wenigen Minuten bis zu mehreren Stunden erforderlich werden. Die anschließende Abkühlung sollte auch beim Normalglühen relativ langsam vonstattengehen.

Ziel des Weichglühens ist, den Stahl weicher zu machen, d. h. eine sehr hohe Verformbarkeit bei niedriger Festigkeit und Härte zu erreichen. So soll der Stahl bestmöglich auf nachfolgende Prozesse, wie beispielsweise Kaltumformungen oder zerspanende Weiterverarbeitungen, vorbereitet werden. Die Glühtemperaturen liegen beim Weichglühen je nach Stahlsorte zwischen 650 °C und 750 °C.

Dem Halten von bis zu 100 Stunden schließt sich ein langsames Abkühlen an.

Es lässt sich festhalten, dass sich die meisten Glühverfahren im Wesentlichen nur hinsichtlich der Glühtemperatur und der Haltedauer voneinander unterscheiden. Der Abkühlvorgang erfolgt bei fast allen Verfahren eher langsam.

3.3.2 Härten

Während Glühverfahren das Ziel verfolgen, die bestehenden Fehlernordnungen im Gitteraufbau durch Vergleichmäßigung zu reduzieren, wird beim Härten gezielt ein anderer Gitteraufbau erzeugt, der in einen Zustand höherer Festigkeit resultiert. Dabei nimmt jedoch gleichzeitig auch die Anzahl der Gitterbaufehler zu.

Das Stahl-Institut VDEh [80] beschreibt Härten als „Wärmebehandlung, die infolge fundamentaler Änderungen der Gitterstruktur des Werkstoffs zu harten Gefügebestandteilen führt.“ Die Steigerung von Härte und Festigkeit geht jedoch i. d. R. auf Kosten der Zähigkeit.

Läpple [62] unterteilt die Hauptgruppe der Härteverfahren je nachdem, ob das Werkstück in seinem gesamten Querschnitt oder nur an der Oberfläche gehärtet wird, nochmals in

- durchgreifendes Härten und
- Oberflächenhärten.

Bei letzterem kann wieder zwischen

- Randschichthärten und
- thermochemischem Behandeln

unterschieden werden. Während die chemische Zusammensetzung der Randschicht beim Randschichthärten nicht verändert wird, zielt das thermochemische Behandeln genau darauf ab.

Das durchgreifende Härten (Durchhärten) verfolgt das Ziel einer Härtezunah-

me im gesamten Werkstück, was bedeutet, dass die Erwärmung langsam erfolgen muss. Sobald Oberfläche und Kern die Behandlungstemperatur (Härtetemperatur) erreicht haben, wird diese entweder gar nicht oder nur sehr kurz (maximal eine Stunde) gehalten. Im Anschluss daran wird das Werkstück abgeschreckt, d. h. es wird einer Abkühlung unterzogen, die mit höherer Geschwindigkeit abläuft als an ruhender Luft. Entscheidend für den Erfolg des Härtens sind insbesondere die Wahl der für die jeweilige Stahlsorte optimalen Härtetemperatur sowie der Abschreckgeschwindigkeit und des Abschreckmittels. Gängige Härtetemperaturen beim Durchhärten liegen zwischen 800 °C und 1000 °C. Als Abschreckmittel dient hauptsächlich Wasser. Entsprechend der auf das Werkstück aufgebrauchten Wassermenge lässt sich somit auch die gewünschte Abkühlgeschwindigkeit regeln.

Bei einigen hochwertigen und hochbelasteten Bauteilen ist es durchaus erforderlich, dem Inneren eines Werkstückes eine hohe Zähigkeit, der Oberfläche hingegen eine hohe Härte und Verschleißbeständigkeit zu verleihen. Durch Verfahren des Oberflächenhärtens wird versucht, diese komplizierte Forderung zu erfüllen. Wie der Name schon sagt, beschränken sich alle Oberflächenhärteverfahren auf die Oberfläche bzw. die Randschicht des jeweiligen Werkstücks.

Beim Randschichthärten wird das Gefüge, nicht aber die chemische Zusammensetzung der Randschicht geändert. Die einzelnen Verfahren, die dieser Gruppe angehören, können wiederum nach der angewandten Wärmequelle unterschieden werden in:

- Tauchhärten (Metallbad oder Salzbad),
- Flammhärten (Gasbrenner),
- Induktionshärten (Induktionsspule),
- Laserstrahlhärten (Lasengerät) und
- Elektronenstrahlhärten (Elektronenkanone).

Unter thermochemischem Behandeln sind Diffusionsverfahren zu verstehen, bei denen insbesondere Metalle oder Nichtmetalle mittels gezielter Diffusion zu einer Veränderung des Gefüges und der chemischen Zusammensetzung der Randschichten führen sollen. Je nach eindiffundierendem Element unterscheidet Läßle [62] zwischen:

- Aufkohlen (Kohlenstoff),
- Carbonitrieren (Kohlenstoff und Stickstoff),
- Borieren (Bor),
- Chromieren (Chrom),
- Nitrieren (Stickstoff),
- Nitrocarburieren (Stickstoff und Kohlenstoff),
- Aluminiumieren (Aluminium),
- Silizieren (Silizium),
- Titanieren (Titan) und
- Verzinken (Zink).

Auf das Aufkohlen und das Carbonitrieren folgt i. d. R. immer auch ein späteres Härten (Einsatzhärten). Auf den genauen Ablauf dieser Verfahren soll hier nicht näher eingegangen werden. Für eine ausführliche Beschreibung sei auf Läßle [62] und Liedtke et al. [61] verwiesen.

3.3.3 Vergüten

Wie bereits erläutert, zielen Glühverfahren eher auf eine hohe Zähigkeit bzw. möglichst gute Verarbeitbarkeit ab, wohingegen Härteverfahren das Ziel eines möglichst harten und festen Werkstücks verfolgen. Ziel des Vergütens ist es nun, die Festigkeit und Zähigkeit optimal auf die durch das jeweilige Anwendungsgebiet vorliegenden Bedingungen anzupassen.

Nach DIN EN 10052 [30] ist Vergüten „eine Wärmebehandlung zum Erzielen hoher Zähigkeit bei bestimmter Zugfestigkeit durch Härten und anschließendes Anlassen meist auf höhere Temperaturen“. Somit ist das Vergüten eigentlich eine Kombination aus zwei Wärmebehandlungen, wobei das Anlassen jedoch generell selten ohne vorheriges Härten durchgeführt wird. So dient die nachfolgende Erwärmung durch das Anlassen hauptsächlich dazu, die Zähigkeit gehärteter Werkstücke zu erhöhen und die Sprödigkeit bzw. das Rissrisiko zu mindern.

Neben diesem Verfahren, das auch als Anlassvergüten bezeichnet wird, zählt

Läpple [62] auch das Bainitisieren und Perlitisieren zu der Gruppe der Vergütungsverfahren. Diese Zuordnung ist jedoch nicht ganz eindeutig. So teilen beispielsweise Liedtke et al. [61] das Bainitisieren und Perlitisieren eher in den Bereich spezieller Härteverfahren ein, da der Ablauf beider Verfahren dem des Härtens in der Erwärm- und Haltephase gleicht. Der Hauptunterschied zum „normalen“ Härten besteht darin, dass beim Bainitisieren und Perlitisieren das Werkstück während des Abschreckens für eine bestimmte Zeit in einem Warmbad (Metall- oder Salzbad) gehalten wird. Untereinander unterscheiden sich das Bainitisieren und Perlitisieren hauptsächlich hinsichtlich der jeweiligen Badtemperatur. Diese liegt beim Bainitisieren je nach Stahlsorte i. d. R. zwischen 300 °C und 400 °C und beim Perlitisieren zwischen 500 °C und 600 °C.

Anhand der Definition des Vergütens lässt sich bereits feststellen, dass das Anlassvergüten das gebräuchlichste Vergütungsverfahren darstellt. Je nach Verwendungszweck bzw. angestrebten Eigenschaftszielwerten unterscheidet Läpple [62] vier Anlassenstufen, wobei die angegebenen Temperaturbereiche lediglich als Anhaltswerte zu verstehen sind:

1. Anlassenstufe (100 °C - 200 °C),
2. Anlassenstufe (200 °C - 320 °C),
3. Anlassenstufe (320 °C - 400 °C),
4. Anlassenstufe (400 °C - 700 °C).

Ein Anlassen auf der 1. Stufe entspricht eher noch einem Entspannen als einem Vergüten. Die Härte des Werkstücks bleibt weitestgehend erhalten, die Rissgefahr infolge der hohen Spannungen wird jedoch erheblich gemindert.

In der 2. Anlassenstufe nehmen Festigkeit und Härte des Werkstücks bereits deutlicher ab. Die Zähigkeit ändert sich hingegen kaum. In vielen Fällen kann es sogar zu einer Zähigkeitsmindernden Anlassversprödung kommen.

Eine Behandlung auf der 3. Anlassenstufe resultiert in einer weiteren Abnahme von Festigkeit und Härte. Die Zähigkeit kann sich zwar leicht verbessern, die Gefahr der Anlassversprödung besteht jedoch auch in diesem Temperaturbereich.

Beim Anlassen auf der 4. Stufe muss zwischen unlegiertem und legiertem Stahl

unterschieden werden. Während die Festigkeitswerte unlegierter Stähle stark abnehmen, können sie bei legierten Stählen abhängig von den Legierungselementen nur leicht abfallen oder sogar leicht ansteigen. Zähigkeit und Verformbarkeit nehmen bei beiden Stahlarten im Allgemeinen deutlich zu.

Die bereits angesprochene Versprödungsgefahr tritt hauptsächlich im Bereich zwischen 200 °C und 500 °C auf. Aus diesem Grund wird das Anlassen auf der 2. und 3. Stufe i. d. R. vermieden. Die Behandlungstemperaturen haben beim Anlassen einen wesentlich größeren Einfluss auf die Härte als die Zeit, in der die jeweilige Temperatur gehalten wird. So wird die Haltedauer eher abhängig von der Stahlsorte und vom Werkstückquerschnitt festgelegt. Sie kann zwischen wenigen Minuten bis hin zu mehreren Stunden andauern.

Hollomon und Jaffe [49] haben einen Zahlenwert für die Wirkung des Anlassens in Abhängigkeit von Anlasstemperatur und Anlassdauer eingeführt. Die Errechnung des sogenannten Zeit-Temperatur-Parameters (bzw. Hollomon-Jaffe-Parameters) kann allgemein durch folgende Formel dargestellt werden:

$$HJP = AT \cdot (C + \lg AD) \quad (3.1)$$

$$\begin{aligned} HJP &= \text{Hollomon-Jaffe-Parameter} \\ AT &= \text{Anlasstemperatur} \\ C &= \text{werkstoffabhängige Konstante} \\ AD &= \text{Anlassdauer} \end{aligned}$$

Aus dieser Abhängigkeit lässt sich rückschließen, dass unterschiedliche Kombinationen aus Anlasstemperatur und Anlassdauer in bestimmten Bereichen zum gleichen Anlasseffekt führen können. Diesem Sachverhalt kommt im Laufe dieser Arbeit noch eine gesteigerte Bedeutung zu.

3.4 Wärmebehandlungsbegriffe, -verfahren und -einrichtungen der Dillinger Hüttenwerke

Da die zu behandelnden Werkstücke sowohl komplette Walztafeln als auch Teile einer Tafel oder einzelne Bleche umfassen können, wird bei den Dillinger

Hüttenwerken in diesem Zusammenhang allgemein von *Waben* gesprochen. Die Wärmebehandlung einer Wärmebehandlungseinheit (Wabe) kann durch eines oder durch Kombination mehrerer der folgenden Verfahren erfolgen:

- Normalisieren,
- Effusionsglühen,
- Kantenentspannen,
- Quetten,
- Anlassen,
- Vorwärmen.

Das *Normalisieren* entspricht einem Normalglühen. Es ist das mit Abstand vorherrschende Glühverfahren im Dillinger Walzwerk. Die Glühtemperaturen von ca. 910 °C werden i. d. R. maximal 15 Minuten gehalten.

Das *Effusionsglühen* ist eine Wärmebehandlung zum Wasserstoffarm- bzw. Wasserstofffreiglühen. Es zählt nicht zu den klassischen Glühverfahren und ist demnach in Kapitel 3.3 auch nicht aufgeführt bzw. behandelt. Ziel ist, die Effusion der im Gefüge eingelagerten Wasserstoffatome, welche das Werkstück spröde machen. Dies geschieht bei Glühtemperaturen von ca. 550 °C und einer Haltedauer von ca. 24 Stunden.

Ein weiteres Glühverfahren der Dillinger Hüttenwerke ist das sogenannte *Kantenentspannen*. Bezogen auf die in Kapitel 3.3 erläuterten Wärmebehandlungsverfahren ist dieses Verfahren mit einem Spannungsarmglühen vergleichbar. Die Glühtemperaturen liegen zwischen 550 °C und 650 °C und werden meist nur wenige Minuten gehalten.

Das *Quetten* ist das gängige Verfahren zum durchgreifenden Härten der Waben. Die Waben werden zunächst in einem Ofen auf Temperaturen zwischen 850 °C und 950 °C erwärmt und nach einem kurzen Halten von maximal 60 Minuten dem Ofen entnommen. Das Abschrecken erfolgt direkt im Anschluss an die Ofenentnahme mittels einer Wasserkühlanlage.

Das *Anlassen* ist für viele Grobbleche der Dillinger Hüttenwerke qualitätsentscheidend. Es kann nicht nur im Anschluss an ein Härteverfahren, sondern auch

einem Glühverfahren nachfolgend oder sogar als alleiniges Wärmebehandlungsverfahren angewendet werden. Die Anlasstemperaturen liegen hauptsächlich im Bereich zwischen 550 °C und 750 °C, die Haltedauern zwischen 5 und 90 Minuten.

Das *Vorwärmen* ist im Grunde genommen keine richtige Wärmebehandlung im klassischen Sinne. Bedingt durch die Tatsache, dass es auch in Wärmebehandlungsöfen vollzogen wird, soll es hier jedoch mitaufgeführt werden. Zudem ist ein Vorwärmen relativ häufig notwendig, da die Waben bei manchen Stahlqualitäten an bestimmten Anlagen im Adjustagebetrieb warm bearbeitet werden müssen. Zu diesen Anlagen gehören z. B. die Warmrichtmaschine oder die Brennschneidanlagen. Die Temperaturen, auf die erwärmt wird, liegen üblicherweise zwischen 250 °C und 540 °C. Ein Halten findet nicht statt.

Für die Wärmebehandlung stehen der Dillinger Hütte prinzipiell mehrere Öfen zur Verfügung, die sich in verschiedenen Punkten (wie z. B. Anlagengeometrie, Temperaturbereich, Brennmedium oder Transportsystem) teilweise voneinander unterscheiden. Aus logistischer Betrachtung ist insbesondere das Transportsystem und die damit verbundene Beschickungsart von großer Bedeutung. In Tabelle 3.1 sind die einzelnen Öfen zusammengefasst und nach Transportsystem und Beschickungsart unterteilt. Zudem sind die auf den jeweiligen Öfen derzeit durchgeführten bzw. durchführbaren Wärmebehandlungsverfahren (WBH-Verfahren) angegeben.

Die drei Herdwagenöfen dürfen nicht mit den Herdwagenöfen verwechselt werden, die vor der Walzung für die Blockerwärmung zuständig sind. Der HWO1, der HWO3 und der HWO4 werden ausschließlich für die Wärmebehandlung verwendet. Wie bereits in Kapitel 2.5 erwähnt, erfolgt die Beschickung bei Herdwagenöfen satzweise. Der Ofen an sich ist eigentlich nur ein großes Gehäuse in das der Herdwagen durch eine Ofentür ein- und ausfahren kann. Die notwendigen Heizelemente sind im Herdwagen selbst integriert. So wird eine bestimmte Ofenbelegung zunächst außerhalb des Ofens auf dem Herdwagen zusammengestellt bzw. angeordnet, anschließend in den Ofen eingefahren und nach Abschluss der Wärmebehandlung schließlich wieder ausgefahren.

Die Beschickung des Einlegeofens (EO) erfolgt ebenfalls satzweise. Das Transportsystem ist zwar ein anderes als bei den Herdwagenöfen, das Prinzip aus

Tabelle 3.1: Wärmebehandlungsöfen der Dillinger Hüttenwerke

| Ofen | Transportsystem | Beschickung | WBH-Verfahren |
|--------------------------|----------------------|----------------|------------------------|
| Herdwagenofen (HWO1) | Herdwagen | satzweise | N, A, KE, VW, EF |
| Herdwagenofen (HWO3) | Herdwagen | satzweise | KE, VW |
| Herdwagenofen (HWO4) | Herdwagen | satzweise | EF |
| Einlegeofen (EO) | Chargiermaschine | satzweise | N, Q, A, KE, VW, EF |
| Normalisierofen (NO1) | Doppelhubsystem | kontinuierlich | N, A, KE, VW |
| Normalisierofen (NO2) | Doppelhubsystem | kontinuierlich | N, Q, A, KE, VW |
| Rollenherdofen (RHO1) | Gruppenrollenantrieb | kontinuierlich | Q, A |
| Rollenherdofen (RHO2) | Gruppenrollenantrieb | kontinuierlich | A, KE, VW, EF |
| Rollenherdofen (RHO3) | Einzelrollenantrieb | kontinuierlich | A, KE, VW, EF |

N = Normalisieren, Q = Quetten, A = Anlassen, KE = Kantenentspannen,
VW = Vorwärmen, EF = Effusionsglühen

logistischer Sicht jedoch vergleichbar. Die Waben werden auf feuerfesten Aufлагesteinen (sogenannten Höckern) aufgelegt und mittels einer Chargiermaschine in den Ofen eingetragen. Die Entnahme erfolgt nach abgeschlossener Wärmebehandlung ebenfalls durch die Chargiermaschine, die sich die Waben mit Hilfe von speziellen Austragarmen wieder aus dem Ofen herausgreift.

Die Normalisieröfen NO1 und NO2 haben auf der Dillinger Hütte ihren Namen erhalten, da sie hauptsächlich für das Normalisieren verwendet werden. Bezüglich des Transportsystems handelt es sich bei diesen Öfen um sogenannte Gleichschrittöfen, die wiederum zu der Gruppe der Durchlauföfen gehören. Durchlauföfen verfügen über einen vom Ofeneingang getrennten Ofenausgang, wobei die Waben den Ofen kontinuierlich vom Eingang bis zum Ausgang komplett durchfahren. Der grundsätzliche Vorgang ist mit einer Autowaschanlage vergleichbar. Bei einem Gleichschrittofen durchlaufen die Waben den Ofen über ein Doppelhubsystem, bestehend aus zwei Gruppen von mehreren parallel zueinander angeordneten Längsbalken. Dabei sind die Balken so aufgeteilt, dass

die Gruppenzugehörigkeit zweier direkt benachbarter Balken immer wechselt und demzufolge nie zwei Balken einer Gruppe direkt nebeneinander liegen. Der Transport der Waben durch den Ofen erfolgt dann über ein abwechselndes, transversales Abheben und Ablegen durch die beiden Balkengruppen.

Auch die Rollenherdöfen RHO1, RHO2 und RHO3 gehören der Gruppe der Durchlauföfen an. Sie unterscheiden sich von den Gleichschrittöfen hauptsächlich hinsichtlich der Art des Transportmechanismus. Bei einem Rollenherdofen durchlaufen die Waben den Ofen auf einer angetriebenen Rollenbahn (Rollgang). Je nach Antriebsart kann zwischen Gruppen- und Einzelrollenantrieb unterschieden werden. Während beim Gruppenantrieb nur bestimmte Rollenrollen (eine Gruppe kann z. B. aus 5 Rollen bestehen) separat angetrieben werden können, ist beim Einzelantrieb jede Rolle getrennt steuerbar. Derzeit sind der RHO1 und der RHO2 mit Gruppenantrieb und der RHO3 mit Einzelantrieb ausgestattet.

Wie aus Tabelle 3.1 zu entnehmen, ist ein Quetten nur am EO, am NO2 und am RHO1 möglich. Dies hat den einfachen Grund, dass für das Quetten eine entsprechende Wasserkühlanlage am Ofenauslauf installiert sein muss. Dies ist sowohl beim NO2 als auch beim RHO1 der Fall. Da der EO in unmittelbarer Nähe zur ACC-Kühlanlage positioniert ist, kann das Quetten auch dort durchgeführt werden.

Aus der Beschreibung der einzelnen Ofenarten lassen sich jeweilige Vor- und Nachteile ableiten. Ein grundsätzlicher Nachteil von Öfen mit satzweiser Beschickung gegenüber Öfen, die kontinuierlich beschickt werden, besteht im geringeren Materialdurchsatz. Ein Grund hierfür sind die langen Belegungs- und Entnahmezeiten, da eine fertige Ofenbelegung erst komplett vom Herdwagen abtransportiert werden muss, bevor wieder eine neue Belegung erfolgt. Neben diesen Leerzeiten resultiert ein weiterer Grund für den schlechteren Durchsatz daraus, dass eine Ofenbelegung im Prinzip nur mit gleichartigen Waben möglich ist. Gleichartig bedeutet in diesem Fall, Waben mit nahezu identischer Dicke (zwecks Durchwärmdauer), identischer Behandlungstemperatur und identischer Haltedauer. Dies schränkt den Optimierungsspielraum bezüglich der Ofenbelegung stark ein. Hinzu kommt, dass eine flächendeckend optimale Belegung des Ofens aufgrund der verschiedenen Wabenabmessungen

im Allgemeinen nicht möglich ist. Einziger Vorteil der Öfen mit satzweiser Beschickung ist ein relativ hohes, zulässiges Metergewicht.

Die Vorteile der kontinuierlich beschickbaren Öfen ergeben sich aus den erläuterten Nachteilen der Öfen mit satzweiser Beschickung. So sind sowohl bei den Gleichschritt- als auch bei den Rollenherdöfen nahezu keine Leerzeiten (nur bei großen Temperaturwechseln) vorhanden, da das Ausfahren einer fertigen Wabe gleichzeitig mit dem Einfahren einer neuen Wabe erfolgen kann. Zudem ist der Optimierungsspielraum bezüglich der Ofenbelegung höher, da diese nicht zwingend aus gleichartigen Waben bestehen muss. So besitzen Durchlauföfen i. d. R. mehrere regelbare Temperaturzonen und ermöglichen dadurch eine simultane Durchführung mehrerer Wärmebehandlungsverfahren.

Des Weiteren besitzen Rollenherdöfen zusätzliche Vorteile gegenüber Gleichschrittöfen. Das Problem bei Gleichschrittöfen liegt darin, dass der gesamte Ofeninhalt nur mit gleicher Geschwindigkeit bewegt werden kann. Zudem dürfen die Waben aus Qualitätsgründen nicht zu lange an einer Stelle im Ofen verweilen, d. h. der Ofen muss eine bestimmte Mindestgeschwindigkeit einhalten. Aus diesen beiden Gründen folgt, dass eine Wabe den Ofen so durchfahren muss, dass ihre Wärmebehandlung am Ofenausgang ankommend abgeschlossen ist. Dies impliziert i. d. R. sehr langsame Ein- und Ausfahrzeiten.

Durch die Tatsache, dass die Rollen bei Rollenherdöfen einzeln oder gruppenweise steuerbar sind, können die einzelnen Waben mit unterschiedlichen Geschwindigkeiten gefahren werden. Ein zu langes Liegen auf der gleichen Stelle ist zwar auch hier nicht erwünscht, jedoch kann dieses Problem durch leichtes Hin- und Herpendeln behoben werden. Somit kann im Prinzip die komplette Wärmebehandlung einer Wabe an der gleichen Stelle im Ofen erfolgen. Die höhere Flexibilität bezüglich der Ofenfahrweise resultiert in zeitlichen und qualitativen Vorteilen. So hat das schnellere Ein- und Ausfahren der Waben teilweise deutlich geringere Bearbeitungszeiten zur Folge. Zudem ermöglicht es eine gleichmäßigere Erwärmung von Wabenkopf- und Wabenfuß (vorderer Teil und hinterer Teil), was der Qualität zugutekommt.

Zum Abschluss dieses Kapitels sei für eine allgemeine Beschreibung bestehender Anlagensysteme zur Wärmebehandlung bzw. Erwärmung von Stahlerzeugnissen auf das LOI Taschenbuch für Thermprocess-Technik [90] verwiesen. Zu-

dem wird bereits darauf hingewiesen, dass der RHO3 in Kapitel 5 noch einmal ausführlicher vorgestellt wird, da er den Hauptbestandteil des in dieser Arbeit betrachteten Optimierungsproblems bildet.

Kapitel 4

Maschinenbelegungsprobleme

Das folgende Kapitel bietet einen Einblick in das Feld der Maschinenbelegungsprobleme. Unterkapitel 4.1 beginnt mit einer grundlegenden Problembeschreibung und zeigt auf, an welchen Stellen diese Probleme im betrieblichen Entscheidungs- und Planungsprozess auftreten können. In Unterkapitel 4.2 folgt eine formale Problembeschreibung. Hier werden elementare Begrifflichkeiten und Notation sowie ein in der Literatur gebräuchliches Klassifikationschema eingeführt. Unterkapitel 4.3 beschäftigt sich zunächst mit der Komplexität von Maschinenbelegungsproblemen, ehe anschließend gängige Lösungsverfahren vorgestellt werden. Mit Unterkapitel 4.4 wird das Kapitel schließlich durch eine separate Behandlung von Maschinenbelegungsproblemen, die eine explizite Betrachtung von beschränkten Ressourcenkapazitäten erfordern, abgeschlossen.

4.1 Beschreibung und Einordnung

In der deutschsprachigen Literatur gibt es eine Vielzahl von Begriffen (wie z. B. Ablaufplanung, Feinplanung oder Reihenfolgeplanung), die häufig synonym mit dem Begriff der Maschinenbelegungsplanung verwendet werden und sich nur bedingt voneinander abgrenzen lassen. Das Forschungsfeld ist jedoch weitestgehend durch die englischsprachige Literatur geprägt, welche die Proble-

matik meist unter dem Begriff *Scheduling* zusammenfasst. Eine allgemeingültige Beschreibung liefert Pinedo [72]: „*Scheduling is a decision-making process that [...] deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives.*“ Dabei können sowohl die zur Verfügung stehenden Ressourcen als auch die durchzuführenden Aufgaben sehr unterschiedliche Ausprägungen annehmen. So können beispielsweise Maschinen, Transportmittel, Menschen oder auch bestimmte Flächen und Räume als Ressourcen dienen. Häufige Aufgaben sind die Durchführung bestimmter Fertigungs-, Transport- oder Projektteilschritte.

Aus obiger Formulierung lässt sich ableiten, dass man konkret von Maschinenbelegungsproblemen spricht, wenn es sich bei den Ressourcen um Maschinen und bei den Aufgaben um entsprechende Fertigungsaufträge handelt. Da dies auf die ab Kapitel 5 betrachtete Problemstellung zutrifft, wird dieser Begriff im weiteren Verlauf der Arbeit verwendet. Die obige, allgemeingültige Beschreibung der Problematik unter dem Begriff *Scheduling* wurde hier nur bevorzugt, da es für die mathematische Formulierung oder die Lösungsmethode derartiger Probleme meist nicht von Bedeutung ist, welche Ressourcen oder Aufgaben genau vorliegen. Vielmehr sind viele Probleme auf andere Probleme übertragbar oder besitzen ähnliche Bestandteile und können demnach auf ähnliche Weise gelöst werden.

Die Maschinenbelegungsplanung kann im betrieblichen Entscheidungsprozess in den Aufgabenbereich der Produktionsplanung und -steuerung (PPS) eingeordnet werden. Das Aachener PPS-Modell, ein insbesondere im deutschsprachigen Raum weit verbreitetes Modell (siehe Schuh [76]), beschreibt und unterscheidet in der Aufgabensicht Kernaufgaben, Netzwerkaufgaben und Querschnittsaufgaben. Abbildung 4.1 zeigt eine strukturierte Darstellung der Aufgabensicht des Aachener PPS-Modells.

Demnach bestehen die Kernaufgaben der PPS in der Produktionsprogrammplanung, der Produktionsbedarfsplanung sowie der Planung und Steuerung von Fremdbezug und Eigenfertigung. Dabei bestimmt die Produktionsprogrammplanung die in den nächsten Zeitperioden zu produzierenden Mengen der verschiedenen Produkte. Daraus leitet die Produktionsbedarfsplanung dann den erforderlichen Material- und Ressourcenbedarf ab und führt die auf diesen

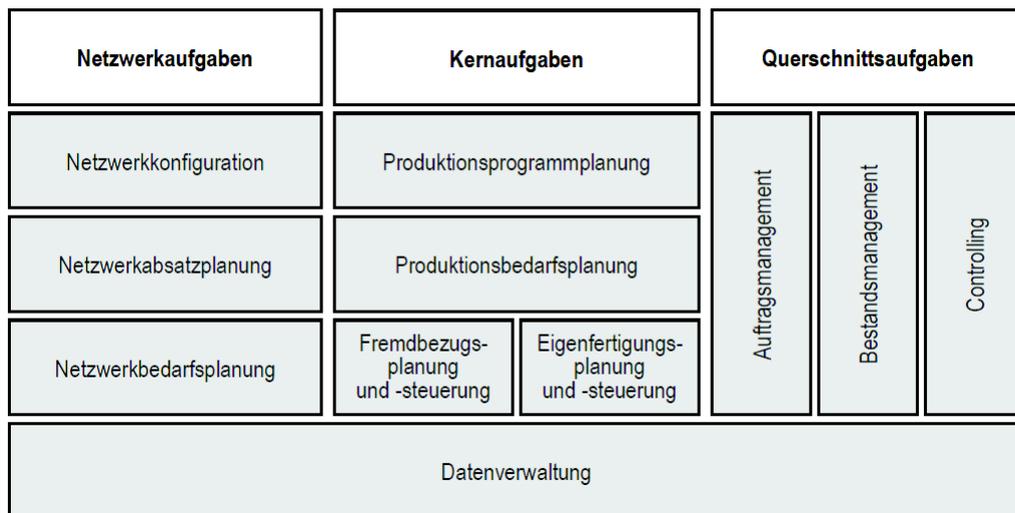


Abbildung 4.1: Aufgabensicht des Aachener PPS-Modells
(Schuh und Gierth [76])

Erkenntnissen basierende Termin- und Kapazitätsplanungen durch. Für die fremd zu beziehenden Komponenten und Teilstücke ist die Fremdbezugsplanung und -steuerung zuständig. Hier sind u. a. Entscheidungen über Bestellgrößen, Liefertermine oder die Lieferantenauswahl zu treffen. Der Eigenfertigungsplanung und -steuerung ordnet das Aachener PPS-Modell u. a. Aufgaben wie die Losgrößenrechnung, die Feinterminierung und Reihenfolgeplanung sowie die Verfügbarkeitsprüfung zu. Die Gruppe der Netzwerkaufgaben, bestehend aus Netzwerkconfiguration, Netzwerkabsatzplanung und Netzwerksbedarfsplanung, stellen den überbetrieblichen Aspekt auf strategischer Ebene dar. Die Gruppe der Querschnittsaufgaben bildet das Bindeglied zwischen Netzwerk- und Kernaufgaben und sorgt demnach für die gesamtheitliche Betrachtung des Produktionssystems. Hierzu zählen u. a. die Koordination der Auftragsabwicklung (Auftragsmanagement), die Bereitstellung von Lagergütern (Bestandsmanagement) sowie das Messen der logistischen Zielerreichung (Controlling). Da schließlich alle genannten Aufgaben auf verschiedene Daten zurückgreifen, ist für eine erfolgreiche Ausführung eine gut strukturierte und funktionierende Datenverwaltung erforderlich.

Ausgehend vom Aachener PPS-Modell sind Maschinenbelegungsprobleme je nach betrachtetem Detaillierungsgrad der Planung prinzipiell den Kernaufgaben der Produktionsbedarfsplanung und der Eigenfertigungsplanung und -steuerung zuzuordnen. Insbesondere aus Letzterem leitet Lödning [60] das

Modell der Fertigungssteuerung ab (siehe Abbildung 4.2). Das Modell setzt sich aus vier unterschiedlichen Elementen zusammen: den Aufgaben, Stellgrößen, Regelgrößen und Zielgrößen. Diese sind durch Wirkzusammenhänge derart miteinander verknüpft, dass die Aufgaben die Stellgrößen festlegen und die Regelgrößen sich aus Abweichungen zweier Stellgrößen ergeben und zudem die Zielgrößen bestimmen.

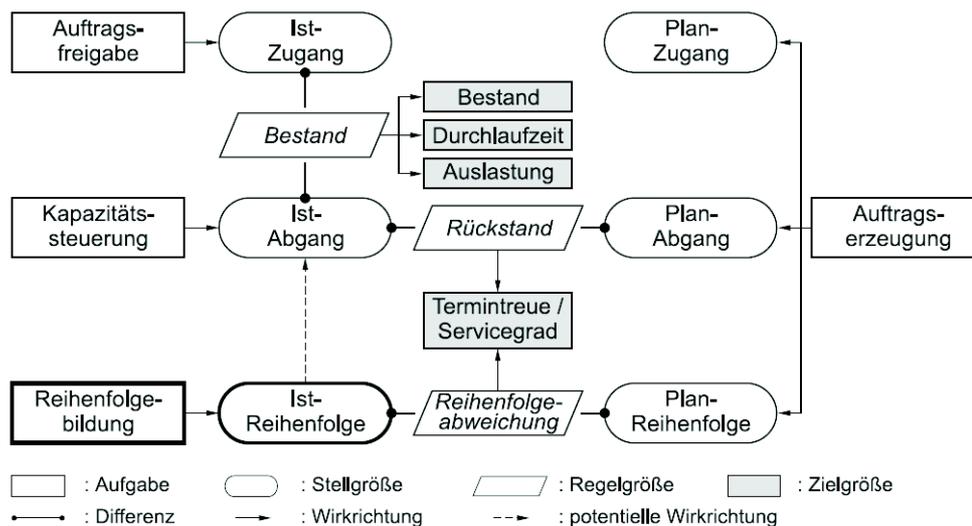


Abbildung 4.2: Das Modell der Fertigungssteuerung nach Lödding (Lödding [60])

Die Auftragserzeugung generiert aus dem Produktionsprogramm Fertigungsaufträge und legt somit den Plan-Zugang, den Plan-Abgang und die Plan-Reihenfolge der Auftragsabarbeitung fest. Die Auftragsfreigabe bestimmt den Zeitpunkt, ab dem die Fertigung einen Auftrag bearbeiten darf und legt damit den Ist-Zugang für die Fertigung fest. Demgegenüber wird der Ist-Abgang hauptsächlich durch die Kapazitätssteuerung beeinflusst, die über Einsatz und Verteilung der Kapazitäten entscheidet. Auftragsfreigabe und Kapazitätssteuerung wirken somit in hohem Maße auf die logistischen Zielgrößen Bestand, Durchlaufzeit und Auslastung. Primäre Aufgabe der Reihenfolgebildung ist die Bestimmung der Reihenfolge, in welcher die Aufträge an einem Arbeitssystem abgearbeitet werden (Ist-Reihenfolge). Abweichungen zwischen Ist-Reihenfolge und Plan-Reihenfolge sind dabei möglichst gering zu halten, um eine hohe Termintreue bzw. einen hohen Servicegrad sicherzustellen. Dabei ist allerdings zu beachten, dass sich diese Zielsetzung negativ auf die Ziele Bestand, Durch-

laufzeit und Auslastung auswirken kann, da die Ist-Reihenfolge in bestimmten Fällen (beispielsweise bei Existenz von aufwändigen Rüstvorgängen) auch den Ist-Abgang der Fertigung und damit die Leistung zum Teil stark beeinflusst. Die Reihenfolgebildung sollte sich daher immer an den Gewichtungen der verschiedenen Zielsetzungen orientieren.

Im Modell der Fertigungssteuerung nach Lödding [60] stehen Maschinenbelegungsprobleme hauptsächlich im Zusammenhang mit der Reihenfolgebildung. Wie das Modell zeigt, wirken jedoch alle Aufgaben auf die logistischen Zielgrößen ein und sollten demnach immer gesamtheitlich betrachtet werden. Nach dieser kurzen Einführung in Maschinenbelegungsprobleme und deren Einordnung in den betrieblichen Entscheidungs- und Planungsprozess werden im folgenden Unterkapitel einige Notationen und ein Klassifikationsschema vorgestellt.

4.2 Notationen und Klassifizierung

Grundsätzlich bestehen Maschinenbelegungsprobleme aus:

- einer Menge M von Maschinen (engl. *machines*) mit $i= 1, \dots, m$ und
- einer Menge J von Aufträgen (engl. *jobs*) mit $j= 1, \dots, n$.

Dabei ist ein Auftrag j i. d. R. mit den folgenden Daten behaftet:

- Bearbeitungsdauer (engl. *processing time*) p_j ,
- Freigabezeitpunkt (engl. *release date*) r_j ,
- Fälligkeitstermin (engl. *due date*) dd_j ,
- Prioritätsfaktor (engl. *weight*) w_j .

Pinedo [72] unterscheidet zwischen zwei grundsätzlichen Modellgruppen: den deterministischen und den stochastischen Maschinenbelegungsproblemen. Der Unterschied besteht darin, dass Auftragsdaten wie Bearbeitungsdauern, Freigabezeitpunkte oder Fälligkeitstermine bei deterministischen Modellen exakt im Voraus bekannt sind. Bei stochastischen Modellen ist dies nicht der Fall. Für diese Daten stehen dann nur Wahrscheinlichkeitsverteilungen zur Verfügung.

Die vorliegende Arbeit wird sich auf deterministische Modelle beschränken.

In der Literatur hat sich ein weitgehend einheitliches Klassifikationsschema für deterministische Maschinenbelegungsprobleme durchgesetzt (vgl. Graham et al. [43]). Das Schema beschreibt und unterscheidet Probleme mittels eines Tripels $(\alpha|\beta|\gamma)$, bestehend aus:

- der Maschinencharakteristik α ,
- der Auftragscharakteristik β sowie
- der Zielcharakteristik γ .

Die folgende Beschreibung potentieller Ausprägungen der einzelnen Charakteristika ist an die von Pinedo [72] angelehnt. Dazu sei bereits erwähnt, dass jede Charakteristik prinzipiell mehrere Ausprägungen aufweisen kann.

4.2.1 Maschinencharakteristik

Die Maschinencharakteristik beschreibt die Anzahl und Anordnung der zur Verfügung stehenden Maschinen. Sie kann folgende Ausprägungen annehmen:

- eine verfügbare Maschine ($\alpha = 1$),
- identische parallele Maschinen ($\alpha = Pm$),
- uniforme parallele Maschinen ($\alpha = Qm$),
- heterogene parallele Maschinen ($\alpha = Rm$),
- *Flow Shop* ($\alpha = Fm$),
- *Flexible Flow Shop* ($\alpha = FFc$),
- *Job Shop* ($\alpha = Jm$),
- *Flexible Job Shop* ($\alpha = FJc$),
- *Open Shop* ($\alpha = Om$).

Im Fall $\alpha = 1$ steht im Gegensatz zu allen anderen Ausprägungen nur eine Maschine zur Verfügung. Dieser Fall stellt demnach die einfachste Ausprägung in Bezug auf die Maschinencharakteristik dar.

Bei den nächsten drei gelisteten Ausprägungen liegen parallele Maschinen vor, wobei dies nicht bedeutet, dass die Maschinen räumlich parallel zueinander stehen. Die Parallelität bezieht sich hier auf die Produktionsstufe, d. h. Aufträge können prinzipiell an mehreren Maschinen bearbeitet werden. Die Unterscheidung zwischen identischen, uniformen und heterogenen Maschinen bezieht sich auf die Produktionsgeschwindigkeit v^p . Bei identischen Maschinen ist sie unabhängig voneinander (v^p), bei uniformen Maschinen abhängig von der Maschine (v_i^p) und bei heterogenen Maschinen sowohl maschinen-, als auch auftragsabhängig (v_{ij}^p).

Von den parallel angeordneten Maschinen unterscheiden sich die *Flow Shop* Modelle durch die Tatsache, dass die Maschinen hier nicht mehr alle auf der gleichen Produktionsstufe liegen, sondern sich über mehrere Stufen verteilen. Je nach Anzahl der Produktionsstufen liegen demnach mehrere Arbeitsgänge für einen Auftrag vor. Beim klassischen *Flow Shop* befindet sich auf jeder Produktionsstufe genau eine Maschine, d. h. alle Aufträge müssen alle Maschinen durchlaufen. Zudem ist dadurch die Reihenfolge der Arbeitsgänge eindeutig festgelegt. Bei Vorliegen eines *Flexible Flow Shop* (oder auch *Hybrid Flow Shop* oder *Multi-Processor Flow Shop* genannt) können dagegen auf einer Produktionsstufe mehrere parallel angeordnete Maschinen existieren.

Beim *Job Shop* werden Maschinen nicht mehr festen Produktionsstufen zugeordnet, da davon ausgegangen wird, dass sich die Aufträge hier hinsichtlich der Reihenfolge der Arbeitsgänge voneinander unterscheiden. Somit durchläuft jeder Auftrag die Maschinen entsprechend seiner vordefinierten Route. Bei Vorliegen eines *Flexible Job Shop* gilt dies zwar auch, jedoch sind die Routen hierbei nicht auf Maschinenebene, sondern auf Arbeitsstättenebene vorgegeben, wobei sich Arbeitsstätten aus mehreren parallel angeordneten Maschinen zusammensetzen.

Gestaltet sich die Maschinencharakteristik dagegen als *Open Shop*, so existieren keine Vorgaben in Bezug auf die Reihenfolgen der Arbeitsgänge. Demnach können die Aufträge in beliebiger Reihenfolge bearbeitet werden, müssen jedoch i. d. R. jede Maschine durchlaufen.

4.2.2 Auftragscharakteristik

Die Auftragscharakteristik beschreibt unterschiedliche Eigenschaften oder Anforderungen der Aufträge. Es können folgende Ausprägungen vorliegen:

- Freigabetermine ($\beta = r_j$),
- Unterbrechbarkeit ($\beta = prmp$),
- Reihenfolgebeziehungen ($\beta = prec$),
- Reihenfolgeabhängige Rüstvorgänge ($\beta = s_{jj'}$),
- Auftrags-Familien ($\beta = fmls$),
- *Batch*-Prozesse ($\beta = batch(b)$),
- Störfälle ($\beta = brkdown$),
- Maschineneignung ($\beta = M_j$),
- Permutation ($\beta = prmu$),
- Blockieren ($\beta = block$),
- kein Warten ($\beta = nwt$),
- Wiederverwendung ($\beta = rcrc$).

Nimmt β die Ausprägung r_j an, so bedeutet dies, dass die Aufträge mit Freigabeterminen behaftet sind. Daraus folgt, dass ein Auftrag erst bearbeitet werden darf, wenn er freigegeben bzw. bereitgestellt ist. Wenn diese Ausprägung nicht gegeben ist, kann jeder Auftrag prinzipiell zu jedem Zeitpunkt starten.

Liegt Unterbrechbarkeit (engl. *preemption*) vor, so kann die Bearbeitung von Aufträgen zu jedem Zeitpunkt unterbrochen und zu einem späteren Zeitpunkt fortgesetzt werden. Nach der Unterbrechung eines Auftrags können zudem andere Aufträge vorgezogen werden. Dabei geht die Zeit, die der unterbrochene Auftrag bereits bearbeitet wurde, nicht verloren. Sie wird von der Gesamtbearbeitungsdauer abgezogen und bildet somit die verbleibende Bearbeitungszeit für den Auftrag.

Im Fall von $\beta = prec$ liegen Reihenfolgebeziehungen bzw. -restriktionen (engl. *precedence constraints*) zwischen einzelnen Aufträgen vor. Das bedeutet, dass bestimmte Aufträge nicht bearbeitet werden dürfen, bevor ein oder mehrere andere Aufträge nicht fertiggestellt wurden.

Bei Vorliegen reihenfolgeabhängiger Rüstvorgänge (engl. *sequence dependent setup times*) sind zwischen einigen Aufträgen Rüstzeiten $st_{jj'}$ zu beachten, falls Auftrag j und Auftrag j' direkt aufeinander folgen. Diese können symmetrisch ($st_{jj'} = st_{j'j}$), asymmetrisch ($st_{jj'} \neq st_{j'j}$) oder auch maschinenabhängig ($st_{ijj'}$) vorgegeben sein.

Auch bei $\beta = fmls$ liegen i. d. R. reihenfolgeabhängige Rüstvorgänge vor, in diesem Fall jedoch nicht zwischen jedem Auftrag. Jeder Auftrag ist einer von F Auftrags-Familien (engl. *job families*) zugeordnet. Rüstzeiten $st_{gg'}$ zwischen zwei aufeinander folgenden Aufträgen entstehen dann nur, wenn die Aufträge unterschiedlichen Familien ($g \neq g'$) zugehörig sind.

Im Fall $\beta = batch(\bar{b})$ wird davon ausgegangen, dass eine Maschine eine bestimmte Anzahl von Aufträgen (\bar{b}), die in einem sogenannten *Batch* zusammengefasst sind, gleichzeitig bearbeiten kann. Hierbei ist zu beachten, dass bei einem solchen *Batch*-Prozess die Bearbeitung aller Aufträge innerhalb eines *Batches* gleichzeitig beginnt und endet, d. h. die Bearbeitungsdauer eines *Batches* wird durch die Bearbeitungsdauern der darin enthaltenen Aufträge determiniert.

Das Vorliegen bzw. die explizite Betrachtung von Störfällen (engl. *breakdowns*) impliziert, dass eine oder mehrere Maschinen für bestimmte Zeitperioden nicht zur Verfügung stehen. Entsprechend der Beschränkung auf deterministische Modelle sind diese Ausfallzeiten exakt vorgegeben.

Die Ausprägung $\beta = M_j$ deutet auf die Existenz von Restriktionen bezüglich der Maschineneignung (engl. *machine eligibility constraints*) hin. Damit ist gemeint, dass nicht jeder Auftrag auf jeder Maschine bearbeitet werden kann. Demnach beinhaltet die Menge M_j alle Maschinen, auf denen die Bearbeitung von Auftrag j prinzipiell möglich ist.

Der Fall $\beta = prmu$ tritt bei *Flow Shops* auf, wenn die Aufträge die Maschinen nach der FIFO-Regel durchlaufen sollen. In diesem Fall muss die Reihenfolge bzw. Permutation (engl. *permutation*) der Bearbeitung an jeder Maschine gleich sein.

Auch das Blockieren (engl. *Blocking*) steht im Zusammenhang mit *Flow Shops*. Die Annahme ist hierbei, dass zwischen zwei hintereinander geschalteten Maschinen beschränkte Lagerkapazitäten bestehen. Wenn ein solches Lager nicht über genügend freie Kapazität verfügt, kann der Auftrag auf der vorgeschalteten Maschine, trotz Erreichens seiner Fertigstellung auf dieser Maschine, diese nicht verlassen. Solange bis das Lager genügend freie Kapazität bereitstellt, ist die vorgelagerte Maschine demnach blockiert und steht somit für die Bearbeitung weiterer Aufträge nicht zur Verfügung.

Die Ausprägung $\beta = nwt$, ebenfalls nur bei *Flow Shops* auftretend, verbietet ein Warten oder Verharren (engl. *no-wait*) von Aufträgen zwischen der Fertigstellung auf einer Maschine und dem Beginn der Bearbeitung auf der nächsten Maschine. Es versteht sich von selbst, dass diese Ausprägung nur bei Vorliegen der *FIFO*-Regel anfallen kann.

Der letzte hier beschriebene Fall $\beta = rerc$ steht im Zusammenhang mit *Job Shops* oder *Flexible Job Shops*. Er tritt ein, wenn eine Wiederverwendung (engl. *recirculation*) von Maschinen bzw. Arbeitsstätten möglich ist und demnach ein Auftrag dieselbe Maschine bzw. Arbeitsstätte mehrmals durchlaufen kann.

4.2.3 Zielcharakteristik

Die Zielcharakteristik beschreibt und definiert die konkrete Zielsetzung. Folgende Ausprägungen kommen im Allgemeinen in Betracht:

- Gesamtfertigungszeit ($\gamma = C_{max}$),
- Summe gewichteter Fertigstellungszeiten ($\gamma = \sum w_j C_j$),
- Summe diskontierter, gewichteter Fertigstellungszeiten ($\gamma = \sum w_j (1 - e^{-rC_j})$),
- maximale Terminabweichung ($\gamma = L_{max}$),
- Summe gewichteter Verspätungen ($\gamma = \sum w_j T_j$),
- gewichtete Anzahl von Verspätungen ($\gamma = \sum w_j U_j$),
- Summe gewichteter Verfrühungen und Verspätungen ($\gamma = \sum w'_j E_j + \sum w''_j T_j$).

Im ersten Fall wird der von allen Aufträgen späteste Fertigstellungszeitpunkt (als C_{max} definiert) minimiert. Dieser Zeitpunkt entspricht automatisch dem Zeitpunkt, zu dem der letzte betrachtete Auftrag das System verlässt und ist damit dem Gesamtfertigstellungszeitpunkt bzw. der Produktionsspanne (engl. *makespan*) gleichzusetzen.

Bei $\gamma = \sum w_j C_j$ sind die einzelnen Fertigstellungszeitpunkte (C_j) der Aufträge mit ihrer Dringlichkeit entsprechenden Gewichtungen behaftet. Das Ziel besteht darin, die Summe aus diesen gewichteten Fertigstellungszeitpunkten (engl. *total weighted completion time*) zu minimieren.

Auch im Fall ($\gamma = \sum w_j (1 - e^{-rC_j})$) werden gewichtete Fertigstellungszeitpunkte minimiert. Die Gewichtungen werden hier allerdings über diskrete Zeiteinheiten hinweg mit einem Diskontierungsfaktor r diskontiert.

Die Ausprägung L_{max} deutet auf das Ziel der Minimierung der maximalen Terminabweichung (engl. *maximum lateness*) hin. Dabei ergibt sich die Terminabweichung L_j eines Auftrags j aus der Differenz zwischen Fertigstellungszeitpunkt und Fälligkeitstermin ($C_j - dd_j$).

Nicht zu verwechseln mit der Terminabweichung eines Auftrags j , ist dessen Verspätung T_j , die sich aus $\max\{L_j, 0\}$ ergibt. Der Unterschied besteht darin, dass sich Verspätungen einzelner Aufträge hier nicht mit einer Fertigstellung vor dem Fälligkeitstermin anderer Aufträge ausgleichen lassen. Bei dem Fall $\gamma = \sum w_j T_j$ wird die Summe der gewichteten Verspätungen (engl. *total weighted tardiness*) minimiert.

Auch bei Vorliegen von $\gamma = \sum w_j U_j$ bilden Verspätungen das Zielkriterium, nur die Art der Bestrafung ist eine andere. Unabhängig von der Höhe der Verspätung eines Auftrags j wird hier nur die Tatsache der Verspätung U_j bestraft. Demnach wird in diesem Fall die Anzahl gewichteter Verspätungen (engl. *weighted number of tardy jobs*) minimiert.

In manchen Fällen kann auch gewünscht sein, dass für einen Auftrag j , dessen verfrühte Fertigstellung bzw. Verfrühung E_j bestraft werden soll. Dies tritt dann häufig in der Form $\gamma = \sum w'_j E_j + \sum w''_j T_j$ auf, welche die Zielsetzung der Minimierung der Summe aus den gewichteten Verfrühungen und Verspätungen

(engl. *total weighted earliness plus total weighted tardiness*) verfolgt.

4.3 Komplexität und Lösungsverfahren

Vor der Entwicklung oder Anwendung eines bestimmten Verfahrens zur Lösung eines Optimierungsproblems ist es sinnvoll, Aussagen über dessen Komplexität treffen zu können. Dabei liegt dem in dieser Arbeit verwendeten Komplexitätsbegriff primär der bis zur Erreichung der optimalen Lösung erforderliche Rechenaufwand zugrunde. Diesbezüglich lassen sich Optimierungsprobleme im Wesentlichen in zwei Komplexitätsklassen unterteilen, die Domschke und Drexl [32] wie folgt definieren:

- „Die mit polynomialem Aufwand lösbaren Probleme gehören zur Klasse P .“
- „Probleme, für die man bislang keinen Algorithmus kennt, der auch das am schwierigsten zu lösende Problem desselben Typs mit polynomialem Aufwand löst, gehören zur Klasse der NP -schweren Probleme.“

Wie die Definition der NP -schweren Probleme zeigt, liegt dieser eine *Worst-Case*-Sicht zugrunde. Im Umkehrschluss bedeutet dies, dass für einige Problemstellungen innerhalb dieser Klasse durchaus Algorithmen mit polynomialem Lösungsaufwand existieren können. Für tiefergehende Ausführungen in die Komplexitätstheorie sei an dieser Stelle auf Garey und Johnson [37], Papadimitriou [71] und Wegener [99] verwiesen.

Blazewicz et al. [9] schlagen ein Analyseschema für *Scheduling*-Probleme vor (vgl. Abbildung 4.3). Es empfiehlt sich, dieses Schema allgemein für den Umgang mit Optimierungsproblemen heranzuziehen.

Im ersten Schritt gilt es, zunächst herauszufinden, ob das Problem NP -schwer ist oder ob es der Klasse P angehört. Hierzu ist nicht immer zwingend eine komplizierte Komplexitätsanalyse erforderlich, da die Komplexität vieler *Scheduling*-Probleme bereits in der Vergangenheit untersucht wurde (vgl. Brucker [13] und Pinedo [72]). Gehört ein Problem der Klasse P an, so spricht man häufig auch von einem „einfachen“ Problem. In diesem Fall muss für das Problem

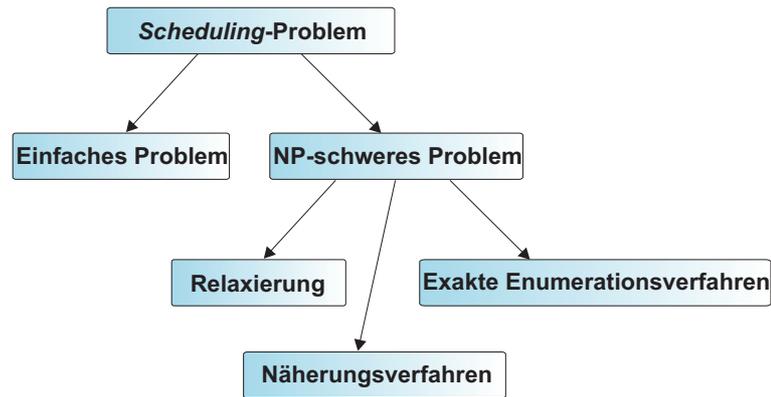


Abbildung 4.3: Ein Analyseschema für *Scheduling*-Probleme
(in Anlehnung an Blazewicz et al. [9])

grundsätzlich ein Algorithmus mit polynomialem Lösungsaufwand existieren.

Bei Vorliegen eines *NP*-schweren Problems gibt es prinzipiell drei Herangehensweisen. Zum einen kann man das Problem durch Relaxierung einer oder mehrerer Nebenbedingungen schrittweise vereinfachen, um dann zu prüfen, ob das jeweilige relaxierte Problem lösbar ist. Wenn dies der Fall und die Lösung des relaxierten Problems auch für das ursprüngliche Problem gültig ist, entspricht diese Lösung auch der optimalen Lösung für das ursprüngliche Problem. Liegt diese Gültigkeit nicht vor, so wurde zumindest eine Schranke für den optimalen Zielfunktionswert gefunden. Insbesondere in der Praxis weit verbreitet ist der Einsatz von Näherungsverfahren bzw. Heuristiken (siehe Kapitel 4.3.2), da diese i. d. R. gute Lösungen liefern und über kurze Rechenzeiten verfügen. In manchen Fällen können Heuristiken sogar zum Optimum führen. Demgegenüber können die Lösungen in anderen Fällen aber auch sehr schlecht und weit von der exakten Lösung entfernt sein. Es bietet sich bei der Verwendung von Heuristiken daher prinzipiell an, sowohl dessen *Worst-Case*-Verhalten als auch dessen Durchschnittsverhalten zu analysieren. Neben den bisher geschilderten Möglichkeiten bietet sich bei *NP*-schweren Problemen in speziellen Fällen auch der Einsatz exakter Enumerationsverfahren (siehe Kapitel 4.3.1) an. Grund hierfür ist die Tatsache, dass das Verhältnis von Rechenzeit und Problemgröße zwischen verschiedenen Instanzen eines Problems stark variieren kann. So existieren Algorithmen für bestimmte, meist eher kleine Instanzen, die einen (pseudo-)polynomialem Lösungsaufwand aufweisen. Ist dies für keine Instanz der Fall, spricht man auch von einem streng *NP*-schweren Problem.

Die meisten Maschinenbelegungsprobleme besitzen die Eigenschaft *NP*-schwer oder sogar streng *NP*-schwer zu sein (vgl. Brucker [13] und Pinedo [72]). Dies gilt insbesondere für komplexe Probleme aus der Praxis, wo Algorithmen zudem meist nur einsatzfähig sind, wenn sie über niedrige Laufzeiten verfügen. Der Einsatz exakter Verfahren ist aus diesem Grund eher selten. In den letzten Jahrzehnten hat sich sowohl in der Wissenschaft als auch in der Industrie ein verstärkter Einsatz bzw. ein verstärktes Aufkommen von Metaheuristiken abgezeichnet. Auch diese Arbeit wird sich insbesondere mit diesen Lösungsverfahren beschäftigen. In Abbildung 4.4 ist eine Klassifizierung von grundlegenden Optimierungsmethoden, insbesondere mit Hinblick auf Metaheuristiken und deren weiterer Unterteilung dargestellt.

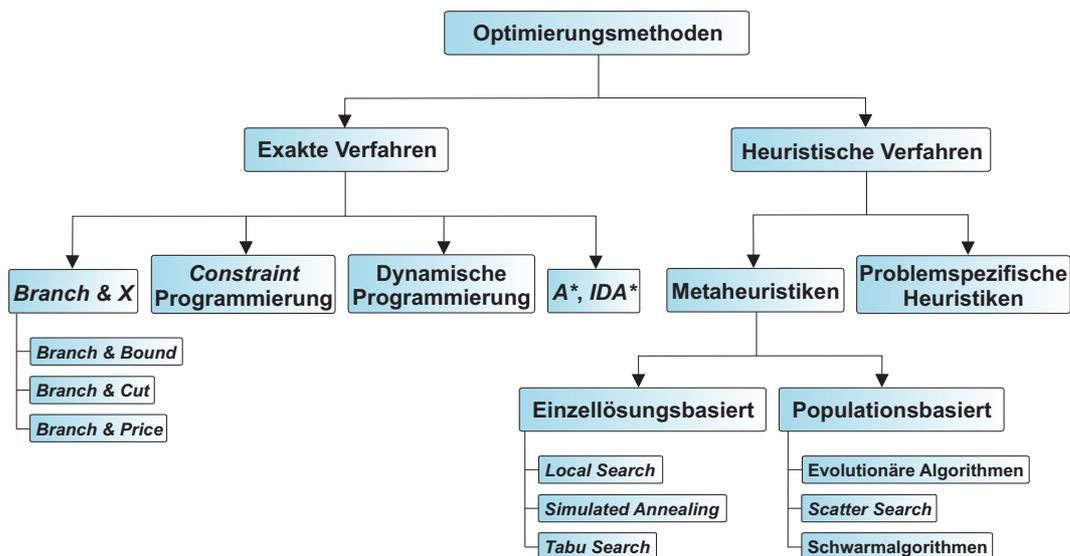


Abbildung 4.4: Eine Klassifizierung von Optimierungsmethoden mit speziellem Hinblick auf Metaheuristiken (in Anlehnung an Talbi [83])

Lösungsverfahren für Maschinenbelegungsprobleme (oder Optimierungsprobleme allgemein) lassen sich prinzipiell einer von zwei Gruppen zuordnen. Im Folgenden wird sich Kapitel 4.3.1 mit den exakten und Kapitel 4.3.2 mit den heuristischen Verfahren befassen.

4.3.1 Exakte Verfahren

Exakte Verfahren garantieren die Optimalität einer Lösung. Intuitiv könnte man auf die Idee kommen, einfach alle Lösungen aus dem Zulässigkeitsbereich zu evaluieren und zu vergleichen. Diese vollständige Enumeration ist jedoch i. d. R. mit extrem hohem Rechenaufwand verbunden und für die meisten Optimierungsprobleme ungeeignet.

Branch&X

Das grundlegende Prinzip von *Branch&Bound*-Verfahren basiert auf einer impliziten Enumeration, bei der man sukzessive versucht, Teilmengen des Zulässigkeitsbereichs zu finden, für die nachgewiesen werden kann, dass sie die Optimallösung nicht enthalten. Die wichtigsten Schritte des *Branch&Bound*-Verfahrens sind das Initialisieren, das Verzweigen (*Branching*) und das Terminieren (*Bounding*). Im Rahmen der Initialisierung wird der Zielfunktionswert für das aktuell betrachtete Teilproblem berechnet. Dieser bildet bei Minimierungsproblemen eine untere und bei Maximierungsproblemen eine obere Schranke für das Ausgangsproblem. Die aktuell beste Schranke wird gespeichert. Das Verzweigen ist gleichbedeutend mit dem Zerlegen eines (Teil-)Problems in weitere Teilprobleme. Während die Art der Problemzerlegung eher problemspezifisch ist, gibt es für das Durchsuchen des Lösungsraums grundlegende Strategien. Einen Überblick über klassische Suchstrategien bietet Winston [100]. Im Rahmen der Terminierung wird ermittelt, ob das Verzweigen eines Teilproblems grundsätzlich nicht mehr in Betracht gezogen werden muss bzw. das Problem ausgelotet werden kann. Dies ist der Fall, wenn der Lösungsbereich des Teilproblems leer ist oder dessen Zielfunktionswert schlechter als die beste bisher bekannte Schranke für das Ausgangsproblem ist.

Für einen effizienten Ablauf des *Branch&Bound*-Verfahrens sind insbesondere die Art der Problemzerlegung und die Suchstrategie entscheidend. Eine Effizienzsteigerung kann jedoch auch durch Kombination mit anderen Verfahren erzielt werden. Das *Branch&Cut*-Verfahren resultiert beispielsweise aus einer Kombination des *Branch&Bound*-Verfahrens mit dem *Cutting Plane*-Verfahren. Die Idee des *Cutting Plane*-Verfahrens besteht darin, den Lösungsraum des betrachteten Problems durch gezieltes Hinzufügen zusätzlicher Re-

Restriktionen (*cuts*) schrittweise einzuschränken, um so schneller zur optimalen Lösung zu gelangen. Das *Branch&Cut*-Verfahren macht sich dies zu nutze. Dabei wird ausgehend vom grundsätzlichen Ablauf des *Branch&Bound*-Verfahrens jedes Teilproblem wieder mit dem *Cutting Plane*-Verfahren gelöst. Dadurch steigt zwar der Rechenaufwand für die einzelnen Teilprobleme, jedoch führen die zusätzlichen Restriktionen meist zu besseren Schranken und damit meist zu einem insgesamt geringeren Rechenaufwand gegenüber reinen *Branch&Bound*-Verfahren, da sich Teilprobleme früher ausloten lassen.

Branch&Price-Verfahren arbeiten ähnlich dem *Branch&Cut*-Verfahren. Hier werden die Teilprobleme jedoch mit einem *Column Generation*-Algorithmus gelöst. Die Anwendung von *Column Generation* eignet sich insbesondere bei Problemen mit sehr hoher Variablenanzahl. In einem ersten Schritt wird das betrachtete Problem nur für eine kleine Teilmenge der Variablen des Ausgangsproblems gelöst. Um die Qualität dieser Lösung zu prüfen, muss ein sogenanntes *Pricing*-Problem gelöst werden. Dieses bestimmt die im aktuell betrachteten Problem nicht enthaltenen, aber für die optimale Lösung relevanten Variablen und überprüft anhand der reduzierten Kosten, ob durch deren Aufnahme der Zielfunktionswert verbessert werden kann. Ist eine solche Variable gefunden, wird sie dem Problem hinzugefügt, welches daraufhin von neuem gelöst wird. Kann keine solche Variable mehr gefunden werden, ist die aktuelle Lösung auch optimal für das Ausgangsproblem. Ein Effizienzvorteil von *Branch&Price*- gegenüber *Branch&Bound*-Verfahren liegt vor allem in der besseren Verarbeitung großer Variablenmengen.

Die Erläuterungen zu den unter dem Begriff *Branch&X* zusammengefassten Verfahren beschränken sich auf die Vermittlung der grundsätzlichen Verfahrensprinzipien. Eine umfassendere Beschreibung findet sich z. B. in Chen [20].

Constraint-Programmierung

Eine in den letzten Jahren zunehmend verbreitete Lösungsmethode für Optimierungsprobleme ist die *Constraint-Programmierung*, die Mitte der 1980er Jahre durch Kombination aus *Constraint-Lösen* und Logikprogrammierung entstand. Ein *Constraint*-Problem setzt sich aus einer Menge von Variablen und Restriktionen zusammen. In diesem Zusammenhang beschäftigt sich das

Constraint-Lösen mit der effizienten Suche nach einer Belegung der Variablen, die alle Restriktionen erfüllt. Idee der Logikprogrammierung ist, Probleme logisch zu beschreiben, d. h. bestehendes Wissen und getroffene Annahmen in bestimmten Regeln mittels logischer Formeln auszudrücken.

Durch die Fusion von *Constraint*-Lösen und Logikprogrammierung hat sich die *Constraint*-Programmierung zu einer für den Benutzer einfachen und eleganten Lösungsmethode für Optimierungsprobleme entwickelt. Insbesondere für *Scheduling*- bzw. Maschinenbelegungsprobleme eignet sich z. B. der von IBM angebotene ILOG CPLEX CP Optimizer [50]. Für einen tiefergehenden Einblick in das Feld der *Constraint*-Programmierung (speziell mit Hinblick auf *Scheduling*-Probleme) sei an dieser Stelle auf Rossi et al. [75] verwiesen.

Dynamische Programmierung

Die dynamische Programmierung basiert auf dem Optimalitätsprinzip von Bellman [7], welches besagt, dass sich die Optimallösung bei einigen Optimierungsproblemen aus optimalen Teillösungen zusammensetzt. Um diesem Prinzip zu genügen, müssen die Teilprobleme so gestaltet sein, dass sie voneinander abhängig bzw. aufeinander aufbauend sind. Dynamische Programmierung kann daher nicht für jedes Optimierungsproblem angewendet werden. Es eignen sich vor allem Probleme mit einem über mehrere Stufen ablaufenden Entscheidungsprozess.

Die der dynamischen Programmierung zugrundeliegende Idee sieht vor, die einzelnen Teilprobleme in jeder Stufe separat zu lösen, um aus diesen rekursiv gewonnenen optimalen Teillösungen auf die optimale Lösung des Gesamtproblems schließen zu können. Dabei kann die Komplexität eines Problems sowohl mit steigender Stufenanzahl als auch insbesondere mit steigender Anzahl der möglichen Zustände in den einzelnen Stufen stark anwachsen. Der erfolgreiche Einsatz dynamischer Programmierung hängt somit nicht nur von dem vorliegenden Optimierungsproblem an sich, sondern auch von einer effizienten Gestaltung bzw. Modellierung der Teilprobleme ab. Umfassendere Ausführungen zu dynamischer Programmierung finden sich in Bertsekas [8].

A^* , IDA*

Auf das Vorliegen und die Bedeutung verschiedener Suchstrategien wurde bereits im Zusammenhang mit der Beschreibung des *Branch&Bound*Verfahrens hingewiesen. A^* -Algorithmen entstammen dem Forschungsgebiet der künstlichen Intelligenz und gehören der Klasse der informierten Suchalgorithmen an. Diese unterscheiden sich von uninformierten („blinden“) Suchalgorithmen dadurch, dass sie Schätzfunktionen bzw. Heuristiken einsetzen, um ein zielgerichtetes und damit Laufzeit minimierendes Suchen zu ermöglichen. Die Effizienz eines solchen Suchalgorithmus hängt vor allem von der Güte der Schätzfunktion ab. Da A^* -Algorithmen i. d. R. einen hohen Speicherplatzbedarf aufweisen, haben sich zahlreiche Varianten bzw. Erweiterungen entwickelt. Ein klassisches Beispiel sind die sogenannten IDA*-Algorithmen (*iterative deepening algorithms*), die versuchen den Speicherbedarf durch iterative Tiefensuche möglichst klein zu halten. Abschließend sei für eine detailliertere Vorstellung klassischer Suchstrategien nochmals auf Winston [100] verwiesen.

4.3.2 Heuristische Verfahren

Der Begriff Heuristik stammt aus dem Griechischen (*heuriskein* = finden). Bezogen auf komplexe Optimierungsprobleme kann man Heuristiken zusammenfassend als Verfahren beschreiben, die mit möglichst geringer Laufzeit zu einer zulässigen und guten, aber nicht zwangsweise optimalen Lösung führen sollen. Dabei werden die Begriffe Heuristik und Näherungs- bzw. Approximationsverfahren häufig synonym verwendet. Genau genommen, bezeichnet man Näherungs- bzw. Approximationsverfahren jedoch als Heuristiken, für die eine *Worst-Case*-Schranke angegeben werden kann. Grundsätzlich lassen sich Metaheuristiken und problemspezifische Heuristiken unterscheiden.

Metaheuristiken

Metaheuristiken stellen universell einsetzbare Heuristiken dar und können somit auf ein breites Spektrum von Problemen angewandt werden. Die folgenden Ausführungen orientieren sich hauptsächlich an dem Werk von Talbi [83], welches einen sehr guten Überblick über klassische Verfahren und aktuelle Ent-

wicklungen im Zusammenhang mit Metaheuristiken bietet.

Wie in Abbildung 4.4 zu sehen, können grundsätzlich einzellösungsbasierte und populationsbasierte Metaheuristiken unterschieden werden. Einzellösungsbasierte Verfahren gehen, wie der Name schon sagt, auf Basis einer einzelnen Lösung vor und versuchen diese durch iteratives Transformieren zu verbessern. Populationsbasierte Verfahren betrachten dagegen in jedem Iterationsschritt mehrere Lösungen („Population“). Abhängig von deren Güte werden diese Lösungen auf eine bestimmte Weise kombiniert, um dadurch neue, möglichst bessere Lösungen zu erhalten.

Ein im Zusammenhang mit Metaheuristiken häufig auftauchender Konflikt besteht in der Wahl der Suchintensivierung bzw. -diversifizierung (*exploitation versus exploration*). Ein intensives Durchsuchen eines bestimmten Bereichs im Lösungsraum resultiert meist in einem schnellen Auffinden eines lokalen Optimums, welches jedoch nicht zwingend einem globalen Optimum entsprechen muss. Das Problem eines intensiven Suchverfahrens besteht darin, dass es in einem einmal gefundenen lokalen Optimum meist verharret. Dem versuchen diversifizierende Suchverfahren entgegenzuwirken, indem sie den gesamten Lösungsraum weitreichend durchsuchen. Dabei ist zu beachten, dass eine zu oberflächliche Suche dazu führen kann, dass die lokalen Optima in den einzelnen Bereichen gar nicht erst gefunden werden. Einzellösungsbasierte Metaheuristiken weisen meist eine eher intensive Suche, populationsbasierte Metaheuristiken dagegen eine eher diversifizierende Suche auf. Im Folgenden werden die klassischen Verfahren der beiden Gruppen kurz vorgestellt.

Zu den grundlegenden einzellösungsbasierten Metaheuristiken gehören:

- *Local Search*,
- *Simulated Annealing* und
- *Tabu Search*.

Local Search stellt die älteste und einfachste Form von Metaheuristiken dar. In einem ersten Schritt gilt es, eine zulässige Lösung zu generieren und eine Nachbarschaft zu definieren. Eine Nachbarschaft bezeichnet eine Menge von Lösungen, die von einer gegebenen Lösung durch eine einfache Veränderungsoperation erreicht werden können. In der klassischen Variante wird mit *Local*

Search in jeder Iteration die Nachbarschaft der aktuellen Lösung nach Lösungen durchsucht, die zu einer Zielfunktionswertverbesserung führen. Wird eine solche Lösung nicht gefunden, stoppt der Algorithmus. Ansonsten wird in der nächsten Iteration i. d. R. mit der Lösung fortgefahren, mit der die größte Zielfunktionswertverbesserung erzielt werden kann. In diesem Fall spricht man auch von einem *Greedy*-Verfahren. Die Problematik dieser Vorgehensweise besteht darin, dass der Algorithmus meist recht schnell in einem lokalen Optimum verharrt. Zudem hängt die Güte des lokalen Optimums in Relation zum globalen Optimum sehr stark von der gewählten Startlösung ab.

Simulated Annealing ist ein Verfahren, welches versucht, das Verharren in einem lokalen Optimum durch temporäres Akzeptieren verschlechternder Lösungen zu verhindern. Es basiert auf der Nachbildung bzw. Steuerung des physikalischen Abkühlungsprozesses von Metallen. Wird in einer Iteration in der Nachbarschaft keine verbessernde Lösung gefunden, so wird mit einer bestimmten Wahrscheinlichkeit eine verschlechternde Lösung gewählt. Diese Wahrscheinlichkeit hängt einerseits von der Differenz zwischen dem Zielfunktionswert der schlechteren Lösung und dem der aktuell besten bekannten Lösung ab. Andererseits ist sie von einer sich mit fortlaufender Iterationsanzahl verringernden Temperaturvariable abhängig, d. h. zum Anfang des Verfahrens sind verschlechternde Lösungen noch eher erlaubt als gegen Ende. Die Initialisierung und Steuerung dieser Temperaturvariable ist für einen effizienten Ablauf des Verfahrens von großer Bedeutung.

Tabu Search funktioniert auf ähnliche Weise wie *Simulated Annealing*. Auch hier sind verschlechternde Lösungen temporär erlaubt. In der klassischen Variante wird beim Durchsuchen der Nachbarschaft grundsätzlich die beste gefundene Lösung (auch wenn sie eine Verschlechterung darstellt) gewählt. Dabei besteht die Gefahr, dass der Algorithmus nach der Wahl einer verschlechternden Lösung in der darauffolgenden Iteration wieder zur vorherigen Lösung zurückspringt. Um zu verhindern, dass der Algorithmus nur noch zwischen zwei Lösungen hin und her springt, werden Lösungen für eine bestimmte Dauer tabuisiert, d. h. sie dürfen für eine bestimmte Zeit nicht gewählt werden. Hierzu muss der Lösungsverlauf regelmäßig gespeichert werden. Der Algorithmus verwaltet somit eine Art Gedächtnis. Ein entscheidender Parameter ist dabei die gesetzte Tabudauer, d. h. die Anzahl an Iterationen, für die eine Tabuisierung

gelten soll.

Populationsbasierte Metaheuristiken lassen sich nach Talbi [83] einer der drei folgenden Gruppen zuordnen:

- Evolutionäre Algorithmen,
- *Scatter Search* und
- Schwarmalgorithmen.

Das Vorbild evolutionärer Algorithmen ist, wie der Name schon sagt, die biologische Evolution bzw. die natürliche Selektion. Bezieht man diesen Mechanismus auf Optimierungsalgorithmen, so lässt sich eine einzelne Lösung als „Individuum“ und eine Lösungsmenge als „Population“ beschreiben, wobei eine Population zu einem bestimmten Zeitpunkt bzw. Iterationsschritt auch als „Generation“ bezeichnet wird. Gemäß der natürlichen Selektion vermehren sich starke bzw. gut an ihre Umwelt angepasste Individuen stärker als weniger starke bzw. schlechter angepasste Individuen. So steigt bei evolutionären Algorithmen die Wahrscheinlichkeit, dass eine bestimmte Lösung seine Bestandteile (Erbanlagen) an die nächste Generation weitergibt mit dessen Güte bezüglich des Zielfunktionswerts (*fitness*). Zu den evolutionären Algorithmen gehören beispielsweise evolutionäre Strategien sowie die evolutionäre und die genetische Programmierung. Am weitesten verbreitet sind jedoch genetische Algorithmen. Die weitere Beschreibung evolutionärer Algorithmen wird sich daher auf diese Verfahrensgruppe beschränken.

Genetische Algorithmen haben insbesondere durch Holland [47][48] bereits Mitte der 1960er Jahre in der Literatur Einzug gefunden. Ihre prinzipielle Vorgehensweise besteht darin, mit Hilfe der genetischen Operatoren Selektion, Rekombination und Mutation solange Populationen zu erzeugen bis ein bestimmtes Abbruchkriterium erfüllt ist. Die Selektion ist dafür zuständig, aus der aktuellen Population einen Genpool zu erzeugen, wobei bessere Lösungen mit höherer Wahrscheinlichkeit aufgenommen werden. Bei der Rekombination werden nun nach und nach jeweils zwei Lösungen aus dem Genpool entnommen und mit einer bestimmten Wahrscheinlichkeit gekreuzt, sodass zwei neue Lösungen entstehen. Durch eine mit bestimmter Wahrscheinlichkeit erfolgende Mutation werden die einzelnen Lösungen zufälligen, meist kleinen Veränderun-

gen unterzogen. Als Abbruchkriterium kann z. B. ein globales Zeitlimit dienen. Ist dies überschritten, stoppt der Algorithmus und gibt den besten gefundenen Lösungskandidat als Ergebnis aus. Die Effizienz genetischer Algorithmen hängt sehr stark von den angewandten Selektions-, Rekombinations-, und Mutationsverfahren sowie von dessen Eintrittswahrscheinlichkeiten ab.

Scatter Search ist ein Verfahren, welches verschiedene Metaheuristiken kombiniert. Es arbeitet ebenso wie genetische Algorithmen auf Basis einer Menge von Lösungen, welche auch als Referenzmenge bezeichnet wird. Die Lösungen werden beim *Scatter Search* jedoch auf systematischere und zielgerichtete Weise kombiniert. Hierbei kommen meist einzellösungsbasierte Metaheuristiken zum Einsatz. Anstatt eine neue Lösung immer zwingend nur aus einem alten Lösungspaar zu generieren, können beim *Scatter Search* zudem auch mehr als zwei Lösungen zu einer neuen Lösung kombiniert werden. Da diese Art der Lösungskombination i. d. R. mit relativ hohem Rechenaufwand verbunden ist, sollte die Referenzmenge grundsätzlich nicht zu groß gewählt werden.

Schwarmalgorithmen bilden eine weitere Untergruppe der populationsbasierten Metaheuristiken. Die Idee besteht darin, das Verhalten bezüglich der Futtersuche von in Schwärmen lebenden Tieren wie Ameisen, Fischen oder Vögeln nachzubilden und dieses auf Optimierungsprobleme zu übertragen. Grundsätzlich simulieren Schwarmalgorithmen eine Menge von Agenten, die miteinander kommunizieren, indem sie aus ihrer Umgebung gesammelte Informationen austauschen. Die einzelnen Agenten werden nicht zentral gesteuert, sondern gehorchen bestimmten Verhaltensregeln. Ihr Handeln ist sowohl von ihrer Umgebung abhängig, als auch teilweise zufällig gesteuert. Die zwei bekanntesten Schwarmalgorithmen (*Ant Colony Optimization* und *Particle Swarm Optimization*) werden im Folgenden kurz vorgestellt.

Ameisen können ihren Weg zur Futtersuche durch Pheromone markieren. Isoliert bewegt sich eine Ameise eher zufällig. Trifft sie jedoch auf eine Pheromonspur, folgt sie dieser mit einer umso höheren Wahrscheinlichkeit je intensiver diese Spur ist. Da Pheromone zudem verdunsten können, besitzen kaum besuchte Pfade mit fortschreitender Zeit immer weniger Anziehungspotential. *Ant Colony Optimization* überträgt die beschriebene Vorgehensweise von Ameisen bei der Futtersuche auf das Lösen von Optimierungsproblemen. So

entspricht eine Pheromonspur einer bestimmten Teillösung und die Pheromonkonzentration der Güte dieser Lösung. Zu Beginn des Algorithmus sind die ersten Lösungen noch recht breit im Lösungsraum verstreut. Mit fortlaufender Zeit intensiviert sich das Durchsuchen des Lösungsraums jedoch in bestimmten Bereichen. Für einen effizienten Ablauf des Verfahrens ist die Steuerung des Pheromongehalts mitentscheidend. Die Art der Berechnung der Wahrscheinlichkeiten, dass bestimmten Teillösungen abhängig von ihrer Pheromonkonzentration gefolgt wird, ist dabei ebenso wichtig wie die Wahl eines effizienten Verdunstungsfaktors.

Particle Swarm Optimization hat sich das Verhalten von Vögel- oder Fischschwärmen zum Vorbild genommen. Zu Beginn des Verfahrens werden verschiedene Teillösungen (Partikel) initialisiert, die möglichst breit im Lösungsraum verteilt sein sollen. In jeder der folgenden Iterationen werden die einzelnen Partikel dann individuell mit einer bestimmten Geschwindigkeit in eine bestimmte Richtung des Lösungsraums verschoben. Dabei orientiert sich ein Partikel zum einen an seiner bisher persönlich besten Lösung (kognitive Komponente) und zum anderen an der aktuell besten bekannten Lösung des gesamten Schwarms (soziale Komponente). Auf diese Weise wird versucht, sowohl einer intensiven als auch einer diversifizierenden Suche nachzukommen. Insbesondere die Steuerung des Einflusses der bisher persönlich besten Lösung und der bisher besten bekannten Lösung des Schwarms auf die Bewegung eines Partikels bildet einen entscheidenden Faktor für die Realisierung eines effizienten Suchablaufs.

Die Vorstellung der bekanntesten Metaheuristiken sei hiermit abgeschlossen. Es soll jedoch noch darauf hingewiesen sein, dass die Verfahren nur in ihren jeweiligen Basisvarianten erläutert wurden. In der Literatur finden sich sowohl zahlreiche Erweiterungen als auch Mischformen verschiedener Metaheuristiken (hybride Ansätze). Abschließend sei diesbezüglich nochmals auf Talbi [83] hingewiesen.

Problemspezifische Heuristiken

Problemspezifische Heuristiken unterscheiden sich von den universell einsetzbaren Metaheuristiken darin, dass sie speziell für ein bestimmtes Problem entwickelt wurden. Damit sind sie üblicherweise auch nur auf dieses Problem

anwendbar und somit auf andere Probleme nicht übertragbar. Im Gegenzug sind problemspezifische Heuristiken meist sehr laufzeiteffizient, da sie spezielles Wissen über die vorliegende Problemstruktur gezielt einbeziehen. Die Literatur beinhaltet eine Vielzahl von Algorithmen, die für spezielle Problemstellungen entwickelt wurden. Hier sind z.B. der *Savings*-Algorithmus von Clarke und Wright [22] für das *Vehicle Routing Problem* oder das *Silver/Meal*-Verfahren [77] zur Lösung des dynamischen Losgrößenmodells zu nennen.

Im Laufe der letzten Jahrzehnte hat der Einsatz rein problemspezifischer Heuristiken mehr und mehr abgenommen. Sie werden dafür immer häufiger mit Metaheuristiken kombiniert, wo sie meist als Eröffnungs- bzw. Startverfahren dienen, um erste gute und zulässige Lösungen zu generieren. Allgemein werden Metaheuristiken immer weniger als klassisch universelle Verfahren eingesetzt. Stattdessen wird auch hier immer häufiger gezielt problemspezifisches Wissen in Form bestimmter Verfahren einbezogen, um insbesondere den Suchraum schnell einschränken zu können und somit das Laufzeitverhalten zu verbessern. Bei diesen Verfahren handelt es sich dann wiederum weniger um eine Metaheuristik im klassischen Sinne als vielmehr um eine problemspezifische Heuristik. Der Übergang zwischen den beiden Gruppen gestaltet sich folglich immer mehr fließend.

4.4 Explizite Betrachtung beschränkter Ressourcenkapazitäten

In der Literatur wird für die meisten Maschinenbelegungsprobleme grundsätzlich angenommen, dass Maschinen zu jedem Zeitpunkt nur höchstens einen Auftrag bearbeiten können d. h. es wird implizit von einer 1-Auftrag-Kapazität pro Maschine ausgegangen. Demgegenüber stehen nur wenige Varianten, bei denen diese Annahme verworfen wird und die demnach explizit beschränkte Maschinenkapazitäten betrachten. Aufgrund ihrer Bedeutung für die in dieser Arbeit ab Kapitel 5 vornehmlich behandelte Problemstellung, werden derartige Maschinenbelegungsprobleme im Folgenden separat erläutert.

Maschinen, die prinzipiell mehrere Aufträge zur gleichen Zeit bearbeiten kön-

nen, stehen häufig in Verbindung mit einem sogenannten *Batching*-Prozess (vgl. Kapitel 4.2.2), bei dem die Aufträge nicht einzeln, sondern gruppenweise zusammengefasst die jeweilige Maschine durchlaufen. Derartige Probleme werden im Folgenden unter dem Begriff *Batching*-Probleme zusammengefasst. Andere Maschinenbelegungsprobleme, bei denen kein *Batching*-Prozess vorliegt, die aber dennoch kapazitätsbeschränkte Ressourcen betrachten, weisen teilweise Gemeinsamkeiten mit dem RCPSP (*Resource-Constrained Project Scheduling Problem*) auf, welches ursprünglich aus der Projektplanung kommt, sich jedoch leicht auf Maschinenbelegungsprobleme übertragen lässt. In den folgenden Unterkapiteln 4.4.1 und 4.4.2 werden die beiden Problemgruppen näher vorgestellt.

4.4.1 *Batching*-Probleme

In Kapitel 4.2.2 wurde *Batching* bereits mit einer Maschine in Verbindung gebracht, die prinzipiell mehrere Aufträge gleichzeitig bearbeiten kann. Dies entspricht nach Potts und Kovalyov [73] jedoch schon einer begrenzten Betrachtungsweise. Nach ihnen sind *Batching*-Probleme immer im Zusammenhang mit Auftrags-Familien ($\beta = fmls$) und ein *Batch* damit als Gruppierung von Aufträgen aus der gleichen Auftrags-Familie zu verstehen. Ein *Batch* determiniert somit lediglich eine bestimmte Zeitdauer, während der eine Resource einen bestimmten Rüstzustand hat und demnach keine Rüstaktivitäten stattfinden. Prinzipiell kann die Abarbeitung der Aufträge eines *Batches* somit sowohl simultan als auch sequentiell erfolgen. Brucker [13] bezeichnet den sequentiellen Fall als *Serial-Batching*-Problem und den simultanen Fall als *Parallel-Batching*-Problem.

Eine andere Betrachtungsweise bzw. Definition in Bezug auf *Batching* liefern Neumann et al. [67]. Sie unterteilen die Problemstellung in zwei Phasen bzw. Subprobleme. Zunächst ist das reine *Batching* durchzuführen, d. h. das Gruppierungsproblem zu lösen, wobei grundsätzliche verfahrenstechnische Restriktionen einzuhalten sind. In der zweiten Phase werden die einzelnen Aufträge nicht mehr explizit betrachtet. Die hier erfolgende Terminierung und Reihenfolgeplanung durch Lösen eines *Batch Scheduling*-Problems basiert nur noch auf den aus der ersten Phase resultierenden *Batches*. Neumann et al. [67] zeigen

zudem, dass sich dieses *Batch Scheduling*-Problem auch als RCPSP formulieren lässt.

Die unterschiedlichen Auffassungen resultieren teilweise aus dem betrachteten Industriezweig. Während Potts und Kovalyov [73] und Brucker [13] eher die fertigungsindustrielle Produktion betrachten, konzentrieren sich Neumann et al. [67] gezielt auf die Prozessindustrie. Witt [101] konkretisiert die unterschiedlichen und gemeinsamen Ansichten der beiden Industriezweige in Bezug auf *Batching*.

Um der expliziten Betrachtung von Maschinenbelegungsproblemen mit beschränkten Ressourcenkapazitäten gerecht zu werden, wird der Begriff *Batching* im weiteren Verlauf dieses Kapitels nur noch mit dem Vorliegen eines *Parallel-Batching*-Problems in Verbindung gebracht. Zudem werden sich die weiteren Ausführungen zur Vereinfachung auf einen Einmaschinenfall beschränken. Die von Uzsoy [91] betrachtete Problemstellung eignet sich als veranschaulichendes Beispiel. Das Problem besteht darin, eine Menge von Aufträgen ($j \in J$) so zu Mengen von *Batches* ($b \in B$) zusammenzufassen und zu verplanen, dass der Gesamtfertigstellungszeitpunkt minimiert wird. Neben der Bearbeitungszeit p_j besitzt jeder Auftrag noch eine bestimmte kapazitätsbeanspruchende Größe kb_j . Für die Gesamtkapazität MK der Maschine gilt: $MK \geq kb_j \forall j \in J$. Die Bearbeitungszeit eines *Batches* b wird mit p^b bezeichnet und ergibt sich aus der längsten Bearbeitungszeit der *Batch* b zugeordneten Aufträge ($p^b = \max \{p_j \forall j \in \text{Batch } b\}$).

Gemäß der in Kapitel 4.2 eingeführten Klassifizierung, lässt sich das Problem durch das Tripel $(1|\text{batch}, kb_j|C_{max})$ beschreiben. Während aus der Ausprägung $\beta = \text{batch}(\bar{b})$ implizit folgt, dass alle betrachteten Aufträge einen identischen Kapazitätsbedarf aufweisen, liegen bei Problemen der Ausprägung $\beta = \text{batch}, kb_j$ nichtidentische Kapazitätsbedarfe vor. Aus diesem Grund kann hier im Vorhinein auch nicht auf eine konstante Anzahl \bar{b} prinzipiell in einem *Batch* zusammenfassbarer Aufträge geschlossen werden.

Uzsoy [91] zeigt, dass das Problem *NP*-schwer ist und schlägt ein heuristisches Lösungsverfahren vor. Des Weiteren haben sich z. B. Melouk et al. [65] und Damodaran et al. [24] mit dem Problem beschäftigt. Bestandteil beider Artikel ist eine mathematische Formulierung des Problems als gemischt-ganzzahliges

Programm (engl. *mixed integer program (MIP)*) mit den folgenden Entscheidungsvariablen:

$$x_{jb} = \begin{cases} 1, & \text{wenn Auftrag } j \text{ Batch } b \text{ zugeordnet wird} \\ 0, & \text{sonst;} \end{cases}$$

$$y_b = \begin{cases} 1, & \text{wenn Batch } b \text{ verwendet wird} \\ 0, & \text{sonst;} \end{cases}$$

Die *MIP*-Formulierung ergibt sich wie folgt:

$$\min \sum_{b \in B} p^b \quad (4.1)$$

u.d.N.

$$\sum_{b \in B} x_{jb} = 1 \quad \forall j \in J \quad (4.2)$$

$$\sum_{j \in J} kb_j x_{jb} \leq MK y_b \quad \forall b \in B \quad (4.3)$$

$$p^b \geq p_j x_{jb} \quad \forall j \in J, b \in B \quad (4.4)$$

$$x_{jb} \leq y_b \quad \forall j \in J, b \in B \quad (4.5)$$

$$x_{jb} \in \{0, 1\} \quad \forall j \in J, b \in B \quad (4.6)$$

$$y_b \in \{0, 1\} \quad \forall b \in B \quad (4.7)$$

Die Zielfunktion (4.1) minimiert die Gesamtfertigstellungszeit. Nebenbedingung (4.2) stellt sicher, dass jeder Auftrag genau einem *Batch* zugeordnet wird. Mit Nebenbedingung (4.3) wird gewährleistet, dass bei keinem *Batch* die Kapazitätsrestriktion verletzt wird. Durch die Nebenbedingungen (4.4) lassen sich zudem die Bearbeitungszeiten der *Batches* ermitteln. Des Weiteren darf ein Auftrag nur einem *Batch* zugeordnet werden, wenn dieser auch verwendet wird bzw. muss umgekehrt ein *Batch* immer dann verwendet werden, wenn ihm auch zumindest ein Auftrag zugeordnet wurde. Diese wechselseitige Beziehung wird durch Nebenbedingung (4.5) sichergestellt. Schließlich bilden (4.6) und (4.7) die erforderlichen Binärbedingungen.

Die beschriebene Problemstellung ist Bestandteil vieler Maschinenbelegungsprobleme, die mit *Batching* in Zusammenhang stehen. In der Literatur existieren zahlreiche Erweiterungen und Varianten, beispielsweise durch Vorliegen

eines *Flow Shops* bestehend aus zwei *Batching*-Maschinen (siehe z. B. Manjeshwar et al. [64]), durch die Existenz von Auftrags-Familien (siehe z. B. Dobson und Nambimadom [31]) oder durch Einbeziehung von Rüstzeiten (siehe z. B. Dang und Kang [25]). Dabei können exakte Lösungsverfahren bereits bei der hier beschriebenen Problemstellung je nach Problemgröße keine optimalen Lösungen mehr in angemessener Zeit liefern. Folglich wird i. d. R. auf heuristische Lösungsverfahren zurückgegriffen.

4.4.2 Das *Resource-Constrained Project Scheduling Problem*

Das RCPSP wurde erstmals von Pritsker et al. [74] beschrieben. Obwohl es ursprünglich für den Bereich der ressourcenbeschränkten Projektplanung entwickelt wurde, ist es auch auf Maschinenbelegungsprobleme übertragbar.

Ein Projekt setzt sich beim RCPSP aus einer Menge von Ressourcen M (analog zur Maschinenmenge) und aus einer Menge von Aktivitäten J (analog zur Auftragsmenge) zusammen. Jede Ressource $i \in M$ stellt eine beschränkte Kapazität MK_i für die Durchführung der Aktivitäten zur Verfügung, wobei der Ressourcenbedarf einer Aktivität j bezüglich Ressource i mit kb_{ji} bezeichnet wird. Zudem ist die Prozessdauer (analog zur Bearbeitungsdauer) einer Aktivität j mit p_j gegeben. Des Weiteren unterstehen die einzelnen Aktivitäten beim RCPSP bestimmten Reihenfolge- bzw. Vorgänger-Nachfolger-Beziehungen ($\beta = prec$). Diese lassen sich in einer Menge VNB in Form von Aktivitätspaaren (j, k) zusammenfassen. Gilt $(j, k) \in VNB$, so impliziert dies dementsprechend, dass Aktivität k erst starten kann, wenn Aktivität j abgeschlossen ist. Als Ziel verfolgt das RCPSP standardmäßig die Minimierung der Projektdauer ($\gamma = C_{max}$).

Aus den Untersuchungen von Blazewicz et al. [10] geht hervor, dass das RCPSP *NP*-schwer ist. Pritsker et al. [74] schlagen eine zeitdiskrete Formulierung vor. Hierzu muss ein Planungshorizont der Form $H = \{0, 1, \dots, T^{max}\}$ definiert werden, wobei T^{max} die maximal angenommene Projektdauer und somit eine obere Schranke für den Zielfunktionswert darstellt. Zudem werden bei der Formulierung von Pritsker et al. [74] die zwei *Dummy*-Aktivitäten 0 und $n + 1$ benötigt,

wobei Aktivität 0 den Projektbeginn und Aktivität $n + 1$ das Projektende repräsentiert. Als Entscheidungsvariable dient:

$$x_{jt} = \begin{cases} 1, & \text{wenn Auftrag } j \text{ zum Zeitpunkt } t \text{ startet} \\ 0, & \text{sonst;} \end{cases}$$

Die Formulierung von Pritsker et al. [74] kann wie folgt dargestellt werden:

$$\min \sum_{t \in H} tx_{n+1,t} \quad (4.8)$$

u.d.N.

$$\sum_{t \in H} tx_{j't} \geq \sum_{t \in H} tx_{jt} + p_j \quad \forall (j, j') \in VNB \quad (4.9)$$

$$\sum_{j=1}^n kb_{ji} \sum_{\tau=t-p_i+1}^t x_{j\tau} \leq MK_i \quad \forall t \in H, i \in M \quad (4.10)$$

$$\sum_{t \in H} x_{jt} = 1 \quad \forall j \in J \cup \{0, n+1\} \quad (4.11)$$

$$x_{jt} \in \{0, 1\} \quad \forall j \in J \cup \{0, n+1\}, t \in H \quad (4.12)$$

Die Zielfunktion (4.8) minimiert den Startzeitpunkt der *Dummy*-Aktivität $n + 1$ und damit den Zeitpunkt des Projektendes. Mit Nebenbedingung (4.9) wird sichergestellt, dass die Reihenfolgebeziehungen zwischen den Aktivitäten eingehalten werden. Die Kapazitätsrestriktionen werden durch Nebenbedingung (4.10) abgedeckt. Schließlich wird mit den Nebenbedingungen (4.11) und (4.12) die Nichtunterbrechbarkeit der Aktivitäten gewährleistet.

Heute sind in der Literatur zahlreiche weitere Formulierungen für das RCPSP zu finden. Neben zeitdiskreten Modellen wurden in den letzten Jahren insbesondere auch zeitkontinuierliche Formulierungen entwickelt. Koné et al. [58] stellen verschiedene Formulierungsmethoden für das RCPSP vor und zeigen deren Vor- und Nachteile auf. Für einen umfassenden Überblick zum RCPSP sowie zu zahlreichen Varianten und Erweiterungen sei auf Hartmann und Briskorn [45] sowie auf Artigues [2] verwiesen.

Wie bereits erwähnt, lässt sich das RCPSP leicht auf Maschinenbelegungsprobleme übertragen. Inwiefern sich die Betriebsart der Maschine dann von der einer *Batching*-Maschine unterscheidet, soll im Folgenden anhand eines

Beispiels aufgezeigt werden. Es besteht aus 8 Aufträgen ($n = 8$) und einer Maschine ($m = 1$), welche eine Kapazität MK bereitstellt. Jeder Auftrag j verfügt über eine Bearbeitungszeit p_j und einen Kapazitätsbedarf kb_j . Tabelle 4.1 beinhaltet die relevanten Daten.

Tabelle 4.1: Beispieldaten für ein Maschinenbelegungsproblem mit Kapazitätsbeschränkung

| Auftrag j | Bearbeitungsdauer p_j | Kapazitätsbedarf kb_j |
|--|-------------------------|-------------------------|
| 1 | 5 | 2 |
| 2 | 6 | 6 |
| 3 | 12 | 8 |
| 4 | 5 | 1 |
| 5 | 6 | 2 |
| 6 | 4 | 3 |
| 7 | 8 | 2 |
| 8 | 8 | 3 |
| Maschinenkapazität MK: | | 10 |

Für den Fall einer *Batching*-Maschine ergibt sich die in Abbildung 4.5 veranschaulichte optimale Lösung. Die Abszisse bildet einen Zeitstrahl bzw. den Planungshorizont ab. Die jeweils dem Zeitpunkt t entsprechende Kapazitätsbeanspruchung $MK(t)$ ist auf der Ordinate abgetragen. Es wurden drei *Batches* gebildet. Der erste *Batch* beinhaltet die Aufträge 1, 4, 5 und 6. Seine Bearbeitung startet zum Zeitpunkt $t = 0$ und endet zum Zeitpunkt $t = 6$, da

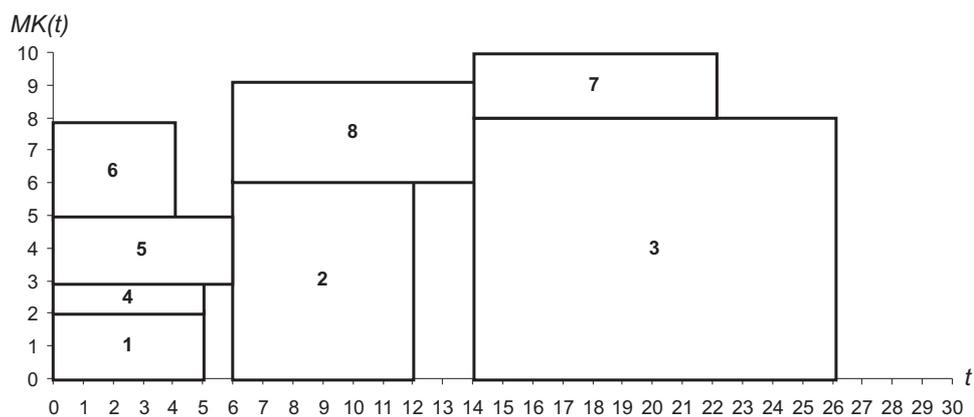


Abbildung 4.5: Lösung des Beispiels bei Modellierung als *Batching*-Problem

$p^1 = \max\{p_1, p_4, p_5, p_6\} = 6$. Dementsprechend sind die Aufträge 2 und 8 in *Batch 2* und die Aufträge 3 und 7 in *Batch 3* zusammengefasst. Die Bearbeitung von *Batch 2* reicht von $t = 6$ bis $t = 14$, die von *Batch 3* von $t = 14$ bis $t = 26$. Folglich ist die gesamte Fertigstellung nach 26 Zeiteinheiten abgeschlossen.

Wird das obige Beispiel nicht als *Batching*-Problem, sondern als RCPSP formuliert, ergibt sich die in Abbildung 4.6 dargestellte Lösung. Die Gesamtfertigstellung ist hier schon zum Zeitpunkt $t = 23$ abgeschlossen. Der Unterschied gegenüber der Lösung des *Batching*-Problems ist darauf zurückzuführen, dass die Aufträge beim RCPSP die Maschine direkt nach ihrer Fertigstellung verlassen können und somit früher Kapazität freisetzen. Es sei jedoch darauf hingewiesen, dass hier zur besseren Vergleichbarkeit keine Reihenfolgebeziehungen zwischen den Aufträgen betrachtet wurden. Es handelt sich hierbei somit um eine vereinfachte Variante des RCPSP.

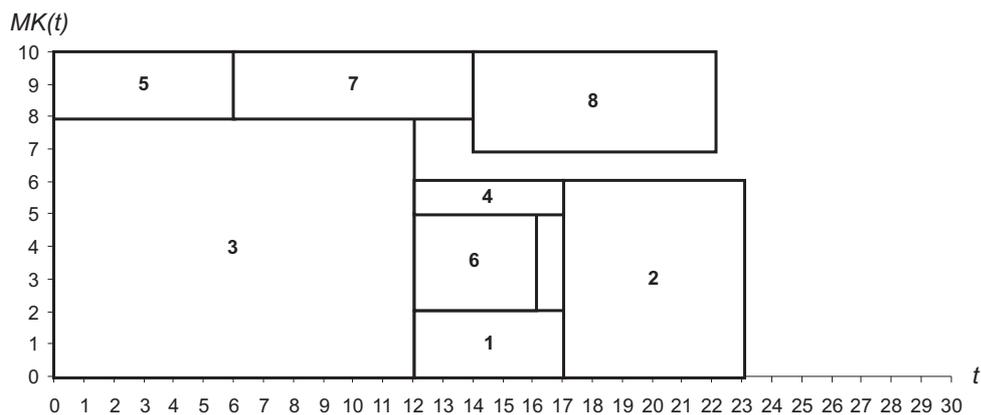


Abbildung 4.6: Lösung des Beispiels bei Modellierung als RCPSP

Der Vergleich der beiden Formulierungen zeigt, welche unterschiedlichen Auswirkungen sich für die Betriebsart einer Maschine ergeben. Allgemein lässt sich beobachten, dass der *Batching*-Modus im Vergleich zum RCPSP-Modus eine weniger kontinuierliche Abarbeitung der Aufträge zur Folge hat. Diesem Aspekt kommt im weiteren Verlauf dieser Arbeit noch eine gesonderte Bedeutung zu.

Kapitel 5

Beschreibung, Modellierung und Analyse eines realen Ofenbelegungsproblems

Dieses Kapitel dient der ausführlichen Beschreibung eines konkreten Ofenbelegungsproblems, dem sich die Dillinger Hüttenwerke bezüglich der Wärmebehandlung konfrontiert sehen. Unterkapitel 5.1 widmet sich zunächst dem relevanten Ofen. Neben wichtigen technischen Eckdaten werden hier insbesondere der Fertigungsablauf und die Ofensteuerung beschrieben, wobei der Fokus auf den jeweiligen logistischen Aspekten liegt. Im Anschluss befasst sich Unterkapitel 5.2 mit der mathematischen Formulierung der Problemstellung, ehe in Unterkapitel 5.3 ein intuitives Lösungsverfahren vorgestellt wird, welches zugleich dem aktuell im Realbetrieb angewandten Verfahren entspricht. Abschließend widmet sich Kapitel 5.4 der Analyse des Optimierungspotentials. In diesem Zusammenhang werden eine eigens entwickelte Materialflusssimulation des Ofens sowie ein spezielles *Branch&Bound*-Verfahren präsentiert.

5.1 Der Rollenherdofen RHO3

Wie aus Tabelle 3.1 hervorgeht, stehen den Dillinger Hüttenwerken neun verschiedene Wärmebehandlungsöfen zur Verfügung. Auf Vor- und Nachteile der jeweiligen Transportsysteme und Beschickungsarten wurde in Kapitel 3.4 bereits hingewiesen. Dabei haben sich die Rollenherdöfen als flexibelste Ofenart hinsichtlich der Ofenfahrweise herauskristallisiert. Aus diesem Grund wird an diesen Öfen das größte Optimierungspotential bezüglich der Ofenbelegung vermutet. Da der RHO3 der neueste und längste Ofen unter den drei Rollenherdöfen ist und zudem als einziger über einen Einzelrollenantrieb verfügt, wurde dieser als Optimierungsobjekt ausgewählt.

5.1.1 Daten und Eigenschaften

Der Rollenherdofen RHO3 wurde im Jahr 2009 in Betrieb genommen. Abbildung 5.1 zeigt eine Aufnahme aus dem Dillinger Walzwerk. Es handelt sich um



Abbildung 5.1: Der Rollenherdofen RHO3 im Walzwerk der Dillinger Hüttenwerke
(Quelle: AG der Dillinger Hüttenwerke)

eine Schrägsicht auf den Ofen und einen Teil des relevanten Lagerbereichs. Die Rollen, die auf der linken Seite zu sehen sind, bilden den Auslaufrollgang. Demnach handelt es sich bei der zu erkennenden Ofenöffnung etwas weiter rechts um die Ofenausgangstür. Nach abgeschlossener Wärmebehandlung verlassen die Waben den Ofen durch diese Tür und werden über den Auslaufrollgang abtransportiert. In den Ofen gelangen die Waben über einen entsprechenden Zulaufrollgang auf der anderen Ofenseite, die auf der Aufnahme jedoch nur schwer zu erkennen ist (siehe rechte Seite).

Die Abmessungen des Ofens und des Zulauf- bzw. Auslaufrollgangs sowie die Formate der relevanten Waben sind aus Tabelle 5.1 zu entnehmen. Aus einem Vergleich der Längswerte lässt sich direkt folgern, dass der Ofen prinzipiell mehrere Waben gleichzeitig bearbeiten kann. Zudem ist bei einem Vergleich der Breitenwerte festzustellen, dass der Ofen ausreichend Platz bereitstellt, um schmale Waben nebeneinander anzuordnen. Auf den letztgenannten Aspekt, der Möglichkeit der sogenannten Verband-Fahrweise, soll jedoch erst in Kapitel 6.1 eingegangen werden. Bis dahin wird von dieser Fahrweise abgesehen.

Tabelle 5.1: Anlagen- und Produktgeometrie des RHO3

| Ofen | |
|-------------------------------------|-------------------|
| Länge | 80.000 mm |
| Breite | 4.940 mm |
| Höhe (gesamt) | 2.600 mm |
| • von Ofenboden bis Rollenmitte | 1.050 mm |
| • von Rollenmitte bis Ofendecke | 1.550 mm |
| Zulauf- bzw. Auslaufrollgang | |
| Länge | 30.000 mm |
| Breite | 4.500 mm |
| Waben | |
| Länge | 2.000 - 20.000 mm |
| Breite | 1.000 - 4.500 mm |
| Dicke | 40 - 250 mm |
| Gewicht | 0 - 45000 kg |

Weitere technische Daten und Eigenschaften des RHO3 sind in Tabelle 5.2 zusammengefasst. Besonders hervorzuheben ist hier die Existenz der 17 Tempera-

turregelzonen. Durch diese ist es möglich, im Ofen gleichzeitig unterschiedliche Temperaturen einzustellen. Hierauf wird in Unterkapitel 5.1.2 genauer eingegangen.

Tabelle 5.2: Technische Daten und Eigenschaften des RHO3

| Heizsystem | |
|-----------------------------|---------------------|
| Brennmedium | Mischgas |
| Anzahl Brenner | 234 |
| Temperaturregelung | Zonen |
| Anzahl Temperaturregelzonen | 17 |
| Produkttemperaturbereich | 200 - 750 °C |
| Transportsystem | |
| Antriebsart | Einzelrollenantrieb |
| Anzahl Rollen | 122 |
| Rollganggeschwindigkeit | 0,15 - 15 m/min |
| Pendelgeschwindigkeit | 2 m/min |
| Pendelstrecke | 1.700 mm |

Bezüglich des Transportsystems wurde bereits in Kapitel 3.4 auf den Einzelrollenantrieb hingewiesen. Beim Ein- bzw. Ausfahren einer oder mehrerer Waben beträgt die Rollganggeschwindigkeit i. d. R. 10 m/min. Die meiste Zeit während ihrer Wärmebehandlung verbringen die Waben jedoch im Pendelbetrieb. Hiermit kann eine zu hohe Wärmeübertragung durch die Waben auf die Rollen verhindert werden. Dies begünstigt zum einen das Vermeiden von Eindrücken an den Stellen, wo die Waben auf den Rollen aufliegen. Zum anderen wird so einer Beschädigung der Rollen entgegengewirkt.

5.1.2 Fertigungsablauf und Ofensteuerung

Der Verlauf der Waben durch den Ofen lässt sich anhand von Abbildung 5.2 erläutern. Zunächst legt ein Kran die als nächstes benötigte Wabe auf dem Zulaufrollgang ab. Sofern genügend freier Rollgang vor dem Ofeneingang zur Verfügung steht, wird eine Freigabe zur Längenmessung erteilt. Ist dies der Fall, fährt die Wabe durch die Strecke der Längenmessung bis zur Warteposition vor dem Ofeneingang, wo sie auf das Einfahrtsignal wartet.

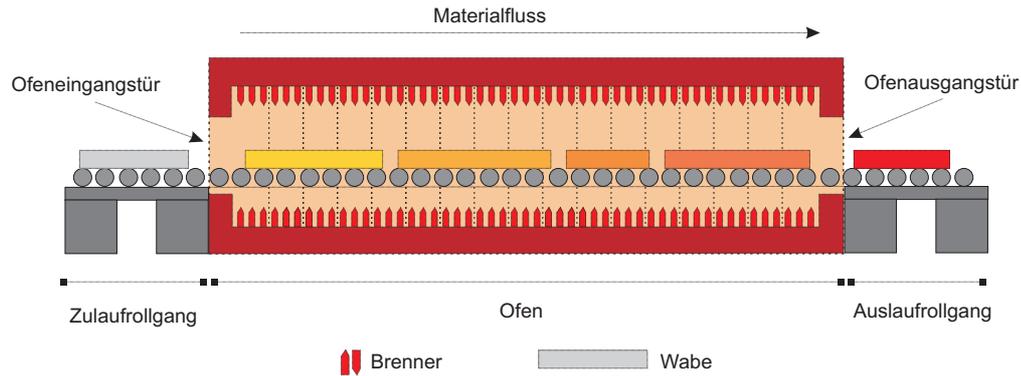


Abbildung 5.2: Eine schematische Darstellung des RHO3

Ist das Einfahrtsignal erteilt, wird zunächst die Ofeneingangstür geöffnet bzw. angehoben. Anschließend fährt die Wabe mit einer Geschwindigkeit von ca. 10 - 15 m/min (Eilfahrt) in den Ofen ein. Sobald das Wabenende (Wabenfuß) sicher in den Ofen eingefahren ist, wird die Ofeneingangstür wieder geschlossen. Die Wabe fährt solange in Eilfahrt weiter bis sie den Mindestabstand zur vorhergehenden Wabe, bzw. falls keine Vorgängergabe mehr im Ofen ist, bis sie den Mindestabstand zur Ofenausgangstür erreicht hat. Sobald die Wabe ihre Endposition eingenommen hat, schaltet der Ofen wieder in den Pendelbetrieb und behält diesen solange bei, bis entweder eine neue Wabe eingefahren oder die Wabe vor der Ofenausgangstür ausgefahren werden soll.

Hat die Wabe vor der Ofenausgangstür ihre Wärmebehandlungstemperatur und Haltedauer erreicht, wird zunächst die Ofenausgangstür geöffnet. Die Wabe fährt dann erneut in Eilfahrt aus dem Ofen. Sobald sie komplett ausgefahren ist, wird die Ofenausgangstür wieder geschlossen. Aufgrund der durch die Ausfahrt freigesetzten Kapazität rücken die nachfolgenden Waben im Ofen alle in Richtung Ofenausgangstür vor. Die ausgefahrene Wabe gelangt schließlich in den Abkühlbereich des Auslaufrollgangs, wo sie zur weiteren Luftabkühlung pendelt, bis sie wiederum per Kran abgehoben werden kann.

Entscheidend für die spätere Optimierung sind die Gründe für das Warten einer Wabe vor dem Ofeneingang. Grundsätzlich können zwei Waben nie (von der Verband-Fahrweise abgesehen) gleichzeitig in den Ofen einfahren, da eine Wabe immer mindestens so lange warten muss, bis ihre Vorgängerwabe komplett in den Ofen eingefahren ist. Der gleiche Sachverhalt gilt übrigens auch bei der Ofenausfahrt. Eine über diese grundsätzliche Mindestwartezeit hinausgehende

Wartezeit ist meist den folgenden Ursachen geschuldet:

- das Warten aufgrund nicht ausreichend freier Ofenkapazität,
- das Warten aufgrund unterschiedlicher Ofenverweilzeiten,
- das Warten aufgrund unterschiedlicher Ofentemperaturen.

Die drei Punkte werden im Folgenden näher erläutert, wobei für den letzteren zunächst auf die Temperaturregelung des Ofens und die damit verbundene Existenz von Temperaturlücken eingegangen werden muss.

Wartezeit aufgrund nicht ausreichend freier Ofenkapazität

Die Tatsache, dass einer vor der Ofeneingangstür wartenden Wabe das Einfahrtsignal zunächst verwehrt bleibt, kann der beschränkten Ofenkapazität zugrunde liegen. Abbildung 5.3 veranschaulicht eine derartige Situation. Sie ist als Sicht von oben auf bzw. in den Ofen zu verstehen. Der Ofenraum ist orange hervorgehoben.

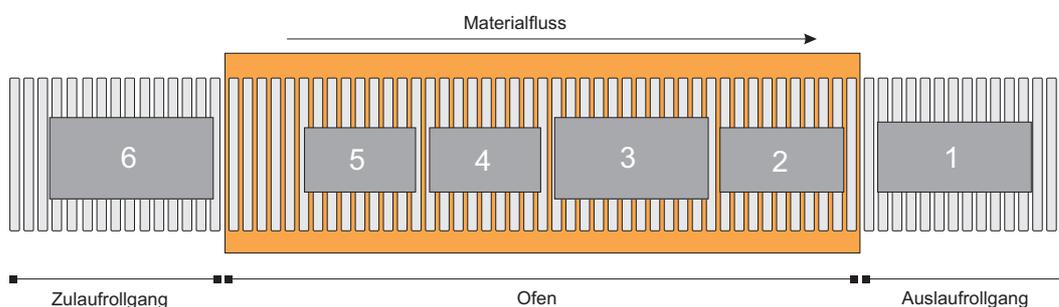


Abbildung 5.3: Beispiel für Wartezeit durch nicht ausreichend freie Ofenkapazität

Wabe 1 ist bereits aus dem Ofen ausgefahren und befindet sich nun auf dem Auslaufrollgang. Die Waben 2, 3, 4 und 5 liegen aktuell im Ofen und Wabe 6 vor dem Ofeneingang. Da der Ofen in der aktuellen Situation nicht genügend freie Kapazität für Wabe 6 bereitstellt, muss diese zunächst warten. Sobald die Wärmebehandlung von Wabe 2 abgeschlossen ist und diese den Ofen verlassen hat, können die Waben 3, 4 und 5 nachrücken und dadurch genügend Platz für Wabe 6 schaffen. In dieser Situation richtet sich die Wartezeit für Wabe 6 somit nach der Ausfahrzeit von Wabe 2. Wäre Wabe 2 jedoch so klein bzw.

Wabe 6 so groß, dass nach der Ausfahrt von Wabe 2 noch nicht ausreichend freie Kapazität bestehen würde, so müsste Wabe 6 sogar zumindest solange warten, bis Wabe 3 den Ofen verlässt.

Wartezeit aufgrund unterschiedlicher Ofenverweilzeiten

Eine weitere Ursache für eine über die grundsätzliche Mindestwartezeit hinausgehende Wartezeit ist den speziellen Wärmebehandlungsvorgaben sowie der Ofenbauweise geschuldet. In Kapitel 3.2 wurden die Wärmebehandlungsphasen Erwärmen, Halten und Abkühlen bereits erläutert. Bezogen auf einen Wärmebehandlungs-ofen lässt sich daraus eine Ofenverweilzeit ableiten, die sich aus der Summe von Erwärmdauer und Haltedauer zusammensetzt. Da es für eine erfolgreiche Wärmebehandlung erforderlich ist, die einzelnen Vorgabezeiten (und -temperaturen) genau einzuhalten, gilt dies somit auch für die Ofenverweilzeit. Diese Forderung kann in Verbindung mit der Bauart des Ofens bei unterschiedlichen Ofenverweilzeiten zweier aufeinanderfolgender Waben zu zusätzlichen Wartezeiten führen. Eine derartige Situation ist in Abbildung 5.4 veranschaulicht.

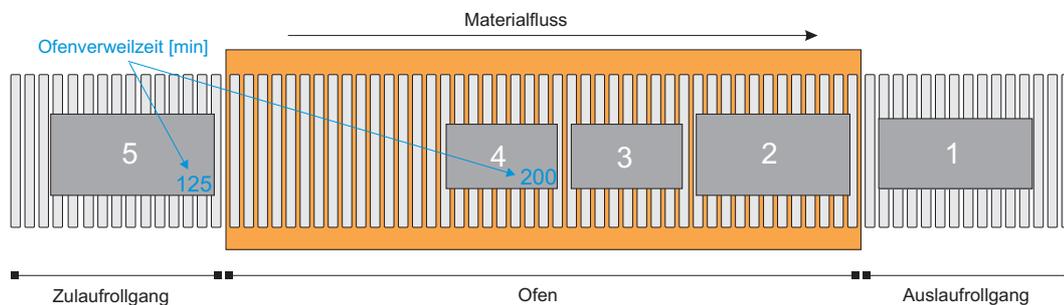


Abbildung 5.4: Beispiel für Wartezeit durch abweichende Ofenverweilzeiten

Zuletzt in den Ofen eingefahren ist Wabe 4. Ihr soll Wabe 5 folgen, welche noch auf das Einfahrtsignal wartet. Wabe 4 hat eine Ofenverweilzeit von 200 min und Wabe 5 von 125 min. Würde Wabe 5 direkt nach Wabe 4 einfahren, so wäre ihre Wärmebehandlung früher fertig als die von Wabe 4. Wabe 5 kann jedoch nicht vor Wabe 4 aus dem Ofen ausfahren, da Überholvorgänge im Ofen nicht realisierbar sind. Sie darf aber auch nicht über ihre Ofenverweilzeit hinaus im Ofen bleiben. Dieses Problem lässt sich nur vermeiden, indem Wabe

5 nach der Einfahrt von Wabe 4 eine bestimmte Zeit lang (in diesem Fall mindestens 75 min) vor dem Ofen wartet. Wenn sie anschließend einfährt, ist ihre Wärmebehandlung kurze Zeit nach der von Wabe 4 abgeschlossen und die direkte Ausfahrtmöglichkeit damit gewährleistet.

Die soeben beschriebene Wartezeit aufgrund unterschiedlicher Ofenverweilzeiten tritt nur auf, wenn eine Wabe auf eine Wabe mit einer höheren Ofenverweilzeit folgt. Der umgekehrte Fall stellt kein Problem dar.

Wartezeit aufgrund unterschiedlicher Ofentemperaturen

Wie bereits erwähnt, können durch die 17 Temperaturregelzonen im Ofen unterschiedliche Temperaturen gleichzeitig eingestellt werden. Dies ist allerdings nur bedingt möglich. So sind große Temperatursprünge von beispielsweise $50\text{ }^{\circ}\text{C}$ zwischen zwei Zonen i. d. R. nicht realisierbar, da hier die wechselseitigen Strahlungseinflüsse zu hoch wären. In diesen Fällen muss die Temperaturdifferenz über mehrere Zonen verteilt werden. Abbildung 5.5 zeigt ein Beispiel. Wabe 1 erfordert eine Ofentemperatur von $620\text{ }^{\circ}\text{C}$. Für die darauf folgende Wabe 2 wird dagegen nur eine Temperatur von $600\text{ }^{\circ}\text{C}$ benötigt. Die gestrichelten Linien kennzeichnen die jeweiligen Temperaturregelzonen mit den in Rot dargestellten Sollwerten.

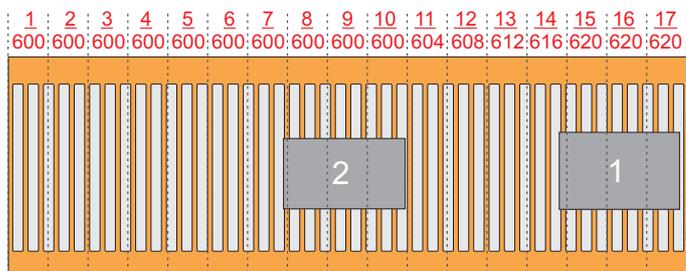


Abbildung 5.5: Temperaturregelung

In diesem Beispiel wird die Temperaturdifferenz zwischen Wabe 1 und Wabe 2 von $20\text{ }^{\circ}\text{C}$ über 3 Zonen verteilt. Der Abstand (bzw. die Temperaturlücke) zwischen den beiden Waben beträgt hier ca. 20 m. Mit dieser Ofenfahr- bzw. Ofenregelweise wird versucht, die wechselseitigen Strahlungseinflüsse möglichst gering zu halten, sodass bei Wabe 1 nahezu $620\text{ }^{\circ}\text{C}$ herrschen und bei Wabe 2 letztendlich nur $600\text{ }^{\circ}\text{C}$. Durch die Existenz von derartigen Temperaturlücken

können wiederum weitere Wartezeiten entstehen, wie Abbildung 5.6 illustriert.

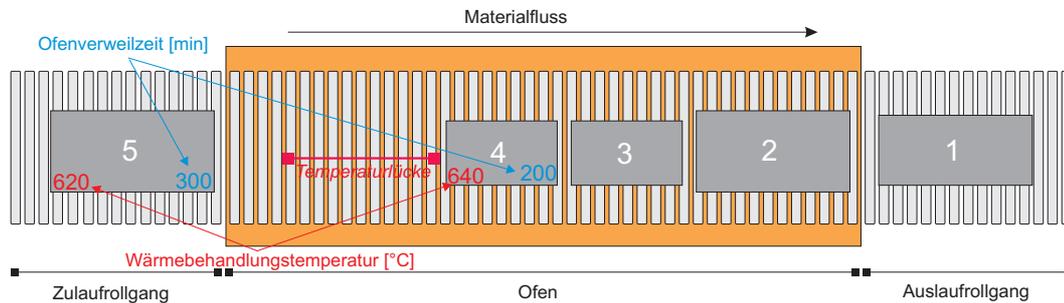


Abbildung 5.6: Beispiel für Wartezeit aufgrund einer Temperaturlücke

Wie man sieht, wartet Wabe 5 vor dem Ofen. Die zuletzt eingefahrene Wabe 4 hat eine Ofenverweilzeit von 200 min und benötigt eine Ofentemperatur von 640 °C. Wabe 5 hat dagegen eine Ofenverweilzeit von 300 min und bedarf einer Ofentemperatur von 620 °C. Da die Ofenverweilzeit von Wabe 5 größer ist als die von Wabe 4 und im Ofen eigentlich genug freie Kapazität zur Verfügung steht, könnte Wabe 5 prinzipiell schon einfahren. Aufgrund der unterschiedlichen Wärmebehandlungstemperaturen muss allerdings eine Temperaturlücke zwischen den beiden Waben eingehalten werden, wodurch sich die freie Ofenkapazität verringert und diese infolgedessen für die Einfahrt von Wabe 5 nicht mehr ausreichend ist. In diesem Fall kann Wabe 5 erst in den Ofen einfahren, wenn Wabe 2 diesen verlassen hat.

5.2 Mathematische Problemformulierung

Ein Maschinenbelegungsproblem wie es sich in Zusammenhang mit der Reihenfolgebildung am RHO3 darstellt fand in der Literatur bislang keine Betrachtung. Ähnliche Problemstellungen wurden beispielsweise von Tang und Zhao [89] oder von Wang und Tang [98] behandelt, jedoch liegen bei keiner dieser beiden Arbeiten Ressourcenrestriktionen vor, die den in Unterkapitel 5.1.2 erläuterten Gegebenheiten entsprechen. Insbesondere die Existenz bzw. die Betrachtung von Temperaturlücken scheint einen Spezialfall darzustellen. Aus diesem Grund wird die hier betrachtete Problemstellung im Folgenden als *Heat*

Treatment Furnace Scheduling Problem with Distance Constraints (HTFSPD) bezeichnet. Das HTFSPD fand im Rahmen der Erstellung der vorliegenden Arbeit erstmals Anfang des Jahres 2012 Einzug in die Literatur (siehe Ebermann et al. [33]).

Das HTFSPD lässt sich als gemischt-ganzzahliges Programm formulieren. Im Folgenden wird hierzu sowohl eine zeitdiskrete als auch eine ereignisbasierte Modellierung vorgestellt. Zunächst werden die erforderlichen Eingangsdaten bzw. Parameter wie folgt definiert:

| | |
|-------------------|--|
| n : | Anzahl Waben, |
| l^{Ofen} : | Ofenlänge, |
| l_j : | Länge von Wabe j , |
| d_j : | Dicke von Wabe j , |
| ewd_j : | Erwärmdauer von Wabe j , |
| hd_j : | Haltedauer von Wabe j , |
| ovz_j : | Ofenverweilzeit von Wabe j , |
| $wbht_j$: | Wärmebehandlungstemperatur von Wabe j , |
| $wbht_j^{Ofen}$: | Ofentemperatur von Wabe j , |
| $wbht_j^{Ziel}$: | Zieltemperatur von Wabe j , |
| $temptol_j^+$: | Obere Temperaturtoleranz von Wabe j , |
| $temptol_j^-$: | Untere Temperaturtoleranz von Wabe j , |
| t^{min} : | Mindestwartezeit zur Vorgängerwabe bzw. Einfahrdauer, |
| $templ_j$: | Temperaturlücke zwischen Wabe j und ihrer Vorgängerwabe, |
| $templ^{min}$: | Mindestabstand zwischen zwei Waben im Ofen, |
| f_{templ} : | Temperaturlückenfaktor. |

Wie man sieht, wird zwischen der Erwärmdauer ewd , der Haltedauer hd und der Ofenverweilzeit ovz unterschieden, wobei diese in folgendem Zusammenhang zueinander stehen:

$$ovz_j = ewd_j + hd_j, \forall j \in J. \quad (5.1)$$

Die Haltedauer hd_j wird ebenso wie die Wärmebehandlungstemperatur $wbht_j$ von der metallurgischen Abteilung der Dillinger Hüttenwerke vorgegeben. Die Wärmebehandlungstemperatur $wbht_j$ lässt sich wiederum von der Ofentempe-

ratur $wbht_j^{Ofen}$ und der Zieltemperatur $wbht_j^{Ziel}$ differenzieren. Hier gilt:

$$wbht_j^{Ofen} = wbht_j + temptol_j^+, \forall j \in J, \quad (5.2)$$

$$wbht_j^{Ziel} = wbht_j - temptol_j^-, \forall j \in J. \quad (5.3)$$

Der Ofen wird grundsätzlich auf die Temperatur $wbht_j^{Ofen}$ geregelt. Sobald eine Wabe ihre Zieltemperatur $wbht_j^{Ziel}$ erreicht hat, geht sie von der Erwärm- in die Haltephase über. Die Erwärmdauer ewd_j ist dabei ofenabhängig und wird bei den Dillinger Hüttenwerken näherungsweise als eine Funktion der Ofentemperatur $wbht_j^{Ofen}$, der Zieltemperatur $wbht_j^{Ziel}$ und der Wabendicke d_j berechnet.

Für die Mindestwartezeit t^{min} , d. h. für die Zeit, die eine Wabe ab dem Zeitpunkt der Ofeneinfahrt ihrer Vorgängerwabe mindestens warten muss, wird ein fixer Wert angenommen. Dieser orientiert sich an der Dauer, die eine Wabe in etwa benötigt, um nach Erteilung des Einfahrtsignals komplett in den Ofen einzufahren (Einfahrdauer). Denn erst wenn die Vorgängerwabe ihre Position im Ofen eingenommen hat, kann die Einfahrt der nachfolgenden Wabe frühestens beginnen. Im Folgenden wird eine Einfahrdauer von 10 min angenommen ($t^{min} = 10$).

Für die Temperaturlücken gibt es noch keine metallurgischen Vorgaben. Es liegt jedoch auf der Hand, dass größere Ofentemperaturdifferenzen zwischen zwei Waben auch größere Temperaturlücken erfordern. Die Ofensteuerer haben daher eine einfache Daumenregel entwickelt, nach der gilt:

$$templ_j = \max\{(|wbht_j^{Ofen} - wbht_{j'}^{Ofen}| \cdot f_{templ}), templ^{min}\}, \forall j, j' \in J, \quad (5.4)$$

wobei j' die Vorgängerwabe von Wabe j kennzeichnet und f_{templ} einen Temperaturlückenfaktor darstellt, der in der Praxis je nach Steuermann zwischen 600 und $1000 \frac{\text{mm}}{^\circ\text{C}}$ liegt. Die Maximum-Funktion wird verwendet, da zwischen zwei Waben grundsätzlich ein Mindestabstand $templ^{min}$ bestehen muss (auch bei gleicher Ofentemperatur). Dieser ist zwingend erforderlich, damit im Ofen benachbarte Waben während des Pendelbetriebs nicht gegeneinander anstoßen. Auch $templ^{min}$ variiert je nach Steuermann und liegt i. d. R. zwischen 1500 und 3000 mm. Da die Temperaturlücken einen entscheidenden Einfluss auf den Ofendurchsatz ausüben, wird bereits an den von den Ofensteuerern ab-

hängigen Schwankungen ersichtlich, dass es hier einer einheitlichen Berechnungsmethode bedarf. Diesem Ziel wird später in Kapitel 7.4.2 nachgegangen.

5.2.1 Ein zeitdiskretes Modell

Das HTFSPD hat, bezogen auf die beiden in Kapitel 4.4 vorgestellten Problemstellungen hinsichtlich ressourcenbeschränkter Maschinenbelegungsprobleme, mit dem RCPSP die meisten Gemeinsamkeiten. Vernachlässigt man die unterschiedlichen Gründe für Wartezeiten, so lässt sich das HTFSPD sogar als RCPSP (ohne Reihenfolgebeziehungen) beschreiben. Aus diesem Grund lehnt sich die folgende zeitdiskrete Formulierung (ZDF) für das HTFSPD an die RCPSP-Formulierung von Pritsker et al. [74] an.

Da das HTFSPD durch die Existenz von Temperaturlücken eine Art reihenfolgeabhängiger Rüstkapazitäten aufweist, müssen die Entscheidungsvariablen bestimmte Vorgänger-Nachfolger-Beziehungen abbilden. So muss z. B. für eine einfahrende Wabe der Bezug zur direkten Vorgängerwabe hergestellt werden, wohingegen bei einer ausfahrenden Wabe der Bezug zur direkten Nachfolgerwabe erforderlich ist. Letzteres ist bedingt durch die Tatsache, dass eine ausfahrende Wabe immer auch die bestehende Temperaturlücke zu ihrer direkten Nachfolgerwabe „mitnimmt“ bzw. die Lücke sich durch die Ausfahrt auflöst. Diese sowie weitere Eigenschaften des HTFSPD erfordern eine Erweiterung des Modells von Pritsker et al. [74], sowohl bezüglich der Entscheidungsvariablen als auch hinsichtlich der Nebenbedingungen.

Es werden die folgenden Entscheidungsvariablen für das (ZDF)-Modell definiert:

$$\begin{aligned}
 x_{jj't} &= \begin{cases} 1, & \text{wenn Wabe } j \text{ zum Zeitpunkt } t \text{ in den Ofen einfährt und} \\ & \text{Wabe } j' \text{ als direkten Vorgänger hat} \\ 0, & \text{sonst;} \end{cases} \\
 y_{jj't} &= \begin{cases} 1, & \text{wenn Wabe } j \text{ zum Zeitpunkt } t \text{ aus dem Ofen ausfährt und} \\ & \text{Wabe } j' \text{ als direkten Nachfolger hat} \\ 0, & \text{sonst;} \end{cases} \\
 c_t &: \text{ Freie Ofenkapazität zum Zeitpunkt } t
 \end{aligned}$$

Das Modell selbst hat folgendes Aussehen:

$$(ZDF) \quad \min \quad \sum_{j=1}^n \sum_{t=0}^{T^{max}} y_{jn+1t} \cdot t \quad (5.5)$$

u.d.N.

$$\sum_{j'=1}^n \sum_{t=0}^{T^{max}} x_{n+1j't} \cdot t \geq \sum_{j'=0}^n \sum_{t=0}^{T^{max}} y_{jj't} \cdot t \quad \forall j = 1, \dots, n \quad (5.6)$$

$$\sum_{j'=0}^{n+1} \sum_{t=0}^{T^{max}} x_{jj't} = 1 \quad \forall j = 0, \dots, n+1 \quad (5.7)$$

$$\sum_{j'=0}^{n+1} \sum_{t=0}^{T^{max}} y_{jj't} = 1 \quad \forall j = 0, \dots, n+1 \quad (5.8)$$

$$\sum_{j=0}^{n+1} \sum_{t=0}^{T^{max}} x_{jj't} = 1 \quad \forall j' = 0, \dots, n+1 \quad (5.9)$$

$$\sum_{j=0}^{n+1} \sum_{t=0}^{T^{max}} y_{jj't} = 1 \quad \forall j' = 0, \dots, n+1 \quad (5.10)$$

$$\sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} x_{jj't} \leq 1 \quad \forall t = 1, \dots, T^{max} \quad (5.11)$$

$$\sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} y_{jj't} \leq 1 \quad \forall t = 0, \dots, T^{max} \quad (5.12)$$

$$\sum_{t=0}^{T^{max}} x_{jj't} + \sum_{t=0}^{T^{max}} y_{j'jt} \leq 1 \quad \forall j, j' = 0, \dots, n+1 \quad (5.13)$$

$$\sum_{j'=0}^{n+1} \sum_{t=1}^{T^{max}} y_{jj't} \cdot t = \sum_{j'=0}^{n+1} \sum_{t=1}^{T^{max}} x_{jj't} \cdot (t + ovz_j + t^{min}) \quad \forall j = 1, \dots, n \quad (5.14)$$

$$\sum_{j'=0}^n \sum_{t=0}^{T^{max}} x_{jj't} \cdot (t + t^{min}) \leq \sum_{j'=0}^{n+1} \sum_{t=0}^{T^{max}} x_{j'jt} \cdot t \quad \forall j = 1, \dots, n \quad (5.15)$$

$$\sum_{j'=1}^{n+1} \sum_{t=0}^{T^{max}} y_{jj't} \cdot t \geq \sum_{j'=0}^{n+1} \sum_{t=0}^{T^{max}} y_{j'jt} \cdot (t + t^{min}) \quad \forall j = 1, \dots, n \quad (5.16)$$

$$\sum_{t=0}^{T^{max}} y_{jj't} = \sum_{t=0}^{T^{max}} x_{j'jt} \quad \forall j, j' = 0, \dots, n+1 \quad (5.17)$$

$$x_{0n+10} = 1 \quad (5.18)$$

$$\sum_{t=0}^{T^{max}} y_{n+10t} = 1 \quad (5.19)$$

$$c_0 = l^{Ofen} - \sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} x_{jj'0} \cdot (l_j + templ_j) \quad (5.20)$$

$$c_t = c_{t-1} - \sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} x_{jj't} \cdot (l_j + templ_j) + \sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} y_{jj't} \cdot (l_j + templ_j) \quad \forall t = 1, \dots, T^{max} \quad (5.21)$$

$$x_{jj't} \in \{0, 1\} \quad \forall j, j' = 0, \dots, n+1 \quad t = 1, \dots, T^{max} \quad (5.22)$$

$$y_{jj't} \in \{0, 1\} \quad \forall j, j' = 0, \dots, n+1 \quad t = 1, \dots, T^{max} \quad (5.23)$$

$$c_t \geq 0 \quad \forall t = 1, \dots, T^{max} \quad (5.24)$$

Die Zielfunktion (5.5) minimiert den Ofenausfahrtzeitpunkt der Wabe, welche die *Dummy*-Wabe $n + 1$ zum Nachfolger hat. Dies impliziert die Minimierung der Gesamtfertigstellungszeit, da Nebenbedingung (5.6) fordert, dass *Dummy*-Wabe $n+1$ erst „einfahren“ kann, wenn alle anderen Waben bereits fertiggestellt sind. Die Nebenbedingungen (5.7) - (5.10) gewährleisten, dass jede Wabe genau einmal in den Ofen ein- und ausfährt und dass jede Wabe genau einmal Vorgänger und einmal Nachfolger einer anderen Wabe sein muss. Man beachte in diesem Zusammenhang, dass *Dummy*-Wabe 0 grundsätzlich den „Nachfolger“ von *Dummy*-Wabe $n + 1$ bildet (siehe Nebenbedingungen (5.18) und (5.19)). Durch die Nebenbedingungen (5.11) und (5.12) wird zudem sichergestellt, dass zu keinem Zeitpunkt t mehrere Waben gleichzeitig ein- bzw. ausfahren können. Nebenbedingung (5.13) sorgt dafür, dass eine Wabe nicht zugleich Vorgänger und Nachfolger einer anderen Wabe sein kann. Es folgt mit Nebenbedingung (5.14) die Forderung, dass jede Wabe ihre Ofenverweilzeit (unter Berücksichtigung der Einfahrdauer) einhalten muss. Zudem stellen die Nebenbedingungen (5.15) und (5.16) sicher, dass Waben nicht vor ihrem Vorgänger bzw. nach ihrem Nachfolger ein- bzw. ausfahren. Dem ofenbedingten Erfordernis einer identischen Einfahr- und Ausfahrreihenfolge kommt Nebenbedingung (5.17) nach. Durch die Nebenbedingungen (5.18) und (5.19) wird zudem sichergestellt, dass *Dummy*-Wabe 0 als erste in den Ofen einfährt und *Dummy*-Wabe $n + 1$ diesen als letzte Wabe verlässt. Die Kapazitätsrestriktionen werden mit

den Nebenbedingungen (5.20) und (5.21) abgebildet. Abschließend geben die Nebenbedingungen (5.22) - (5.24) die Wertebereiche für die verwendeten Entscheidungsvariablen vor.

Für die Effizienz zeitdiskreter Modelle sind vor allem die Anzahl und Abstände der betrachteten Zeitpunkte sowie die Länge des Planungshorizonts entscheidend. Da letztere im Vorhinein vorgegeben werden muss, gilt es, zunächst eine obere Schranke für die Gesamtfertigstellungszeit zu berechnen. Zudem verfügen zeitdiskrete Modelle insbesondere bei langem Planungshorizont über eine sehr große Anzahl an Binärvariablen, was häufig hohe Rechenzeiten zur Folge hat.

5.2.2 Ein ereignisbasiertes Modell

Wie bereits in Kapitel 4.4.2 erwähnt, stellen Koné et al. [58] verschiedene Formulierungsmethoden, insbesondere auch zwei ereignisbasierte Formulierungen, für das RCPSP vor und untersuchen diese bezüglich ihrer Effizienz. Ihre Ergebnisse zeigen auf der einen Seite, dass keine Formulierung existiert, welche alle anderen Formulierungen dominiert. Dennoch scheinen andererseits Probleminstanzen, die eine weite Bandbreite bezüglich der Bearbeitungszeiten aufweisen, mit einer ereignisbasierten Formulierung effizienter zu lösen sein als mit einem zeitdiskreten Modell. Aus diesem Grund wird neben dem zeitdiskreten Modell (ZDF) im Folgenden auch eine ereignisbasierte Formulierung (EBF) für das HTFSPD vorgeschlagen.

Eine ereignisbasierte Modellierung erfordert zunächst die Definition der relevanten Ereignisse. Für das HTFSPD wird ein Ereignis e entweder durch die Ofeneinfahrt oder die Ofenausfahrt einer Wabe ausgelöst. Da auch das (EBF)-Modell die *Dummy*-Waben 0 und $n + 1$ verwendet, ergibt sich grundsätzlich eine Anzahl an Ereignissen von $E = 2 \cdot (n + 2)$. Da sich die Ofenein- und ausfahrt bei den *Dummy*-Waben jedoch überlagert, reichen auch $E = 2 \cdot n + 2$ Ereignisse aus.

Es werden die folgenden Entscheidungsvariablen definiert:

$$\begin{aligned}
x_{jj'e} &= \begin{cases} 1, & \text{wenn Wabe } j \text{ zum Ereignis } e \text{ in den Ofen einfährt und} \\ & \text{Wabe } j' \text{ als direkten Vorgänger hat} \\ 0, & \text{sonst;} \end{cases} \\
y_{jj'e} &= \begin{cases} 1, & \text{wenn Wabe } j \text{ zum Ereignis } e \text{ aus dem Ofen ausfährt und} \\ & \text{Wabe } j' \text{ als direkten Nachfolger hat} \\ 0, & \text{sonst;} \end{cases} \\
c_e &: \text{ Freie Ofenkapazität zum Ereignis } e \\
t_e &: \text{ Eintrittszeitpunkt von Ereignis } e
\end{aligned}$$

Das (EBF)-Modell stellt sich wie folgt dar:

$$(EBF) \quad \min \quad t_{E-1} \quad (5.25)$$

u.d.N.

$$t_0 = 0 \quad (5.26)$$

$$\begin{aligned}
t_f \geq t_e + \left(\sum_{j'=0}^{n+1} ovz_j + t^{min} \right) \cdot x_{jj'e} \\
- (ovz_j + t^{min}) \cdot \left(1 - \sum_{j'=0}^{n+1} y_{jj'f} \right) \quad \forall j = 1, \dots, n, \\
e, f = 0, \dots, E-1, e < f \quad (5.27)
\end{aligned}$$

$$t_{e+1} \geq t_e + t^{min} \quad \forall e = 0, \dots, E-2 \quad (5.28)$$

$$\sum_{j=1}^n \sum_{j'=0}^{n+1} x_{jj'e} \leq 1 \quad \forall e = 0, \dots, E-1 \quad (5.29)$$

$$\sum_{j=1}^n \sum_{j'=0}^{n+1} y_{jj'e} \leq 1 \quad \forall e = 0, \dots, E-1 \quad (5.30)$$

$$x_{0n+10} = 1 \quad (5.31)$$

$$\sum_{j'=0}^{n+1} \sum_{e=0}^{E-1} x_{jj'e} = 1 \quad \forall j = 0, \dots, n+1 \quad (5.32)$$

$$\sum_{j=0}^{n+1} \sum_{e=0}^{E-1} x_{jj'e} = 1 \quad \forall j' = 0, \dots, n+1 \quad (5.33)$$

$$y_{n+10E-1} = 1 \quad (5.34)$$

$$\sum_{j'=0}^{n+1} \sum_{e=0}^{E-1} y_{jj'e} = 1 \quad \forall j = 0, \dots, n+1 \quad (5.35)$$

$$\sum_{j=0}^{n+1} \sum_{e=0}^{E-1} y_{jj'e} = 1 \quad \forall j' = 0, \dots, n+1 \quad (5.36)$$

$$\sum_{j'=0}^{n+1} x_{jj'e} + \sum_{j'=0}^{n+1} y_{jj'e} \leq 1 \quad \forall j = 1, \dots, n, \\ e = 0, \dots, E-1 \quad (5.37)$$

$$x_{0n+10} + \sum_{j'=0}^{n+1} y_{0j'0} = 2 \quad (5.38)$$

$$\sum_{j'=0}^{n+1} x_{n+1j'E-1} + y_{n+10E-1} = 2 \quad (5.39)$$

$$\sum_{e=0}^{E-1} x_{jj'e} + \sum_{e=0}^{E-1} x_{j'je} \leq 1 \quad \forall j, j' = 0, \dots, n+1 \quad (5.40)$$

$$\sum_{e=0}^{E-1} y_{jj'e} + \sum_{e=0}^{E-1} y_{j'je} \leq 1 \quad \forall j, j' = 0, \dots, n+1 \quad (5.41)$$

$$\sum_{e'=e}^{E-1} \sum_{j'=0}^{n+1} x_{jj'e'} + \sum_{e''=0}^{e-1} \sum_{j'=0}^{n+1} y_{jj'e''} \leq 1 \quad \forall j = 0, \dots, n+1, \\ e = 0, \dots, E-1 \quad (5.42)$$

$$\sum_{e=0}^{E-1} y_{jj'e} \cdot e \leq \sum_{j''=0}^{n+1} \sum_{e=0}^{E-1} y_{j'j''e} \cdot e \quad \forall j = 0, \dots, n, \\ j' = 1, \dots, n+1 \quad (5.43)$$

$$\sum_{e=0}^{E-1} x_{jj'e} = \sum_{e=0}^{E-1} y_{j'je} \quad \forall j, j' = 0, \dots, n+1 \quad (5.44)$$

$$\sum_{e=0}^{E-1} x_{jj'e} \cdot e \geq \sum_{j''=0}^{n+1} \sum_{e=0}^{E-1} x_{j'j''e} \cdot e + \\ \left(\sum_{e=0}^{E-1} x_{jj'e} - 1 \right) \cdot G \quad \forall j = 1, \dots, n+1, \\ j' = 0, \dots, n+1 \quad (5.45)$$

$$c_0 = l^{Ofen} - \sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} x_{jj'0} \cdot (l_j + templ_j) \quad (5.46)$$

$$c_e = c_{e-1} - \sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} x_{jj'e} \cdot (l_j + templ_j) \\ + \sum_{j=0}^{n+1} \sum_{j'=0}^{n+1} y_{jj'e} \cdot (l_j + templ_j) \quad \forall e = 1, \dots, E-1 \quad (5.47)$$

$$x_{jj'e} \in \{0, 1\} \quad \forall j, j' = 0, \dots, n+1 \\ e = 1, \dots, E \quad (5.48)$$

$$y_{jj'e} \in \{0, 1\} \quad \forall j, j' = 0, \dots, n + 1 \\ e = 1, \dots, E \quad (5.49)$$

$$c_e \geq 0 \quad \forall e = 1, \dots, E \quad (5.50)$$

$$t_e \geq 0 \quad \forall e = 1, \dots, E \quad (5.51)$$

Die Zielfunktion (5.25) minimiert den Eintrittszeitpunkt des letzten Ereignisses und damit den Gesamtfertigstellungszeitpunkt. Die Nebenbedingung (5.26) sorgt dafür, dass Ereignis 0 zum Zeitpunkt 0 eintritt. Mit Nebenbedingung (5.27) wird die Einhaltung der Ofenverweilzeiten sichergestellt, wohingegen Nebenbedingung (5.28) fordert, dass zwei Ereignisse um mindestens t^{min} Zeiteinheiten versetzt zueinander eintreten müssen. Die Nebenbedingungen (5.29) und (5.30) gewährleisten, dass zu jedem Ereignis nur maximal eine Wabe ein- bzw. ausfahren darf. Die Ein- und Ausfahrtbedingungen für die beiden *Dummy*-Variablen werden mit den Nebenbedingungen (5.31) und (5.34) abgedeckt. Dem Erfordernis, dass jede Wabe zum einen genau einmal in den Ofen ein- bzw. ausfahren und zum anderen genau einmal Vorgänger bzw. Nachfolger einer anderen Wabe sein muss, kommen die Nebenbedingungen (5.32) und (5.33) sowie (5.35) und (5.36) nach. Durch Nebenbedingung (5.37) wird zudem sichergestellt, dass eine Wabe nicht zum gleichen Ereignis ein- und ausfährt, wobei dies nicht für die beiden *Dummy*-Variablen gilt, zumal die Nebenbedingungen (5.38) und (5.39) fordern, dass Wabe 0 zum Zeitpunkt 0 und Wabe $n + 1$ zum Zeitpunkt $E - 1$ ein- und ausfährt. Die weiteren Nebenbedingungen (5.40) - (5.45) gewährleisten Konformität bezüglich der Vorgänger-Nachfolger-Beziehungen und den entsprechenden Ein- und Ausfahrtzeiten der Waben, wobei G in Nebenbedingung (5.45) eine hinreichend große Zahl symbolisiert. Die nächsten beiden Nebenbedingungen (5.46) und (5.47) bilden die Kapazitätsrestriktionen ehe mit den Nebenbedingungen (5.48) - (5.51) schließlich die Wertebereiche der Entscheidungsvariablen festgelegt werden.

Im Gegensatz zu einer zeitdiskreten Modellierung, benötigt ein auf Ereignissen basierendes Modell keine obere Schranke für die Länge des Planungshorizonts bzw. den Zielfunktionswert. Inwiefern sich die beiden vorgestellten Formulierungen für das HTFSPD zum Lösen verschiedener Probleminstanzen eignen, wird u. a. in Kapitel 5.4 untersucht.

5.3 Ein intuitives Lösungsverfahren

Mit Hinblick auf die beschriebenen Gründe für Wartezeiten, könnte man intuitiv auf ein relativ einfaches Lösungsverfahren für das HTFSPD schließen. Die Idee dieses Verfahrens bzw. dessen Herleitung wird im Folgenden vorgestellt. Da es zudem bei den Dillinger Hüttenwerken im alltäglichen Betrieb verwendet wird, bildet es die Basis für eine Analyse des Optimierungspotentials, welcher sich Kapitel 5.4 widmet.

Für das HTFSPD gilt grundsätzlich, dass Temperaturlücken möglichst zu vermeiden sind. Daher sollten Waben unterschiedlicher Wärmebehandlungs- bzw. Ofentemperatur nicht permanent direkt hintereinander in den Ofen einfahren. Somit erscheint es sinnvoll, in einem ersten Schritt eine Gruppierung aller Waben, die identische Ofentemperaturen benötigen, vorzunehmen. Eine derartige Gruppe wird auch als *Temperaturblock* bezeichnet. Man erhält somit eine Menge verschiedener Temperaturblöcke. Da man Temperaturlücken möglichst klein halten möchte, erfolgt im nächsten Schritt eine Sortierung der Temperaturblöcke. Ob es sinnvoll ist, ab- oder aufsteigend zu sortieren, hängt vor allem von der aktuellen Ofenbelegung ab.

Eine weitere Ursache für Wartezeiten war der Fall, wenn eine Wabe mit einer niedrigeren Ofenverweilzeit auf eine Wabe mit einer höheren Ofenverweilzeit folgt. Aus diesem Grund erscheint eine aufsteigende Sortierung der Ofenverweilzeiten sinnvoll. Ausgehend von der vorherigen Sortierung nach den Ofentemperaturen, wird daher anschließend in jedem Temperaturblock noch einmal eine Sortierung der Ofenverweilzeiten vorgenommen.

Als weitere Sortierkriterien werden i. d. R. noch die Sollproduktionswochen der Waben oder deren Lagerplätze bzw. Lagen im betreffenden Stapel herangezogen. Diese beiden Aspekte sollen an dieser Stelle jedoch zunächst vernachlässigt und erst in Kapitel 6.3 betrachtet werden.

Algorithmus 1 stellt die Vorgehensweise der beschriebenen Lösungsmethode in Form eines Pseudocodes dar. Im weiteren Verlauf dieser Arbeit wird das Verfahren als *Rule-based Sorting* (RBS) bezeichnet. Als Ergebnis liefert der Algorithmus die geordnete Menge *folge*, welche die konstruierte Reihenfolge

Algorithmus 1 *Rule-based Sorting*

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , mit $j \in J$, f_{templ} , $templ^{min}$, t^{min} ,
 $sort_{auf}$, $sort_{ab}$

Ensure: Reihenfolge $folge$

```

1: initialisiere  $folge$ , sodass  $folge = (1, 2, \dots, n)$ 
2: for  $p = 1, \dots, n$  do
3:   for  $q = 1, \dots, n$  do
4:     if  $sort_{auf} = \text{true}$  then
5:       if  $wbht_{folge_p} < wbht_{folge_q}$  then
6:         tausche Positionen von  $folge_p$  und  $folge_q$ 
7:       end if
8:     end if
9:     if  $sort_{ab} = \text{true}$  then
10:      if  $wbht_{folge_p} > wbht_{folge_q}$  then
11:        tausche Positionen von  $folge_p$  und  $folge_q$ 
12:      end if
13:    end if
14:    if  $wbht_{folge_p} = wbht_{folge_q}$  then
15:      if  $ovz_{folge_p} < ovz_{folge_q}$  then
16:        tausche Positionen von  $folge_p$  und  $folge_q$ 
17:      end if
18:    end if
19:   end for
20: end for

```

repräsentiert. Die Indizes p und q verweisen auf die einzelnen Positionen innerhalb dieser Reihenfolge. Mit den booleschen Eingabevariablen $sort_{auf}$ und $sort_{ab}$ wird festgelegt, ob nach den Ofentemperaturen auf- oder absteigend sortiert werden soll.

Intuitiv erscheint die Vorgehensweise der (RBS)-Heuristik sinnvoll. Es stellt sich nun die Frage, ob hiermit auch immer eine bezüglich der Gesamtfertigungszeit optimale Reihenfolge erzielt wird. Anhand eines kleinen Beispiels soll die Behauptung, dass dies nicht der Fall ist, im Folgenden belegt werden. Die Eingangsdaten des Beispiels sind in Tabelle 5.3 zusammengefasst. Zudem sei $templ^{min} = 2000$ und $f_{templ} = 600$.

Nach Anwendung der (RBS)-Heuristik würde sich im Falle einer absteigenden Temperatursortierung die Reihenfolge $folge = (5, 4, 2, 1, 3)$ ergeben. Abbildung 5.7 veranschaulicht die Ofensituation zum Zeitpunkt 300 (also nach 300 min).

Tabelle 5.3: Ein Beispieldatensatz für das HTFSPD

| Wabe j | Ofentemperatur $wbht_j^{Ofen}$ | Ofenverweilzeit ovz_j | Länge l_j |
|----------|--------------------------------|-------------------------|-------------|
| 1 | 620 | 250 | 18000 |
| 2 | 620 | 200 | 18000 |
| 3 | 610 | 400 | 8000 |
| 4 | 660 | 300 | 10000 |
| 5 | 660 | 300 | 10000 |

Die grünlich dargestellten Werte an den Waben kennzeichnen die planmäßigen Ofeneinfahr- und ausfahrzeiten, wobei oben links jeweils die Ofeneinfahrzeitpunkte und unten rechts die Ofenausfahrzeitpunkte abgetragen sind. Die Daten entstammen einer speziellen Ofensimulation, die in Unterkapitel 5.4.1 näher vorgestellt wird.

Anhand von Abbildung 5.7 ist zu erkennen, dass sich die Waben 5, 4 und 2 bereits nach 120 min im Ofen befinden. Die Restkapazität des Ofens (zu diesem Zeitpunkt 16000 mm) reicht jedoch nicht aus, um auch noch Wabe 1 aufzunehmen. Aus diesem Grund muss diese warten bis Wabe 5 aus dem Ofen ausfährt. Da dies nach 310 min der Fall ist, kann Wabe 1 erst zu diesem Zeitpunkt in den Ofen einfahren. Weitere 10 min später folgt Wabe 3 als letzte in der Folge. Da sie eine Ofenverweilzeit von 400 min benötigt, kann sie nach 730 min ausfahren, was gleichzeitig den Gesamtfertigstellungszeitpunkt widerspiegelt.

Nun sei für das gleiche Beispiel die Reihenfolge $folge = (5, 4, 2, 3, 1)$ gegeben, d. h. die Waben 1 und 3 haben ihre Position gegenüber der Lösung nach An-

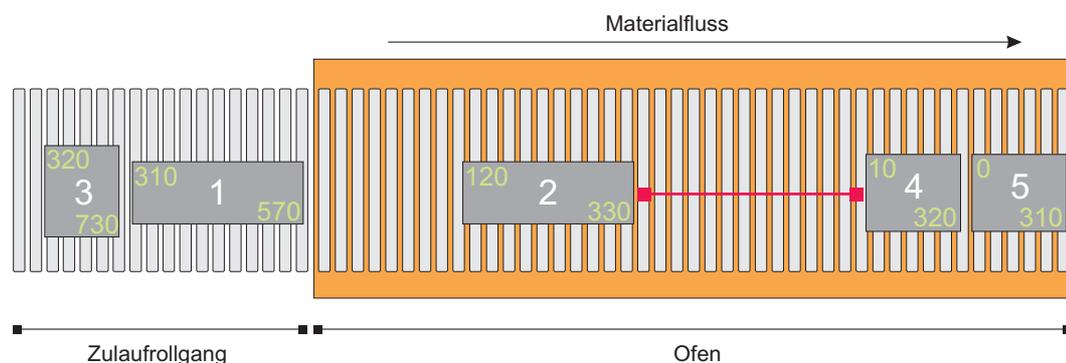


Abbildung 5.7: Lösung des HTFSPD-Beispiels nach Anwendung der *Rule-based Sorting*-Heuristik

wendung der *Rule-based Sorting*-Heuristik getauscht. Die Ofensituation nach 300 min stellt sich nun wie in Abbildung 5.8 veranschaulicht dar.

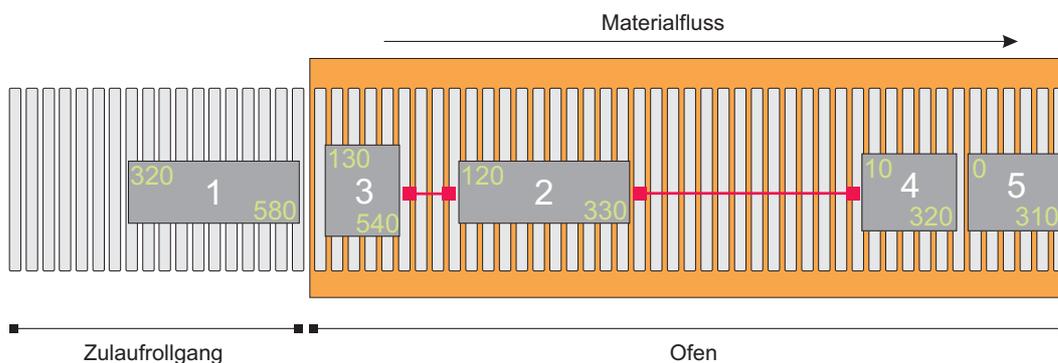


Abbildung 5.8: Bessere Lösung des HTFSPD-Beispiels gegenüber der Lösung der *Rule-based Sorting*-Heuristik

Wie man sieht, war die Restkapazität (von 16000 mm) nach Einfahrt von Wabe 2 für Wabe 3 ausreichend ($16000 - templ_3 - l_3 > 0$), um zum Zeitpunkt 130 in den Ofen einzufahren. Allerdings reicht die Restkapazität nach Ausfahrt von Wabe 5 nun nicht mehr für Wabe 1 aus. Da Wabe 4 jedoch 10 min nach Wabe 5 ausfährt und sich dadurch auch die Temperaturlücke von 24 m zu Wabe 2 auflöst, kann Wabe 1 nach 320 min in den Ofen einfahren. Für diese Folge ergibt sich somit eine Gesamtfertigstellungszeit von 580 min. Dies ist eine Verbesserung von ca. 20,5 % gegenüber der Lösung der (RBS)-Heuristik.

Das Beispiel zeigt, dass die (RBS)-Heuristik nicht zwingend zur optimalen Lösung führt. Das Optimierungspotential soll im folgenden Unterkapitel näher untersucht werden.

5.4 Analyse des Optimierungspotentials

Für eine aussagekräftige Analyse des Optimierungspotentials von Heuristiken empfiehlt sich am ehesten der Vergleich mit einem exakten Verfahren. Auf der einen Seite bieten sich hierzu handelsübliche Solver für gemischt-ganzzahlige Optimierungsprobleme an, welche somit auf das (ZDF)- oder das (EBF)-Modell angewendet werden können. Für beide Modelle ergaben sich nach Anwendung des von IBM angebotenen Solvers ILOG CPLEX Optimi-

zer [51] auf einem Pentium(R) Dual-Core CPU mit 2,8 GHz und einem Arbeitsspeicher von 2 GB (wird auch bei allen weiteren Berechnungen dieser Arbeit verwendet) die in Tabelle 5.4 aufgeführten Ergebnisse bezüglich fünf verschiedener Testinstanzen.

Tabelle 5.4: Vergleich der Rechenergebnisse der (RBS)-Heuristik mit denen des (ZDF)- und des (EBF)-Modells

| Daten- satz | RBS | | ZDF | | EBF | | |
|----------------|-----------|-----------|--------|----------|-----------|--------|----------|
| | C_{max} | C_{max} | Gap | Zeit | C_{max} | Gap | Zeit |
| <i>dat3</i> | 548 | 548 | 0.00 | < 1.00 | 548 | 0.00 | < 1.00 |
| <i>dat5</i> | 846 | 846 | 0.00 | 8125.86 | 846 | 64.72 | 18000.00 |
| <i>dat8</i> | 704 | 652 | 47.01 | 18000.00 | 704 | 76.19 | 18000.00 |
| <i>dat10</i> | 713 | - | 100.00 | 18000.00 | 1441 | 85.63 | 18000.00 |
| <i>dat12</i> | 931 | - | 100.00 | 18000.00 | - | 100.00 | 18000.00 |

In der ersten Spalte wird auf die Datensätze verwiesen, wobei deren Größen (Wabenanzahl n) anhand der Datensatzendungen abzulesen sind. Die zweite Spalte enthält die nach Anwendung der (RBS)-Heuristik resultierenden Lösungen. Für deren Ermittlung wurde wiederum die Ofensimulation herangezogen, welche in Unterkapitel 5.4.1 noch im Detail vorgestellt wird. Die darauffolgenden Spalten beinhalten die entsprechenden Ergebnisse nach Lösen des (ZDF)- und des (EBF)-Modells, wobei eine Laufzeitbegrenzung von fünf Stunden gesetzt wurde. Die Werte für C_{max} entsprechen Minutenangaben, wohingegen die (Lauf-)Zeiten in Sekunden angegeben sind. In den dazwischen liegenden Spalten sind jeweils der *Gap* (in Prozent) in Bezug auf die untere Schranke des Optimalwerts angegeben. Allen Ergebnissen liegt die Annahme von $f_{templ} = 1000$ und $templ^{min} = 2000$ zugrunde.

Bezüglich der Ergebnisse des (ZDF)-Modells ist darauf hinzuweisen, dass hier die Lösung der *Rule-based Sorting*-Heuristik herangezogen wurde, um die Länge des Planungshorizonts zu beschränken. Zudem wurde eine Zeitdiskretisierung vorgenommen, sodass aufeinanderfolgende Zeitpunkte jeweils um 10 Minuten (t^{min}) auseinander liegen. Die Ofenverweilzeiten mussten daher gegebenenfalls auf die nächste Zehnerstelle aufgerundet werden.

Die Ergebnisse zeigen, dass das (ZDF)-Modell leichte Vorteile gegenüber dem (EBF)-Modell aufweist. So kommen zwar beide Modelle bezüglich Datensatz

dat5 zur gleichen Lösung, jedoch kann das (ZDF)-Modell dessen Optimalität noch innerhalb der Laufzeitbegrenzung garantieren. Zudem ist die Lösung des (ZDF)-Modells bezüglich Datensatz *dat8* besser als die Lösung bei Anwendung des (EBF)-Modells. Andererseits liefert das (ZDF)- im Gegensatz zum (EBF)-Modell selbst nach fünfstündiger Laufzeit keine zulässige Lösung für Datensatz *dat10*.

Es bleibt festzuhalten, dass weder das (ZDF)- noch das (EBF)-Modell, außer für die sehr kleinen Probleminstanzen *dat3* und *dat5*, innerhalb der fünf Stunden optimal gelöst werden können. Zum Teil wird nicht einmal eine zulässige Lösung gefunden. Falls doch, so sind diese i. d. R. relativ schlecht, verglichen mit den Lösungen der (RBS)-Heuristik. Folglich lassen die erhaltenen Rechenergebnisse keine aussagekräftige Analyse des Optimierungspotentials der (RBS)-Heuristik zu.

Das Ziel, auch Probleminstanzen mittlerer Größe exakt zu lösen, soll mit einem *Branch&Bound*-Verfahren erreicht werden, welches in Unterkapitel 5.4.2 erläutert wird. Zunächst wird jedoch die bereits erwähnte Ofensimulation näher vorgestellt, welcher auch im Zusammenhang mit dem *Branch&Bound*-Verfahren eine wichtige Rolle zukommt.

5.4.1 Eine Materialflusssimulation des Ofens

Das Werkzeug Simulation ist in der heutigen Zeit aus vielen technischen und wissenschaftlichen Bereichen nicht mehr wegzudenken. Dies gilt auch für den Bereich der Logistik, wo Simulation beispielsweise zur Planung, Steuerung und Überwachung, aber auch zur Optimierung bestimmter Material-, Personen- und Informationsflüsse eingesetzt wird. Im Rahmen der VDI Richtlinie VDI 3633 [94] wird der Begriff Simulation wie folgt definiert: „Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“ Zur Konkretisierung dieser Definition sei die Klassifizierung von Law und Kelton [59] herangezogen. Demnach unterscheidet man

- statische vs. dynamische,

- deterministische vs. stochastische und
- kontinuierliche vs. diskrete Simulationsmodelle.

Bei statischen Systemen wird nur ein bestimmter Zeitpunkt betrachtet oder der Zeitbetrachtung kommt keine besondere Rolle zu. In dynamischen Simulationsmodellen wird das zeitliche Verhalten eines Systems hingegen explizit abgebildet. Als deterministisch beschreibt man ein System, welches keine zufallsabhängigen Komponenten enthält. Im Gegensatz dazu, liegen diese bei stochastischen Modellen ausdrücklich vor. Letztendlich lassen sich kontinuierliche, d. h. ständigen Änderungen unterworfenen Modelle noch von diskreten Modellen unterscheiden, bei denen sich die Systemzustände nur zu bestimmten diskreten Zeitpunkten ändern.

Die im Rahmen der vorliegenden Arbeit entwickelte Ofensimulation bildet ein dynamisches, deterministisches und (ereignis-)diskretes System ab. Sie wird zur Ermittlung der Gesamtfertigstellungszeitpunkte C_{max} bestimmter Produktionsfolgen verwendet und dient im Zuge der späteren Entwicklung verschiedener Lösungsverfahren demnach als Bewertungsfunktion. In diesem Zusammenhang spricht man auch von simulationsgestützter Optimierung. März et al. [66] beschreiben verschiedene Möglichkeiten zur Kopplung von Simulation und Optimierung. Den theoretischen Ausführungen schließen sich zudem verschiedene Fallbeispiele aus der Praxis an. Eine Einführung in die Erstellung ereignisdiskreter Simulationsmodelle bietet z. B. Eley [35].

Im Folgenden wird der konkrete Ablauf der Ofensimulation vorgestellt. Zunächst werden die folgenden Zeiten definiert:

- $FAZO_j$: Frühestmöglicher Anfangszeitpunkt von Wabe j ,
 $FEZO_j$: Frühestmöglicher Endzeitpunkt von Wabe j .

$FAZO_j$ bezeichnet demnach den frühestmöglichen Zeitpunkt, zu dem die Ofeneinfahrt von Wabe j beginnen kann. Analog bezeichnet $FEZO_j$ den frühestmöglichen Zeitpunkt, zu dem die Ofenausfahrt von Wabe j beginnen kann.

Des Weiteren werden die folgenden Notationen eingeführt:

| | |
|---------------------|---|
| W^{Ofen} : | Geordnete Menge der aktuell im Ofen befindlichen Waben, |
| l_{rest}^{Ofen} : | Aktuelle Restkapazität des Ofens, |
| wl_j : | Wartezeit aufgrund der Länge für Wabe j . |

Zudem dient der Index p wiederum dazu, um auf die einzelnen Positionen innerhalb der Mengen $folge$ und W^{Ofen} zu verweisen.

Die Materialflusssimulation des Ofens lässt sich algorithmisch durch den in Algorithmus 2 dargestellten Pseudocode abbilden. Er wird im Folgenden als *Simulation-1* bezeichnet, da im Verlauf dieser Arbeit noch weitere Varianten hinzukommen werden.

Algorithmus 2 *Simulation-1*

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , mit $j \in J$,
 $folge$, f_{templ} , $templ^{min}$, t^{min}

Ensure: $FAZO_j$, $FEZO_j$, mit $j \in J$, C_{max}

```

1: for  $p = 1, \dots, n$  do
2:   if  $p = 1$  then
3:      $templ_{folge_p} = 0$ 
4:      $FAZO_{folge_p} = 0$ 
5:   else
6:      $templ_{folge_p} = \max\{(|wbht_{folge_p}^{Ofen} - wbht_{folge_{p-1}}^{Ofen}| \cdot f_{templ}), templ^{min}\}$ 
7:      $FAZO_{folge_p} = FAZO_{folge_{p-1}} + t^{min}$ 
8:   end if
9:    $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
10:  if  $|W^{Ofen}| > 0$  then
11:    if  $FEZO_{W_1^{Ofen}} + t^{min} \geq FEZO_{folge_p}$  then
12:       $FAZO_{folge_p} \pm FEZO_{W_1^{Ofen}} + t^{min} - FEZO_{folge_p}$ 
13:       $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
14:    end if
15:    while  $FEZO_{W_{|W^{Ofen}|}^{Ofen}} \leq FAZO_{folge_p}$  do
16:      if  $|W^{Ofen}| \leq 1$  then
17:         $l_{rest}^{Ofen} = l^{Ofen}$ 
18:      else
19:         $l_{rest}^{Ofen} \pm l_{W_{|W^{Ofen}|}^{Ofen}} + templ_{W_{|W^{Ofen}|-1}^{Ofen}}$ 
20:      end if
21:       $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}^{Ofen}\}$ 
22:    end while
23:    while  $l_{rest}^{Ofen} \leq l_{folge_p} + templ_{folge_p}$  do

```

```

24:         if  $|W^{Ofen}| = 0$  then
25:              $l_{rest}^{Ofen} = l^{Ofen}$ 
26:         end if
27:         if  $FEZO_{W_{|W^{Ofen}|}} > FAZO_{folge_p}$  then
28:              $wl_{folge_p} = FEZO_{W_{|W^{Ofen}|}} - FAZO_{folge_p}$ 
29:         end if
30:         if  $|W^{Ofen}| = 1$  then
31:              $l_{rest}^{Ofen} = l^{Ofen}$ 
32:         else
33:              $l_{rest}^{Ofen} \pm l_{W_{|W^{Ofen}|}}^{Ofen} + templ_{W_{|W^{Ofen}|-1}}^{Ofen}$ 
34:         end if
35:          $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}^{Ofen}\}$ 
36:     end while
37:      $FAZO_{folge_p} \pm wl_{folge_p}$ 
38:      $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
39: end if
40: if  $|W^{Ofen}| = 0$  then
41:      $l_{rest}^{Ofen} \equiv l_{folge_p} - templ_{folge_p}$ 
42: else
43:      $l_{rest}^{Ofen} \equiv l_{folge_p}$ 
44: end if
45:      $W^{Ofen} \leftarrow W^{Ofen} \cup \{folge_p\}$ 
46: end for
47:  $C_{max} = FEZO_{folge_n}$ 

```

Der Algorithmus beginnt mit $folge_1$ und somit der Wabe, welche in der Bearbeitungsreihenfolge an erster Stelle positioniert ist. In den Zeilen 2 - 9 werden zunächst die Temperaturlücken zur Vorgängerwabe sowie die vorläufigen Anfangs- und Endzeitpunkte errechnet. Da weiterhin die Annahme gilt, dass der Ofen zum Startzeitpunkt 0 leer ist, werden $templ_{folge_1}$ und $FAZO_{folge_1}$ auf 0 gesetzt. Damit entfallen für die erste Wabe auch die Berechnungen der Zeilen 10 - 39. Die Wabe kann direkt einfahren, verringert damit die Restkapazität des Ofens um ihre Länge (Zeile 41) und wird folglich der aktuellen Ofenbelegung zugeordnet (Zeile 45).

Für die folgenden Waben müssen auch die Berechnungen in den Zeilen 10 - 39 durchgeführt werden. Zunächst wird geprüft, ob sich die vorläufigen Anfangs- und Endzeitpunkte aufgrund von Wartezeiten durch unterschiedliche Ofenverweilzeiten erhöhen (Zeilen 11 - 14). Anschließend wird die Ofenbelegung sowie

die Restkapazität des Ofens aktualisiert (Zeilen 15 - 22). Zuletzt muss noch geprüft werden, ob die Restkapazität des Ofens für die Länge der betreffenden Wabe zuzüglich ihrer Temperaturlücke ausreicht oder ob Wartezeiten aufgrund der Länge entstehen, die dann wiederum auf die Anfangs- und Endzeitpunkte aufzuschlagen sind (Zeilen 23 - 38). Der Gesamtfertigstellungszeitpunkt C_{max} ergibt sich schließlich aus dem frühestmöglichen Endzeitpunkt der in der Bearbeitungsreihenfolge an letzter Stelle positionierten Wabe (Zeile 47).

Die in *Simulation-1* errechneten Ofenein- und ausfahrzeiten ermöglichen neben der Ermittlung des Gesamtfertigstellungszeitpunkts C_{max} auch die Nachbildung der sich im Zeitablauf verändernden Ofensituationen. Dieser Aspekt wurde bereits für die Erstellung der Abbildungen 5.7 und 5.8 genutzt. Entscheidend ist allerdings, dass mit *Simulation-1* nun ein Werkzeug vorliegt, um beliebige Produktionsfolgen bewerten und damit vergleichen zu können. Dies bildet die Grundlage für die Entwicklung effizienter Lösungsverfahren, so z. B. auch für das im nachfolgenden Unterkapitel beschriebene *Branch&Bound*-Verfahren.

5.4.2 Ein *Branch&Bound*-Verfahren

Die Idee, ein *Branch&Bound*-Verfahren für das HTFSPD zu entwickeln, beruht auf einem von Baker und Trietsch [4] vorgeschlagenen Ansatz für ein Einmaschinen-Reihenfolgeproblem. Jedes Teilproblem wird dabei durch die Fixierung bestimmter Positionen in der Bearbeitungsreihenfolge repräsentiert.

Ein Teilproblem bzw. Knoten des *HTFSPD-Branch&Bound*-Verfahrens setzt sich aus den folgenden Elementen zusammen:

- einer fixierten Reihenfolge,
- einer oberen Schranke (engl. *upper bound*) und
- einer unteren Schranke (engl. *lower bound*).

Abbildung 5.9 stellt einen beispielhaften Knoten dar. Die (Teil-)Lösung dieses Knotens wird durch $folge' = (4, 5, 8, 2)$ repräsentiert. Die ersten vier Positionen der (Gesamt-)Lösung sind somit fixiert. Im Fall von $n = 8$ wären die Waben

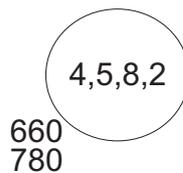


Abbildung 5.9: Ein beispielhafter Knoten des
HTFSPD-Branch&Bound-Verfahrens

1, 3, 6 und 7 folglich noch zu verplanen. Die Werte unten links bilden die jeweiligen Schranken. Dabei kennzeichnet der obere Wert die untere Schranke und der untere entsprechend die obere Schranke, wobei es sich hier um Minutenangaben handelt. Die optimale Gesamtfertigstellungszeit C_{max} muss folglich zwischen diesen beiden Werten liegen. Im Folgenden sollen die für das HTFSPD speziellen Berechnungen der Schranken (insbesondere der unteren) näher erläutert werden.

Eine untere Schranke muss garantieren, dass der optimale Lösungswert in jedem Fall größer oder gleich dem Schrankenwert ist. Dies kann einerseits mit einfachen Schrankenberechnungen, die zu sehr kleinen Werten führen, realisiert werden. Andererseits lassen sich Teilprobleme bei größeren unteren Schranken schneller ausloten, was der Effizienz des Verfahrens zugutekommt. Demnach empfiehlt es sich durchaus, sich gezielt mit der Berechnung der unteren Schranken zu befassen. Dabei ist jedoch darauf zu achten, den Rechenaufwand für diese Berechnungen möglichst gering zu halten, da sie im Verlauf eines *Branch&Bound*-Verfahrens sehr häufig durchgeführt werden müssen.

Die untere Schrankenberechnung für das *HTFSPD-Branch&Bound*-Verfahren beruht auf der Idee bzw. Annahme, den Ofenraum immer komplett ausnutzen zu können. Hierzu werden die noch nicht fixierten Waben zu einer langen, fiktiven und beliebig teilbaren Wabe \tilde{j} zusammengefasst. Deren Länge $l_{\tilde{j}}$ setzt sich dann sowohl aus den einzelnen Wabenlängen als auch aus den mindestens erforderlichen Temperaturlücken zusammen.

Die einzelnen Schritte zur Berechnung einer unteren Schranke C_{max}^{lb} für das HTFSPD lassen sich anhand des in Algorithmus 3 abgebildeten Pseudocodes eingehender erläutern. Gegeben seien eine geordnete Menge fixierter Waben *folge'* der Mächtigkeit n' und eine Menge unverplanter Waben *unvp*. Zudem seien die minimalen und maximalen Ofentemperaturen der Waben innerhalb

von $unvp$ mit $wbht_{min}^{Ofen,unvp}$ bzw. mit $wbht_{max}^{Ofen,unvp}$ bezeichnet und analog hierzu die Ofenverweilzeiten mit ovz_{min}^{unvp} bzw. ovz_{max}^{unvp} .

Algorithmus 3 *HTFSPD-Branch&Bound-lb*

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , mit $j \in J$, f_{templ} , $templ^{min}$, t^{min} , $folge'$, n' , $unvp$, $wbht_{min}^{Ofen,unvp}$, $wbht_{max}^{Ofen,unvp}$, ovz_{min}^{unvp} , ovz_{max}^{unvp}

Ensure: C_{max}^{lb}

```

1: if  $|wbht_{folge_{n'}}^{Ofen} - wbht_{min}^{Ofen,unvp}| < |wbht_{folge_{n'}}^{Ofen} - wbht_{max}^{Ofen,unvp}|$  then
2:    $templ_{max}^{unvp} = \max\{|wbht_{folge_{n'}}^{Ofen} - wbht_{max}^{Ofen,unvp}| \cdot f_{templ}, templ^{min}\}$ 
3:   sortiere  $unvp$  bis  $wbht_{folge_{n'}}^{Ofen}$  nach  $wbht^{Ofen}$  absteigend
4:   sortiere  $unvp$  ab  $wbht_{folge_{n'}}^{Ofen}$  nach  $wbht^{Ofen}$  aufsteigend
5: else
6:    $templ_{max}^{unvp} = \max\{|wbht_{folge_{n'}}^{Ofen} - wbht_{min}^{Ofen,unvp}| \cdot f_{templ}, templ^{min}\}$ 
7:   sortiere  $unvp$  bis  $wbht_{folge_{n'}}^{Ofen}$  nach  $wbht^{Ofen}$  aufsteigend
8:   sortiere  $unvp$  ab  $wbht_{folge_{n'}}^{Ofen}$  nach  $wbht^{Ofen}$  absteigend
9: end if
10: for  $p = 1, \dots, |unvp|$  do
11:   if  $p = 1$  then
12:      $templ_{unvp_p} = \max\{|wbht_{unvp_p}^{Ofen} - wbht_{folge_{n'}}^{Ofen}| \cdot f_{templ}, templ^{min}\}$ 
13:   else
14:      $templ_{unvp_p} = \max\{|wbht_{unvp_p}^{Ofen} - wbht_{unvp_{p-1}}^{Ofen}| \cdot f_{templ}, templ^{min}\}$ 
15:   end if
16:    $l_j^{\pm} \stackrel{\pm}{=} l_{unvp_p} + templ_{unvp_p}$ 
17: end for
18: call Simulation-1 für  $folge'$ 
19:  $u = 0$ 
20: while  $l_j^{\pm} > 0$  do
21:    $u \stackrel{\pm}{=} 1$ 
22:   if  $u = 1$  then
23:      $FAZO_{j_u} = FAZO_{W_{|W^{Ofen}|}^{Ofen}} + t^{min}$ 
24:      $FEZO_{j_u} = FAZO_{j_u} + t^{min} + ovz_{max}^{unvp}$ 
25:      $templ_{j_u} = templ_{max}^{unvp}$ 
26:   else
27:      $FAZO_{j_u} = FEZO_{W_{|W^{Ofen}|}^{Ofen}}$ 
28:      $FEZO_{j_u} = FAZO_{j_u} + t^{min} + ovz_{min}^{unvp}$ 
29:      $templ_{j_u} = 0$ 
30:   end if
31:   if  $FEZO_{W_1^{Ofen}} + t^{min} \geq FEZO_{j_u}$  then
32:      $FAZO_{j_u} \stackrel{\pm}{=} FEZO_{W_1^{Ofen}} + t^{min} - FEZO_{j_u}$ 
33:   if  $h = 1$  then

```

```

34:         FEZO $\tilde{j}_u$  = FAZO $\tilde{j}_u$  +  $t^{min}$  +  $ovz_{max}^{unvp}$ 
35:     else
36:         FEZO $\tilde{j}_u$  = FAZO $\tilde{j}_u$  +  $t^{min}$  +  $ovz_{min}^{unvp}$ 
37:     end if
38: end if
39: while FEZO $_{W_{|W^{Ofen}|}}$  ≤ FAZO $\tilde{j}_u$  do
40:      $l_{rest}^{Ofen} \pm l_{W_{|W^{Ofen}|}}^{Ofen} + templ_{W_{|W^{Ofen}|-1}}^{Ofen}$ 
41:      $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}\}$ 
42: end while
43:  $l_{\tilde{j}_u} = l_{rest}^{Ofen}$ 
44:  $\tilde{l}^j = l_{\tilde{j}_u}$ 
45:  $l_{rest}^{Ofen} = 0$ 
46:  $W^{Ofen} \leftarrow W^{Ofen} \cup \{\tilde{j}_u\}$ 
47: end while
48:  $C_{max}^{lb} = FEZO_{\tilde{j}_u}$ 

```

Zunächst werden die unverplanten Waben derart sortiert, dass die erforderlichen Temperaturlücken minimiert werden (Zeilen 1 - 9). Für die Sortierrichtung ist dabei die Ofentemperatur der Wabe ausschlaggebend, die der letzten Position von *folge'* zugeordnet ist. Zudem wird die größte Abweichung dieser Ofentemperatur in Bezug auf die Ofentemperaturen der in *unvp* enthaltenen Waben ermittelt ($templ_{max}^{unvp}$). In den Zeilen 10 - 17 werden dann für alle Waben aus *unvp* die entsprechenden Temperaturlücken berechnet und zusammen mit den Wabengängen zu $l_{\tilde{j}}$ aufaddiert. Anschließend wird *folge'* durch Aufruf von *Simulation-1* simuliert (Zeile 18).

In dem Abschnitt von Zeile 20 bis Zeile 47 folgen schließlich die Berechnungen für die fiktive Wabe \tilde{j} bzw. für deren Teile, die mit dem Index *u* fortlaufend gekennzeichnet sind. Zunächst werden die vorläufigen Anfangs- und Endzeitpunkte initialisiert (Zeilen 22 - 30), wobei dem ersten fiktiven Wabenteil die Ofenverweilzeit ovz_{max}^{unvp} und die Temperaturlücke $templ_{max}^{unvp}$ zugeordnet wird. Sofern Wartezeiten aufgrund der Ofenverweilzeiten entstehen, sind die vorläufigen Anfangs- und Endzeitpunkte entsprechend zu aktualisieren (Zeilen 31 - 38). Des Weiteren wird in den Zeilen 39 - 42, falls erforderlich, die aktuelle Ofenbelegung sowie die damit verbundene Restkapazität des Ofens ermittelt, welche schließlich für die Initialisierung der Länge des fiktiven Wabenteils \tilde{j}_u maßgebend ist (Zeile 43). Daraufhin wird diese Länge $l_{\tilde{j}_u}$ von der Gesamtlänge

ge l_j abgezogen (Zeile 44), die Restkapazität des Ofens gleich 0 gesetzt (Zeile 45) und \tilde{j}_u der aktuellen Ofenbelegung zugeordnet (Zeile 46). Sofern l_j noch positiv ist, wiederholen sich die beschriebenen Berechnungen, wobei den folgenden Wabenteilen stets die Ofenverweilzeit ovz_{min}^{unvp} zugeordnet und von einer Temperaturlücke abgesehen wird. Gilt $l_j \leq 0$, so endet der Algorithmus mit einer unteren Schranke, die dem Endzeitpunkt des zuletzt der Ofenbelegung zugeordneten fiktiven Wabenteils entspricht (Zeile 48).

Für die Berechnung einer oberen Schranke C_{max}^{ub} ist mit der (RBS)-Heuristik im Prinzip schon ein potentiell Verfahren vorhanden. Da die (RBS)-Heuristik jedoch beim Start immer von einem leeren Ofen ausgeht, muss die Sortierung wie schon bei der unteren Schrankenberechnung an die Ofentemperatur der an letzter Stelle von $folge'$ positionierten Wabe angepasst werden.

Algorithmus 4 fasst die Schritte zur Berechnung einer oberen Schranke C_{max}^{ub} für das HTFSPD zusammen. Im Gegensatz zur unteren Schrankenberechnung ist hier zu beachten, dass die Menge der unverplanten Waben nach deren Sortierung (Zeilen 1 - 7) der Menge der bereits fixierten Waben zugeordnet wird (Zeile 8). Demnach liegt bei Aufruf von *Simulation-1* (Zeile 9) mit $folge'$ eine vollständige Reihenfolge vor. Die obere Schranke C_{max}^{ub} ist folglich dem Endzeitpunkt der letzten Wabe aus $folge'$ gleichzusetzen (Zeile 10).

Algorithmus 4 HTFSPD-Branch&Bound-ub

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , mit $j \in J$, f_{templ} , $templ^{min}$, t^{min}
 $folge'$, n' , $unvp$, $wbht_{min}^{Ofen,unvp}$, $wbht_{max}^{Ofen,unvp}$

Ensure: C_{max}^{ub}

- 1: **if** $|wbht_{folge_{n'}}^{Ofen} - wbht_{min}^{Ofen,unvp}| < |wbht_{folge_{n'}}^{Ofen} - wbht_{max}^{Ofen,unvp}|$ **then**
 - 2: sortiere $unvp$ bis $wbht_{folge_{n'}}^{Ofen}$ nach $wbht^{Ofen}$ absteigend
 - 3: sortiere $unvp$ ab $wbht_{folge_{n'}}^{Ofen}$ nach $wbht^{Ofen}$ aufsteigend
 - 4: **else**
 - 5: sortiere $unvp$ bis $wbht_{folge_{n'}}^{Ofen}$ nach $wbht^{Ofen}$ aufsteigend
 - 6: sortiere $unvp$ ab $wbht_{folge_{n'}}^{Ofen}$ nach $wbht^{Ofen}$ absteigend
 - 7: **end if**
 - 8: $folge' \leftarrow folge' \cup \{unvp\}$
 - 9: **call** *Simulation-1* für $folge'$
 - 10: $C_{max}^{ub} = FEZO_{folge'_{n'}}$
-

Domschke und Drexl [32] zählen zu den wichtigsten Komponenten eines jeden *Branch&Bound*-Verfahrens:

1. die Anwendung einer Startheuristik,
2. Regeln zur Bildung von Relaxationen,
3. Regeln zum Ausloten von Teilproblemen,
4. Regeln bezüglich der Suchstrategie und
5. Regeln zum Verzweigen eines Teilproblems.

Für das *HTFSPD-Branch&Bound*-Verfahren wird als Startheuristik (1. Komponente) die (RBS)-Heuristik verwendet. Sie wird jeweils einmal mit $sort_{auf} = true$ und einmal mit $sort_{ab} = true$ durchgeführt und mit Hilfe von *Simulation-1* bewertet. Der bessere Lösungswert sowie die zugehörige Reihenfolge werden schließlich gespeichert.

Die Bildung einer Relaxation eines bestimmten Teilproblems (2. Komponente) wird entsprechend des Ansatzes von Baker und Trietsch [4] durch Fixierung einer weiteren Position in der Reihenfolge realisiert. Ein Teilproblem mit $folge' = (4, 5, 8, 2, 1)$ wäre somit eine Relaxation des in Abbildung 5.9 dargestellten Teilproblems. Insgesamt können daraus demnach vier relaxierte Teilprobleme erzeugt werden, welche sich wiederum selbst relaxieren lassen. Sobald jedoch $n - 1$ Waben fixiert sind, ist automatisch auch eine fixierte Gesamtbehandlungsreihenfolge erreicht. In diesem Fall kann aus dem entsprechenden Teilproblem dann keine weitere Relaxation mehr erzeugt werden.

In diesem Abschnitt folgen nun die Regeln des *HTFSPD-Branch&Bound*-Verfahrens zum Ausloten von Teilproblemen (3. Komponente). So kann ein Teilproblem ausgelotet werden, wenn:

- dessen untere Schranke größer oder gleich der aktuell besten bekannten Lösung ist,
- die Anzahl fixierter Waben $n - 1$ entspricht oder
- allgemein die Anzahl der bereits erzeugten Relaxierungen gleich der Anzahl der unverplanten Waben ist.

Als Suchstrategie (4. Komponente) werden für das *HTFSPD-Branch&Bound*-

Verfahren drei verschiedene Varianten vorgeschlagen:

- *Depth First Promise Next* (BB-DFPN),
- *Promise First Depth Next* (BB-PFDN),
- *Promise Depth Relation* (BB-PDR).

Die (BB-DFPN)-Suchstrategie entspricht im Prinzip der klassischen Tiefensuche. Es wird immer das Teilproblem als nächstes betrachtet, welches von allen möglichen Kandidaten die meisten bereits fixierten Waben enthält. Wenn diese Anzahl für einige Teilprobleme identisch sein sollte, so wird daraus der aussichtsreichste Knoten gewählt, wobei diese Eigenschaft anhand des oberen Schrankenwerts bemessen wird. Die (BB-PFDN)-Strategie geht genau umgekehrt vor. Hier wird zunächst das Teilproblem mit der kleinsten oberen Schranke gewählt. Erst bei mehrfacher Existenz dieses Wertes wird aus den betreffenden Teilproblemen dasjenige gewählt, welches die meisten bereits fixierten Waben enthält.

Die beiden ersten Varianten sind klassisch und weisen demnach typische Stärken und Schwächen auf. Bei Anwendung der (BB-DFPN)-Strategie bzw. allgemein der Tiefensuche resultieren i. d. R. lange Laufzeiten. Dafür ist jedoch der Speicherbedarf im Vergleich zur (BB-PFDN)-Suchstrategie wesentlich geringer. Da diese eher einer klassischen Breitensuche entspricht, kann dabei schnell ein sehr hoher Speicherplatzbedarf und Verwaltungsaufwand entstehen. Dagegen ist die erste erhaltene zulässige Lösung hier im Vergleich zur (BB-DFPN)-Strategie i. d. R. schon relativ gut.

Die (BB-PDR)-Strategie stellt eine Mischform der beiden bereits erläuterten Suchstrategien dar. Hier werden die Ziele einer möglichst kleinen oberen Schranke und einer möglichst großen Anzahl an bereits fixierten Waben gegeneinander abgewogen. So wird bei Anwendung der (BB-PDR)-Strategie immer der Knoten als nächstes gewählt, für den der Wert pdr gemäß

$$pdr = \frac{(n - |folge'|)}{n} + \frac{(C_{max}^{ub} - C_{max}^{akt})}{C_{max}^{akt}} \quad (5.52)$$

minimal ist, wobei C_{max}^{akt} den aktuell besten bekannten Lösungswert bezeichnet.

Bezüglich der Verzweigungsregeln eines Teilproblems (5. Komponente) gibt es zwei denkbare Varianten. Zum einen könnten immer direkt alle $n - |folge'|$ möglichen Relaxierungen erzeugt werden. Eine derartige Strategie hat den Vorteil, das ursprüngliche Teilproblem direkt ausloten zu können. Andererseits resultiert aus dieser Strategie, insbesondere bei steigender Wabenanzahl, ein sehr hoher Speicherplatzbedarf. Um dies zu vermeiden, empfiehlt sich als Verzweigungsregel eher die zweite Variante, bei der zunächst immer nur eine Relaxierung erzeugt wird. Für die weiteren Ausführungen kommt daher ausschließlich diese Variante zum Tragen.

Algorithmus 5 stellt den Ablauf des *HTFSPD-Branch&Bound*-Verfahrens in Form eines Pseudocodes dar. In einem ersten Schritt wird C_{max}^{akt} nach Aufruf der (RBS)-Heuristik und anschließender Bewertung durch *Simulation-1* mit C_{max} initialisiert (Zeilen 1 - 2). Daraufhin wird ein Ursprungsknoten k erzeugt und einer Kandidatenliste K zugeordnet (Zeile 3 - 4). Dabei wird der Ursprungsknoten mit einer leeren fixierten Wabenmenge $folge'_k$ initialisiert. In der Kandidatenliste werden im weiteren Verlauf des Verfahrens alle noch zu prüfenden bzw. noch nicht ausgeloteten Knoten gespeichert. Die Schritte in den Zeilen 6 - 38 werden folglich solange durchgeführt, bis K leer ist. Es sei darauf hingewiesen, dass die einzelnen in K enthaltenen Knoten nach ihrer Priorität absteigend sortiert werden. Ein neu aufgenommenen Knoten wird somit automatisch an die seiner Priorität entsprechende Position gesetzt, wobei für diese Priorisierung die gewählte Suchstrategie maßgebend ist.

Solange K noch Knoten enthält, wird für K_1 zunächst geprüft, ob dessen unterer Schrankenwert C_{max,K_1}^{lb} größer oder gleich C_{max}^{akt} ist oder ob der Knoten bereits vollständig verzweigt ist (Zeilen 6 - 8). Hierfür wird die boolesche Variable $open_{k,j}$ eingeführt, welche angibt ob Knoten k bereits in Richtung Wabe j verzweigt wurde. Sofern eine der beiden Bedingungen erfüllt sind, wird K_1 aus K entfernt. Wenn dies nicht der Fall sein sollte, so wird ein neuer Knoten k mit $folge'_k \leftarrow folge'_{K_1} \cup \{j\}$ erzeugt (Zeilen 11 - 13) ehe im nächsten Schritt die Schranken für den neu entstandenen Knoten berechnet werden (Zeile 14).

Falls der untere Schrankenwert $C_{max,k}^{lb}$ größer oder gleich C_{max}^{akt} sein sollte, so wird Knoten k direkt wieder gelöscht (Zeilen 15 - 16). Sollte dagegen der obere Schrankenwert $C_{max,k}^{ub}$ unter C_{max}^{akt} liegen, so liegt eine Lösungsverbesserung vor.

Algorithmus 5 *HTFSPD-Branch&Bound*

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , mit $j \in J$, f_{templ} , $templ^{min}$, t^{min}

Ensure: C_{max}^{glob} , $folge^{glob}$

```

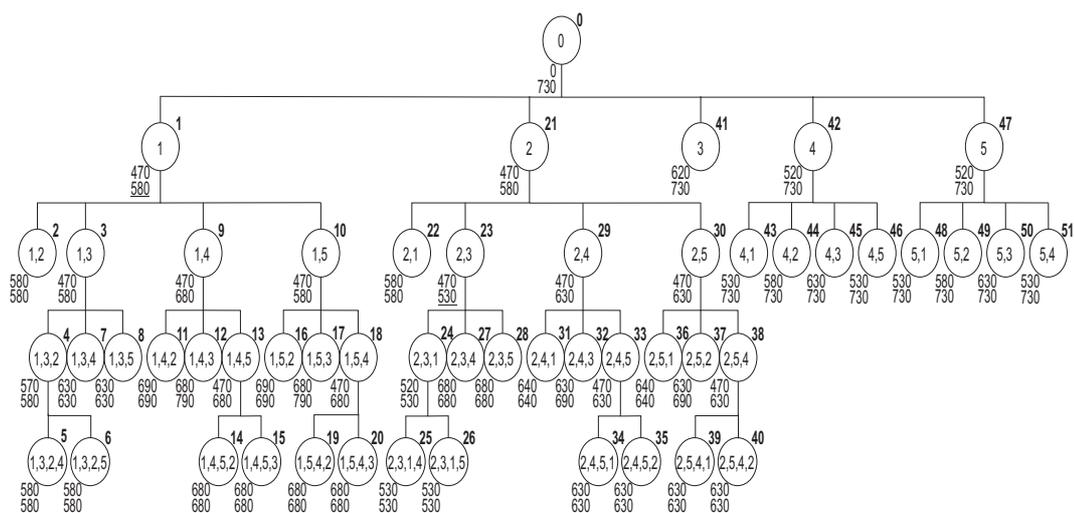
1: call Rule-based Sorting
2: call Simulation-1 und setze  $C_{max}^{akt} = C_{max}$ 
3: erzeuge Ursprungsknoten  $k$  mit  $folge'_k = ()$ ,  $C_{max,k}^{lb} = 0$ ,  $C_{max,k}^{ub} = C_{max}^{akt}$ 
4:  $K \leftarrow K \cup \{k\}$ 
5: while  $|K| \neq 0$  do
6:   if  $C_{max,K_1}^{lb} \geq C_{max}^{akt}$  or  $open_{K_1,j} = \mathbf{true} \forall j \in \{J\}$  then
7:      $K \leftarrow K \setminus \{K_1\}$ 
8:   else
9:     for  $j = 1, \dots, n$  do
10:      if  $j \notin folge'_{K_1}$  and  $open_{K_1,j} = \mathbf{false}$  then
11:        erzeuge neuen Knoten  $k$ 
12:         $open_{K_1,j} = \mathbf{true}$ 
13:         $folge'_k \leftarrow folge'_{K_1} \cup \{j\}$ 
14:        berechne  $C_{max,k}^{lb}$  und  $C_{max,k}^{ub}$ 
15:        if  $C_{max,k}^{lb} \geq C_{max}^{akt}$  then
16:          lösche Knoten  $k$ 
17:        else if  $C_{max,k}^{ub} < C_{max}^{akt}$  then
18:          aktualisiere  $folge^{akt}$  und  $C_{max}^{akt}$ 
19:          for  $k' \in K$  do
20:            if  $C_{max,k'}^{lb} \geq C_{max}^{akt}$  then
21:               $K \leftarrow K \setminus \{k'\}$ 
22:            end if
23:          end for
24:           $K \leftarrow K \cup \{k\}$ 
25:        else if  $|folge'_k| = n - 1$  then
26:          aktualisiere  $folge^{akt}$  und  $C_{max}^{akt}$ 
27:          for  $k' \in K$  do
28:            if  $C_{max,k'}^{lb} \geq C_{max}^{akt}$  then
29:               $K \leftarrow K \setminus \{k'\}$ 
30:            end if
31:          end for
32:        else
33:           $K \leftarrow K \cup \{k\}$ 
34:        end if
35:        break
36:      end if
37:    end for
38:  end if
39: end while
40:  $C_{max}^{glob} = C_{max}^{akt}$ ,  $folge^{glob} = folge^{akt}$ 

```

Neben der Aktualisierung von C_{max}^{akt} und der damit verbundenen aktuell besten Reihenfolge $folge^{akt}$, werden dann alle Knoten aus der Menge K entfernt, die sich aufgrund des neuen Lösungswerts jetzt ausloten lassen (Zeilen 17 - 23). Im Anschluss daran, wird der Knoten k selbst noch der Kandidatenliste hinzugefügt (Zeile 24). Dieselben Berechnungen, mit Ausnahme des Einfügens von k in die Kandidatenliste, werden in den Zeilen 26 - 31 vorgenommen, falls Knoten k nur noch einen unfixierten Knoten ($n - 1$ fixierte Knoten) aufweist.

Sollten die Bedingungen aus den Zeilen 15, 17 und 25 nicht erfüllt sein, so wird Knoten k einfach nur der Kandidatenliste zugefügt. Sobald K keine Knoten mehr enthält, ist die bis dahin beste gefundene Lösung C_{max}^{akt} optimal. In diesem Fall endet der Algorithmus mit der Ausgabe des optimalen Zielfunktionswerts C_{max}^{glob} , sowie der zugehörigen Reihenfolge $folge^{glob}$ (Zeile 40).

Abbildung 5.10 (bzw. im Anhang Abbildung A.1) veranschaulicht den Ablauf des *HTFSPD-Branch&Bound*-Verfahrens für das in Kapitel 5.3 eingeführte Beispiel (vgl. Tabelle 5.3). Die an den Knoten jeweils oben rechts abgebildeten Werte kennzeichnen die Reihenfolgennummer hinsichtlich der Erstellung des betreffenden Knotens und lassen damit die Suchstrategie erkennen. In dem abgebildeten Beispiel wurde die (BB-DFPN)-Strategie angewendet. Nach Erstellung des Ursprungsknotens wird zunächst der Knoten (1) bzw. (0,1) erzeugt. Da dessen obere Schranke (mit 580 min) unter dem aktuell besten bekannten Lösungswert (730 min) liegt, ist somit bereits eine Lösungsverbesserung erzielt



worden. Im nächsten Schritt wird Knoten (1,2) erzeugt, welcher dann aufgrund seiner unteren Schranke (mit 580 min) direkt ausgelotet werden kann. Das Verfahren schreitet mit der Erstellung der Knoten (1,3), (1,3,2), (1,3,2,4), (1,3,2,5), (1,3,4), usw. fort.

An Knoten (2,3) angekommen, kann der beste bekannte Lösungswert aufgrund der oberen Schranke (mit 530 min) ein weiteres Mal aktualisiert werden. Da im weiteren Verlauf des Verfahrens keine weiteren Verbesserungen mehr erzielt werden, ist dieser Lösungswert optimal. Die zugehörige Bearbeitungsreihenfolge ist $folge^{glob} = (2, 3, 1, 4, 5)$. Die in Abbildung 5.8 dargestellte Lösung war somit, obwohl bereits besser als das Ergebnis der (RBS)-Heuristik, nicht optimal. Demzufolge ist die optimale Lösung sogar um ca. 27,5 % besser als die Lösung der (RBS)-Heuristik.

Tabelle 5.5 zeigt Rechenergebnisse von acht Datensätzen, die sich nach Anwendung des *HTFSPD-Branch&Bound*-Verfahrens (jeweils mit den drei vorgestellten Suchstrategien) ergaben. Dabei handelt es sich bei den ersten fünf, um die gleichen Datensätze, die schon in Tabelle 5.4 zur Analyse der Lösungen des (ZDF)- und des (EBF)-Modells herangezogen wurden. Zwecks Vergleichbarkeit wurde auch hier $f_{templ} = 1000$ und $templ^{min} = 2000$ zugrunde gelegt und die Laufzeit auf fünf Stunden begrenzt.

Tabelle 5.5: Vergleich der Rechenergebnisse der (RBS)-Heuristik mit denen des *HTFSPD-Branch&Bound*-Verfahrens (mit $f_{templ} = 1000$)

| Daten- satz | RBS | | BB-PFDN | | | BB-DFPN | | | BB-PDR | | |
|----------------|-----------|-----------|---------|----------|-----------|---------|----------|-----------|--------|----------|--|
| | C_{max} | C_{max} | Gap | Zeit | C_{max} | Gap | Zeit | C_{max} | Gap | Zeit | |
| <i>dat3</i> | 548 | 548 | 0.00 | < 1.00 | 548 | 0.00 | < 1.00 | 548 | 0.00 | < 1.00 | |
| <i>dat5</i> | 846 | 846 | 0.00 | < 1.00 | 846 | 0.00 | < 1.00 | 846 | 0.00 | < 1.00 | |
| <i>dat8</i> | 704 | 614 | 0.00 | < 1.00 | 614 | 0.00 | < 1.00 | 614 | 0.00 | < 1.00 | |
| <i>dat10</i> | 713 | 656 | 0.00 | < 1.00 | 656 | 0.00 | < 1.00 | 656 | 0.00 | < 1.00 | |
| <i>dat12</i> | 931 | 885 | 0.00 | 135.66 | 885 | 0.00 | 138.71 | 885 | 0.00 | 144.75 | |
| <i>dat15</i> | 871 | 843 | 0.00 | 267.43 | 843 | 0.00 | 415.49 | 843 | 0.00 | 277.62 | |
| <i>dat20</i> | 920 | 920 | 46.74 | 451.71* | 920 | 100.00 | 18000.00 | 920 | 100.00 | 4908.52 | |
| <i>dat30</i> | 1311 | 1306 | 54.82 | 18000.00 | 1311 | 100.00 | 18000.00 | 1311 | 100.00 | 18000.00 | |

*Abbruch aufgrund des Speicherplatzbedarfs

Die Ergebnisse zeigen, dass das *HTFSPD-Branch&Bound*-Verfahren für die Datensätze *dat3* - *dat15* optimale Lösungen liefert. Zudem liegen die benötig-

ten Laufzeiten für diese Datensätze in einem angemessenen Rahmen. Bei den Datensätzen *dat20* und *dat30* kommt es dagegen aufgrund der Laufzeitbegrenzung oder eines zu hohen Speicherbedarfs zum Abbruch des Verfahrens. Beim Vergleich der Ergebnisse in Bezug auf die drei Suchstrategien sind nur geringfügige Unterschiede zu erkennen. Allgemein schneidet die (BB-PFDN)-Strategie, trotz des hohen Speicherbedarfs, jedoch am besten ab. Interessant ist, dass die (BB-PFDN)-Strategie bei Datensatz *dat20* bereits nach wenigen Minuten aufgrund von zu hohem Speicherbedarf abbricht, wohingegen bei Datensatz *dat30* erst das Erreichen der Laufzeitbegrenzung einen Abbruch auslöst. Der Grund hierfür liegt voraussichtlich darin, dass die Streuung der Wabendaten bei Datensatz *dat20* wesentlich höher ist, als bei Datensatz *dat30*.

Im Gegensatz zu den Lösungen des (ZDF)- und des (EBF)- Modells können die Lösungen des *HTFSPD-Branch&Bound*-Verfahrens zur Analyse des Optimierungspotentials der (RBS)-Heuristik durchaus herangezogen werden. Während die (RBS)-Heuristik für die kleinen Datensätze (*dat3* und *dat5*) noch die optimalen Lösungen liefert, so gilt dies für die größeren Datensätze nicht zwingend. Hier weicht die (RBS)-Heuristik zum Teil um ca. 12,8 % von der Optimallösung ab (*dat8*).

Bislang wurde immer von $f_{templ} = 1000$ ausgegangen. Die Vermutung liegt jedoch nahe, dass das Optimierungspotential bei kleineren Temperaturlücken noch höher ist. Die in Tabelle 5.6 zusammengefassten Rechenergebnisse für $f_{templ} = 600$ belegen diese Aussage. Besonders auffallend ist, dass nun auch für die Datensätze *dat20* und *dat30* deutliche Lösungsverbesserungen gegenüber der (RBS)-Heuristik erzielt werden. So führt das *HTFSPD-Branch&Bound*-Verfahren bei Anwendung der (BB-PFDN)-Strategie für Datensatz *dat20* zu einer um ca. 15,66 % besseren Lösung als die (RBS)-Heuristik. Zudem sprechen diese Ergebnisse noch eher dafür, die (BB-PFDN)-Strategie den anderen beiden Suchstrategien vorzuziehen.

Zusammenfassend lässt sich festhalten, dass das derzeit im Betrieb der Dillinger Hüttenwerke verwendete Lösungsverfahren noch Optimierungspotential offenbart. Mit Hilfe des vorgestellten *HTFSPD-Branch&Bound*-Verfahrens konnte dieses Potential zwar aufgezeigt werden, jedoch ist das Verfahren nicht für den Realbetrieb geeignet, da es über zu lange Laufzeiten verfügt. Im Real-

Tabelle 5.6: Vergleich der Rechenergebnisse der (RBS)-Heuristik mit denen des *HTFSPD-Branch&Bound*-Verfahrens (mit $f_{templ} = 600$)

| Daten- satz | RBS | | BB-PFDN | | BB-DFPN | | | BB-PDR | | |
|----------------|-----------|-----------|---------|----------|-----------|--------|----------|-----------|--------|----------|
| | C_{max} | C_{max} | Gap | Zeit | C_{max} | Gap | Zeit | C_{max} | Gap | Zeit |
| <i>dat3</i> | 417 | 417 | 0.00 | < 1.00 | 417 | 0.00 | < 1.00 | 417 | 0.00 | < 1.00 |
| <i>dat5</i> | 846 | 846 | 0.00 | < 1.00 | 846 | 0.00 | < 1.00 | 846 | 0.00 | < 1.00 |
| <i>dat8</i> | 612 | 543 | 0.00 | < 1.00 | 543 | 0.00 | < 1.00 | 543 | 0.00 | < 1.00 |
| <i>dat10</i> | 703 | 605 | 0.00 | < 1.00 | 605 | 0.00 | < 1.00 | 605 | 0.00 | < 1.00 |
| <i>dat12</i> | 768 | 723 | 0.00 | 21.70 | 723 | 0.00 | 26.91 | 723 | 0.00 | 22.39 |
| <i>dat15</i> | 798 | 778 | 0.00 | 688.89 | 778 | 0.00 | 871.05 | 778 | 0.00 | 712.61 |
| <i>dat20</i> | 900 | 759 | 37.55 | 514.07* | 883 | 100.00 | 18000.00 | 769 | 100.00 | 18000.00 |
| <i>dat30</i> | 1303 | 1197 | 44.44 | 18000.00 | 1303 | 100.00 | 18000.00 | 1296 | 100.00 | 18000.00 |

*Abbruch aufgrund des Speicherplatzbedarfs

betrieb treten i. d. R. Datensätze mit einer Wabenanzahl von ca. 20 - 80 Waben auf. Die Laufzeit des verwendeten Lösungsverfahrens sollte dabei nicht mehr als 10 Minuten betragen. Kapitel 7 widmet sich daher unter Verwendung heuristischer Ansätze der Entwicklung eines derartigen Verfahrens. Zuvor müssen in Kapitel 6 jedoch noch einige Problemerkweiterungen und -varianten eingeführt werden.

Kapitel 6

Problemerweiterungen und -varianten

Bei der in Kapitel 5 im Rahmen des HTFSPD beschriebenen Problemstellung ist bewusst von einigen im Realbetrieb vorliegenden Situationen und Optionen abstrahiert worden. Diese zusätzlichen Problemeigenschaften werden in diesem Kapitel jeweils separat eingeführt und untersucht. Unterkapitel 6.1 stellt die Verband-Fahrweise vor, die darin besteht, Waben nicht nur hintereinander, sondern auch nebeneinander in den Ofen zu fahren. Anschließend widmet sich Unterkapitel 6.2 der *Hollomon*-Fahrweise, bei der Änderungen an den Wärmebehandlungsvorgaben in Betracht gezogen werden. Schließlich befasst sich Unterkapitel 6.3 mit den speziellen Auswirkungen der Einbeziehung bestimmter dem Ofen vor- und nachgelagerter Prozesse.

6.1 Die Verband-Fahrweise

Bisher wurden nur Ofenbelegungen in Betracht gezogen, bei denen Waben aufeinanderfolgend durch den Ofen fahren. Für den RHO3 besteht jedoch auch die Möglichkeit der Verband-Fahrweise. Dabei werden mehrere Waben parallel zueinander angeordnet und in einem Verband zusammengefasst. Dieser Verband wird dann wie eine einzelne Wabe behandelt, d. h. die Waben innerhalb

des Verbands fahren gleichzeitig in den Ofen ein und auch gleichzeitig wieder aus dem Ofen aus.

Abbildung 6.1 stellt eine beispielhafte Verband-Fahrweise grafisch dar. In diesem Beispiel werden die Waben 2 und 3 sowie 4 und 5 jeweils in einem Verband gefahren. Wie man sieht, ergibt sich die Länge eines Verbands aus der maximalen Länge ihrer enthaltenen Waben.

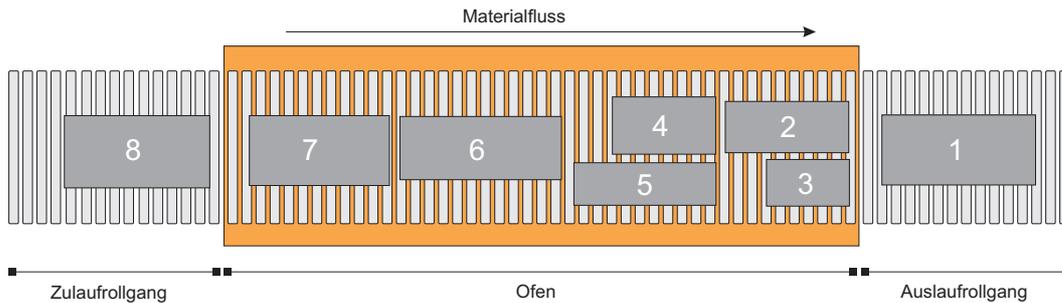


Abbildung 6.1: Die Verband-Fahrweise

Allgemein müssen für einem Verband zugehörige Waben die folgenden Bedingungen erfüllt sein:

- Die Summe der Wabenbreiten darf nicht größer sein als die zulässige Ofen- bzw. die Ofenrollgangsbreite.
- Die Wärmebehandlungstemperaturen (Ofen- und Zieltemperatur) der Waben müssen identisch sein.
- Die Ofenverweilzeiten der Waben müssen identisch sein.

Zudem sei nochmals explizit darauf hingewiesen, dass ein Verband als Wabe bzw. Pseudowabe gehandhabt wird. Das schließt mit ein, dass für diese Pseudowabe dieselben Regeln in Bezug auf Wartezeiten und Temperaturlücken gelten, wie für eine einzelne Wabe. Zudem folgt daraus, dass eine verschachtelte Anordnung, wie in Abbildung 6.2 dargestellt, nicht erlaubt ist.

Es liegt auf der Hand, dass die Verband-Fahrweise dem Ofendurchsatz nur förderlich sein kann, da durch sie der Ofenraum besser genutzt wird. Dennoch kommt diese Methode in der betrieblichen Praxis bei den Dillinger Hüttenwerken nur sehr selten zum Einsatz. Zudem ist dies auch hier wieder stark vom jeweiligen Ofensteuerer abhängig. Speziell aus diesem Grund, ist ein ein-

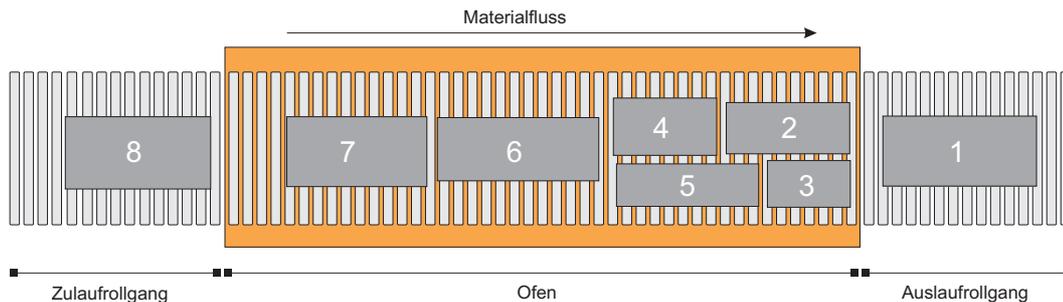


Abbildung 6.2: Unerlaubte Verband-Fahrweise

heitliches Optimierungsverfahren erforderlich, welches die Verband-Fahrweise gezielt einbezieht.

6.1.1 Ein Optimierungsmodell für die Verband-Fahrweise

Im Folgenden wird ein Optimierungsmodell für die Verband-Fahrweise formuliert. Im Prinzip könnte das Nebeneinanderfahren von Waben im *HTFSPD-Branch&Bound*-Verfahren integriert werden. Diese simultane Variante würde jedoch vermutlich auf Kosten stark ansteigender Laufzeiten gehen. Die Optimierung der Verbandbildung kann aber auch einfach der Reihenfolgebildung vorgelagert erfolgen. Diese sequentielle Vorgehensweise wird voraussichtlich weniger Laufzeit bedürfen als die simultane Variante. Bedenken bezüglich eines möglichen Optimalitätsverlusts mit Hinsicht auf das Gesamtproblem sind unbegründet, da ohnehin nur Waben mit identischen Ofentemperaturen und Ofenverweilzeiten zu einem Verband zusammengefasst werden dürfen.

Es stellt sich nun zunächst die Frage, wie eine optimale Lösung für die Verband-Fahrweise überhaupt aussieht. Bedenkt man, dass die Länge eines Verbands durch die maximale Länge der darin enthaltenen Waben bestimmt wird, so kann daraus gefolgert werden, dass die Lösung anzustreben ist, welche die Gesamtlänge der betrachteten Waben minimiert. Das im Folgenden vorgestellte Optimierungsmodell für die Verband-Fahrweise (VF) beruht daher auf dieser Zielsetzung.

Das Modell erfordert die Einführung der Parameter:

- b^{Ofen} : Ofenbreite,
 b^{min} : Mindestabstand zwischen Waben innerhalb eines Verbands,
 b_j : Breite von Wabe j ,
 $g_{jj'} = \begin{cases} 1, & \text{wenn Wabe } j \text{ und Wabe } j' \text{ prinzipiell im gleichen} \\ & \text{Verband angeordnet werden dürfen} \\ 0, & \text{sonst;} \end{cases}$

Es sei darauf hingewiesen, dass $g_{jj'}$ vorab berechnet werden muss. Dabei gelten zwei Waben als prinzipiell im gleichen Verband anordbar, wenn sie über identische Ofentemperaturen und Ofenverweilzeiten verfügen.

Die Entscheidungsvariablen des (VF)-Modells sind wie folgt definiert:

- $x_{jv} = \begin{cases} 1, & \text{wenn Wabe } j \text{ Verband } v \text{ zugeordnet wird} \\ 0, & \text{sonst;} \end{cases}$
 l_v^{vb} : Länge von Verband v

Das (VF)-Modell wird folgendermaßen formuliert:

$$(VF) \quad \min \sum_{v=1}^n l_v^{vb} \quad (6.1)$$

u.d.N.

$$x_{jv} + x_{j'v} \leq 1 + g_{jj'} \quad \forall v, j, j' = 1, \dots, n \quad (6.2)$$

$$\sum_{j=1}^n x_{jv} = 1 \quad \forall v = 1, \dots, n \quad (6.3)$$

$$\sum_{j=1}^n x_{jv} \cdot (b_j + b^{min}) \leq b^{Ofen} + b^{min} \quad \forall v = 1, \dots, n \quad (6.4)$$

$$x_{jv} \cdot l_j \leq l_v^{vb} \quad \forall j, v = 1, \dots, n \quad (6.5)$$

$$x_{jv} \in \{0, 1\} \quad \forall j, v = 1, \dots, n \quad (6.6)$$

$$l_v^{vb} \geq 0 \quad \forall v = 1, \dots, n \quad (6.7)$$

Mit der Zielfunktion (6.1) wird die Gesamtlänge der betrachteten Waben minimiert. Nebenbedingung 6.2 stellt sicher, dass zwei Waben j und j' nur zusammen einem Verband v zugeordnet werden können, wenn dies auch prinzipiell möglich ist. Mit Nebenbedingung 6.3 wird garantiert, dass jede Wabe einem

Verband zugeordnet sein muss, wobei hier ein Verband auch nur aus einer Wabe bestehen kann. Für die Einhaltung der zulässigen Ofenbreite sorgt daraufhin Nebenbedingung 6.4. Mit Nebenbedingung 6.5 werden die Längen der Verbände ermittelt, die sich aus den maximalen Längen der jeweils enthaltenen Waben ergeben. Schließlich bilden die Nebenbedingungen 6.6 und 6.7 die Binär- und Nichtnegativitätsbedingungen für die entsprechenden Entscheidungsvariablen.

Um Aussagen über das Optimierungspotential der Verband-Fahrweise treffen zu können, werden wieder die bereits in Kapitel 5.4.2 untersuchten Datensätze herangezogen. Der ILOG CPLEX Optimizer kommt für das (VF)-Modell zu den in Tabelle 6.1 gelisteten Ergebnissen.

Tabelle 6.1: Rechenergebnisse des (VF)-Modells

| Daten- satz | vorher | | (VF)-Modell | | | | |
|----------------|--------|--------|-------------|----------|--------|-------|-----------|
| | n | l^V | n^V | n^{V*} | l^V | Gap | Zeit |
| <i>dat3</i> | 3 | 45408 | 3 | 0 | 45408 | 0.00 | < 1.00 |
| <i>dat5</i> | 5 | 48000 | 4 | 1 | 37725 | 0.00 | < 1.00 |
| <i>dat8</i> | 8 | 51098 | 8 | 0 | 51098 | 0.00 | < 1.00 |
| <i>dat10</i> | 10 | 68042 | 9 | 1 | 58287 | 0.00 | < 1.00 |
| <i>dat12</i> | 12 | 80865 | 10 | 2 | 73780 | 0.00 | < 1.00 |
| <i>dat15</i> | 15 | 111909 | 13 | 2 | 92936 | 0.00 | 1.56 |
| <i>dat20</i> | 20 | 81376 | 16 | 4 | 64806 | 0.00 | 8.92 |
| <i>dat30</i> | 30 | 171104 | 22 | 8 | 127656 | 11.14 | 10973.89* |

*Abbruch aufgrund des Speicherplatzbedarfs

Die erste Spalte verweist wieder auf den jeweiligen Datensatz. Die Spalten 2 und 3 zeigen die Anzahl der Waben und die Summen der Wabengängen der jeweiligen Datensätze vor Ausführung des (VF)-Modells. Anhand der Spalte 4 und 5 ist dann zu erkennen, welche Verbandanzahlen (n^V) sich nach Ausführung des (VF)-Modells ergaben und wie viele davon mehr als eine Wabe enthalten (n^{V*}). Die daraus resultierenden Verbandlängensummen l^V sind in Spalte 6 abgebildet. Es folgen schließlich wiederum Angaben zu dem jeweiligen Gap und den Laufzeiten.

Es zeigt sich, dass bei den Datensätzen *dat3* und *dat8* kein Verband aus mehr als einer Wabe besteht und somit keine „echten“ Verbände erzeugt werden konnten. Bei allen anderen Datensätzen war dies jedoch möglich. So ließen sich

bei Datensatz *dat30* beispielsweise 16 Waben zu acht Verbänden zusammenfassen und die ursprüngliche Wabengängensumme so um ca. 25,39 % reduzieren. Bei Betrachtung des *Gap* zeigt sich jedoch, dass diese Lösung nicht zwingend optimal sein muss, da der Solver aufgrund der Speicherkapazität nach ca. drei Stunden abbrechen musste. Die Datensätze *dat3* bis *dat20* konnten zwar in wenigen Sekunden optimal gelöst werden, doch scheint das (VF)-Modell bei größeren Datensätzen zu sehr langen Laufzeiten zu tendieren.

Es stellt sich nun die Frage, wie die Verband-Fahrweise auf die Reihenfolgebildung wirkt bzw. wie hoch das Optimierungspotential ist. Hierzu werden die „neuen“ Datensätze, die sich aus den Ergebnissen des (VF)-Modells ergaben, nochmals mit dem *HTFSPD-Branch&Bound*-Verfahren gelöst. Die Ergebnisse (für $f_{templ} = 1000$) sind in Tabelle 6.2 zusammengefasst.

Tabelle 6.2: Vergleich der Rechenergebnisse bei Verband-Fahrweise
(mit $f_{templ} = 1000$)

| Daten- satz | RBS | | BB-PFDN | | | |
|----------------|-----------|-------------|-----------|-------------|------------|----------|
| | C_{max} | C_{max}^V | C_{max} | C_{max}^V | <i>Gap</i> | Zeit |
| <i>dat5</i> | 846 | 836 | 846 | 836 | 0.00 | < 1.00 |
| <i>dat10</i> | 713 | 703 | 656 | 626 | 0.00 | < 1.00 |
| <i>dat12</i> | 931 | 921 | 885 | 865 | 0.00 | 4.88 |
| <i>dat15</i> | 871 | 786 | 843 | 754 | 0.00 | 16.88 |
| <i>dat20</i> | 920 | 880 | 920 | 877 | 50.51 | 829.94* |
| <i>dat30</i> | 1311 | 1230 | 1306 | 1004 | 42.23 | 4905.02* |

*Abbruch aufgrund des Speicherplatzbedarfs

Untersucht wurden die (RBS)-Heuristik und das (BB-PFDN)-Verfahren. Zum besseren Vergleich sind die Lösungswerte ohne Verband-Fahrweise C_{max} nochmals neben den sich bei Vorliegen der Verband-Fahrweise ergebenden Lösungen C_{max}^V abgebildet. Man stellt fest, dass die Verband-Fahrweise bei manchen Datensätzen nur zu eher kleinen, bei anderen dagegen zu erheblichen Verbesserungen führt. So z. B. bei Datensatz *dat15*, für den sich mit der Verband-Fahrweise nochmals ca. 10,56 % gegenüber der optimalen Lösung ohne Verband-Fahrweise einsparen lassen. Auch ein Vergleich der (RBS)-Heuristik und des (BB-PFDN)-Verfahrens ist interessant. So ist die Lösung des (BB-PFDN)-Verfahrens mit Verband-Fahrweise bei Datensatz *dat30* um ca. 20,32 % besser als die entsprechende Lösung der (RBS)-Heuristik.

Mit Tabelle 6.3 wird die Untersuchung um die Ergebnisse ergänzt, die sich bei $f_{templ} = 600$ ergeben. Auch hier wurde sich wiederum auf die (RBS)-Heuristik und das (BB-PFDN)-Verfahren beschränkt.

Tabelle 6.3: Vergleich der Rechenergebnisse bei Verband-Fahrweise (mit $f_{templ} = 600$)

| Daten- satz | RBS | | BB-PFDN | | | Zeit |
|----------------|-----------|-------------|-----------|-------------|--------|----------|
| | C_{max} | C_{max}^V | C_{max} | C_{max}^V | Gap | |
| <i>dat5</i> | 846 | 836 | 846 | 836 | 0.00 | < 1.00 |
| <i>dat10</i> | 703 | 621 | 605 | 535 | 0.00 | < 1.00 |
| <i>dat12</i> | 768 | 748 | 723 | 650 | 0.00 | < 1.00 |
| <i>dat15</i> | 798 | 758 | 778 | 697 | 0.00 | 54.49 |
| <i>dat20</i> | 900 | 860 | 759 | 632 | 0.00 | 571.67 |
| <i>dat30</i> | 1303 | 1220 | 1197 | 877 | 100.00 | 3459.73* |

*Abbruch aufgrund des Speicherplatzbedarfs

Die Ergebnisse zeigen allgemein noch deutlichere Verbesserungen als zuvor. So führt die Verband-Fahrweise nun auch bei den Datensätzen *dat10* und *dat12* zu größeren Einsparungen. Zudem ist die Lösung des (BB-PFDN)-Verfahrens mit Verband-Fahrweise bei Datensatz *dat30* nun sogar um ca. 28,11 % besser als die entsprechende Lösung der (RBS)-Heuristik.

Aus den Ergebnissen der Untersuchung lässt sich die Bedeutung des gezielten Einsatzes der Verband-Fahrweise ableiten. Da jedoch das Lösen des (VF)-Modells für größere Datensätze in recht hohen Laufzeiten resultiert, ist dieses Modell für den Einsatz im Realbetrieb der Dillinger Hüttenwerke nicht geeignet. Eine Möglichkeit es dennoch zu verwenden, besteht darin, eine maximal zulässige Laufzeit zu definieren und die Berechnungen einfach nach Ablauf dieser Zeit abzubrechen. Als Lösung dient dann das bis dahin beste gefundene Ergebnis. Eine andere Möglichkeit wäre die Verwendung eines heuristischen Ansatzes, dem sich das folgende Unterkapitel widmet.

6.1.2 Eine Heuristik für die Verband-Fahrweise

Ein heuristischer Ansatz für die Verband-Fahrweise lässt sich relativ leicht und intuitiv entwickeln. Zunächst erscheint es sinnvoll, alle Waben, die prinzipiell in

einem Verband zusammengefasst werden können, zu gruppieren. Hierzu kann, wie schon beim (VF)-Modell, der Parameter $g_{jj'}$ herangezogen werden. Aus einer Wabengruppe, die nur aus einer einzelnen Wabe besteht, kann dann logischerweise kein „echter“ Verband gebildet werden. Für die folgenden Schritte kommen daher nur noch die Wabengruppen in Betracht, die mehr als eine Wabe enthalten.

In jeder Wabengruppe werden die Waben nun ihrer Länge nach absteigend sortiert. Anschließend wird zunächst aus der ersten (längsten) Wabe ein Verband gebildet. Daraufhin wird von oben herab geprüft, ob sich die nächste Wabe diesem Verband aufgrund der Breitensumme zuordnen lässt. Ist dies nicht der Fall, so wird diese Prüfung mit der folgenden Wabe fortgesetzt. Sobald eine Wabe gefunden wurde, die nicht zu breit ist, wird sie dem Verband zugeordnet. Nach der Prüfung der letzten (kürzesten) Wabe beginnt die beschriebene Vorgehensweise wieder von vorne, wobei nur noch Waben, die noch nicht zu einem Verband zusammengefasst wurden, in Betracht zu ziehen sind. Im letzten Schritt muss für alle Verbände schließlich noch die jeweilige Verbandlänge ermittelt werden.

Die beschriebene Vorgehensweise wird im Folgenden als (VF)-Heuristik bezeichnet und durch den in Algorithmus 6 dargestellten Pseudocode zusammengefasst. Die Länge eines einzelnen Verbandes v wird wiederum mit l_v^{vb} bezeichnet. Analog verweist b_v^{vb} auf die entsprechende Breite von Verband v . Zudem werden die in einer Wabengruppe w enthaltenen Waben in der Menge wg_w zusammengefasst. Mehrere Wabengruppen werden dann wiederum in der Menge WG vereinigt werden, sodass $WG = \{wg_1, \dots, wg_{|W|}\}$ gilt. Als Ergebnis liefert die (VF)-Heuristik zum einen eine Verbandmenge V , mit $V = \{vb_1, \dots, vb_{|V|}\}$, wobei vb_v die Menge der Verband v zugeordneten Waben repräsentiert. Zum anderen wird auch hier wieder die Verbandlängensumme l^V ausgegeben, da diese schließlich den zu minimierenden Zielfunktionswert bildet.

Im Folgenden wird die (VF)-Heuristik mit dem (VF)-Modell verglichen. Hierzu werden die bereits mit dem (VF)-Modell gelösten Datensätze erneut mit der (VF)-Heuristik nachberechnet. Tabelle 6.4 zeigt die erzielten Rechenergebnisse. Als Vergleichsbasis sind die entsprechenden Lösungen des (VF)-Modells mit aufgeführt. Wie man unschwer erkennen kann, sind die Zielfunktionswerte

Algorithmus 6 (VF)-Heuristik

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , b_j , $g_{jj'}$ mit $j, j' \in J$, b^{min}

Ensure: Verbandmenge V , l^V

```

1: for  $j = 1, \dots, n$  do
2:   if  $j \notin WG$  then
3:     bilde Wabengruppe  $wg_{|WG|+1}$  mit  $wg_{|WG|+1} \leftarrow wg_{|WG|+1} \cup \{j\}$ 
4:   end if
5:   for  $j' = 1, \dots, n$  do
6:     if  $g_{jj'} = 1$  and  $j' \notin WG$  then
7:        $wg_{|WG|+1} \leftarrow wg_{|WG|+1} \cup \{j'\}$ 
8:     end if
9:   end for
10:   $WG \leftarrow WG \cup \{wg_{|WG|+1}\}$ 
11: end for
12: for  $w = 1, \dots, |WG|$  do
13:   if  $|wg_w| = 1$  then
14:     bilde Verband  $vb_{|V|+1}$  mit  $vb_{|V|+1} = wg_w$ 
15:      $V \leftarrow V \cup \{vb_{|V|+1}\}$ 
16:   else
17:     sortiere  $wg_w$  nach  $l$  absteigend
18:     for  $p = 1, \dots, |wg_w|$  do
19:       if  $wg_{w_p} \notin V$  then
20:         bilde Verband  $vb_{|V|+1}$  mit  $vb_{|V|+1} \leftarrow vb_{|V|+1} \cup \{wg_{w_p}\}$ 
21:          $b_{|V|+1}^{vb} = b_{wg_{w_p}}$ 
22:         for  $q = p + 1, \dots, |wg_w|$  do
23:           if  $b_{|V|+1}^{vb} + b_{wg_{w_q}} + b^{min} \leq b^{Ofen}$  and  $wg_{w_q} \notin V$  then
24:              $vb_{|V|+1} \leftarrow vb_{|V|+1} \cup \{wg_{w_q}\}$ 
25:              $b_{|V|+1}^{vb} \stackrel{\pm}{=} b_{wg_{w_q}}$ 
26:           end if
27:         end for
28:          $V \leftarrow V \cup \{vb_{|V|+1}\}$ 
29:          $l_v^{vb} = \max\{l_j | j \in vb_v\}$ 
30:       end if
31:     end for
32:   end if
33: end for
34:  $l^V = \sum_{v \in V} l_v^{vb}$ 

```

l^V der (VF)-Heuristik mit denen des (VF)-Modells für alle Datensätze identisch. Zudem liegen die Laufzeiten der (VF)-Heuristik alle unter einer Sekunde. Damit sind diese insbesondere bei den größeren Datensätzen, wesentlich niedriger als die entsprechenden Laufzeiten des (VF)-Modells (vgl. z. B. Datensatz

Tabelle 6.4: Vergleich der Rechenergebnisse der (VF)-Heuristik mit denen des (VF)-Modells

| Daten- satz | (VF)-Modell | | | (VF)-Heuristik | |
|----------------|-------------|-------|-----------|----------------|--------|
| | l^V | Gap | Zeit | l^V | Zeit |
| <i>dat3</i> | 45408 | 0.00 | < 1.00 | 45408 | < 1.00 |
| <i>dat5</i> | 37725 | 0.00 | < 1.00 | 37725 | < 1.00 |
| <i>dat8</i> | 51098 | 0.00 | < 1.00 | 51098 | < 1.00 |
| <i>dat10</i> | 58287 | 0.00 | < 1.00 | 58287 | < 1.00 |
| <i>dat12</i> | 73780 | 0.00 | < 1.00 | 73780 | < 1.00 |
| <i>dat15</i> | 92936 | 0.00 | 1.56 | 92936 | < 1.00 |
| <i>dat20</i> | 64806 | 0.00 | 8.92 | 64806 | < 1.00 |
| <i>dat30</i> | 127656 | 11.14 | 10973.89* | 127656 | < 1.00 |

*Abbruch aufgrund des Speicherplatzbedarfs

dat30). Die Ergebnisse beweisen zwar nicht, dass die (VF)-Heuristik immer die optimale Lösung liefert, jedoch scheint sie allein aufgrund der geringen Laufzeiten für den Einsatz im Realbetrieb der Dillinger Hüttenwerke besser geeignet zu sein als das (VF)-Modell.

6.2 Die *Hollomon*-Fahrweise

Der Begriff *Hollomon* tauchte bereits in Kapitel 3.3.3 im Zusammenhang mit dem Hollomon-Jaffe-Parameter auf. Dieser Zeit-Temperatur-Parameter wird auch bei den Dillinger Hüttenwerken zur Charakterisierung einer Anlasswärmebehandlung herangezogen. Wie bereits erwähnt, errechnet sich der Hollomon-Jaffe-Parameter HJP_j einer Wabe j grundsätzlich als eine Formel der Zieltemperatur $wbht_j^{Ziel}$ und der Haltedauer hd_j . Bei den Dillinger Hüttenwerken gehen allerdings noch von der Wabendicke d_j abhängige Aufheiz- und Abkühlraten als Einflussgrößen mit ein.

Die Berechnung des Hollomon-Jaffe-Parameters lässt sich grafisch anhand von Abbildung 6.3 veranschaulichen. Bei dem abgebildeten Koordinatensystem ist auf der x-Achse die Zeit und auf der y-Achse die Temperatur abgetragen. Die rote Linie kennzeichnet die Aufheizphase, die grüne Linie die Haltephase und die blaue Linie dementsprechend die Abkühlphase.

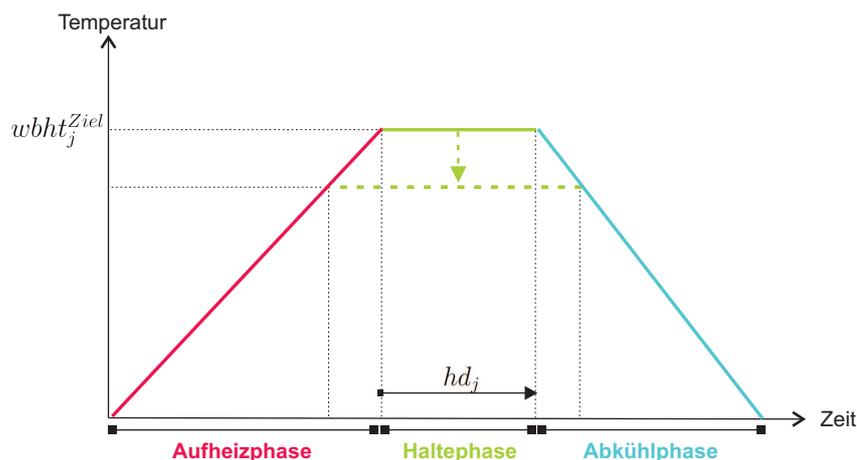


Abbildung 6.3: Veranschaulichung der Berechnung des Hollomon-Jaffe-Parameters

Für den Wärmebehandlungseffekt sind insbesondere die Ziel- bzw. Haltetemperatur $wbht_j^{Ziel}$ und die Haltedauer hd_j entscheidend. Aus diesen beiden Parametern lässt sich der Hollomon-Jaffe-Parameter ableiten. Er stellt eine dimensionslose Größe dar. Für diese Arbeit von besonderem Interesse ist die Tatsache, dass sich der gleiche Hollomon-Jaffe-Parameter bzw. Wärmebehandlungseffekt durch mehrere Temperatur-Zeit-Kombinationen einstellen lässt. Bezogen auf Abbildung 6.3 bedeutet dies, dass man den gleichen, sich aus den ursprünglichen Vorgaben für $wbht_j^{Ziel}$ und hd_j ergebenden Hollomon-Jaffe-Parameter beispielsweise auch mit niedrigerer Zieltemperatur bei längerem Halten erzielen kann (vgl. grün gestrichelte Linie).

Die Idee der *Hollomon-Fahrweise* (HF) besteht darin, sich die Variierbarkeit der Wärmebehandlungsparameter für die Reihenfolgeoptimierung zunutze zu machen. Bisher wird dieser Aspekt bei den Dillinger Hüttenwerken noch gar nicht bzw. nur indirekt in Betracht gezogen. Über die Wärmebehandlungsvorgaben entscheidet die metallurgische Abteilung, wobei diese Entscheidungen noch vor dem eigentlichen Produktionsbeginn getroffen werden und demnach nicht an eine Reihenfolgeoptimierung gekoppelt sind. Mit dem Hintergedanken, dennoch möglichst hohe Durchsätze erzielen zu wollen, wird das Temperatur-Zeit-Verhältnis immer so gewählt, dass die aus metallurgischer Sicht maximal erlaubte Temperatur realisiert und demnach die erforderliche Haltedauer bzw. die daraus resultierende Ofenverweilzeit so gering wie möglich gehalten wird. Anhand eines kleinen Beispiels soll gezeigt werden, dass dies jedoch nicht zwin-

gend optimal für den Ofendurchsatz sein muss.

Das folgende Beispiel beruht auf den in Tabelle 6.5 zusammengefassten Ausgangsdaten. Anhand der in den Klammern dargestellten Werte lässt sich auf die möglichen Temperatur-Zeit-Kombinationen schließen. So ergeben beispielsweise bei Wabe 1 die Kombinationen 600 °C Haltetemperatur mit 233 min Ofenverweilzeit, 590 °C Haltetemperatur mit 291 min Ofenverweilzeit und 580 °C Haltetemperatur mit 392 min Ofenverweilzeit den gleichen Hollomon-Jaffe-Parameter wie die ursprüngliche Vorgabe von 610 °C Haltetemperatur mit 200 min Ofenverweilzeit. Es sei an dieser Stelle darauf hingewiesen, dass grundsätzlich nur Zieltemperatursenkungen in 10 °C Schritten betrachtet werden.

Tabelle 6.5: Ein Beispieldatensatz für die *Hollomon*-Fahrweise

| j | $wbht_j^{Ziel}$ | ovz_j | HJP_j | l_j |
|-----|---------------------|---------------------|---------|-------|
| 1 | 610 (600, 590, 580) | 200 (233, 291, 392) | 17.35 | 13000 |
| 2 | 610 (600, 590, 580) | 200 (233, 291, 392) | 17.35 | 13000 |
| 3 | 630 (620, 610, 600) | 300 (324, 365, 434) | 17.61 | 10000 |
| 4 | 630 (620, 610, 600) | 300 (324, 365, 434) | 17.61 | 10000 |
| 5 | 640 (630, 620, 610) | 250 (270, 305, 362) | 17.76 | 15000 |

Abbildung 6.4 stellt eine beispielhafte Ofenbelegung des eingeführten Beispiels dar. Wie bereits aus ähnlichen vorherigen Abbildungen bekannt, repräsentieren die grün abgebildeten Werte die Ofeneinfahr- und ausfahrzeiten der jeweiligen Wabe. In der dargestellten Situation befinden sich die Waben 1, 2, 3, und 4 aktuell im Ofen. Wabe 5 muss dagegen zunächst warten, da der Ofen noch nicht ausreichend Restkapazität bereitstellt. So kann Wabe 5 erst einfahren, sobald Wabe 1 den Ofen verlässt. Bei dieser Konstellation ergibt sich eine Gesamtfertigstellungszeit von 470 min. Es sei darauf hingewiesen, dass hierbei alle Wärmebehandlungsparameter den ursprünglichen Vorgaben entsprechen.

Abbildung 6.5 zeigt nun für das gleiche Beispiel eine Ofenbelegung, bei der die *Hollomon*-Fahrweise in Betracht gezogen wird. Dabei wurde die Haltetemperatur von Wabe 5 um 10 °C auf 630 °C herabgesetzt. Infolgedessen hat sich die Ofenverweilzeit für Wabe 5 um 20 min auf 270 min erhöht. Man sieht jedoch, dass nun keine Temperaturlücke zwischen Wabe 4 und 5 mehr erforderlich ist und Wabe 5 demnach früher in den Ofen einfahren kann. Da die Ofenverweilzeit

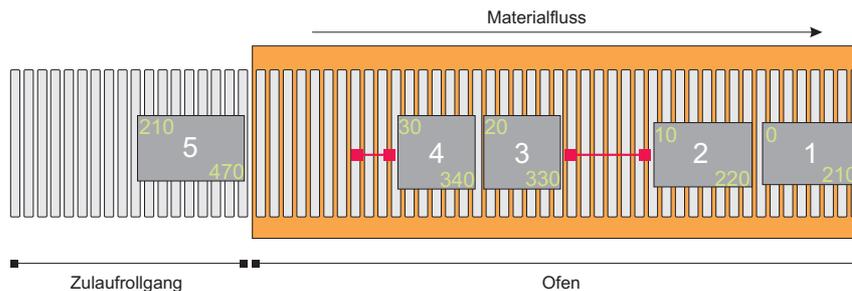


Abbildung 6.4: Beispielhafte Ausgangssituation zur Veranschaulichung der Idee der *Hollomon*-Fahrweise

von Wabe 5 trotz dessen Erhöhung immer noch kleiner ist als die Ofenverweilzeit von Wabe 4, muss Wabe 5 jedoch noch kurz (30 min) warten. Dennoch führt die Erhöhung der Ofenverweilzeit von Wabe 5 nicht zu einer Erhöhung der Gesamtbearbeitungszeit. Im Gegenteil, der Gesamtfertigstellungzeitpunkt dieser Konstellation ist dann mit 350 min um 25,5% geringer als bei der vorherigen Situation ohne Betrachtung der *Hollomon*-Fahrweise.

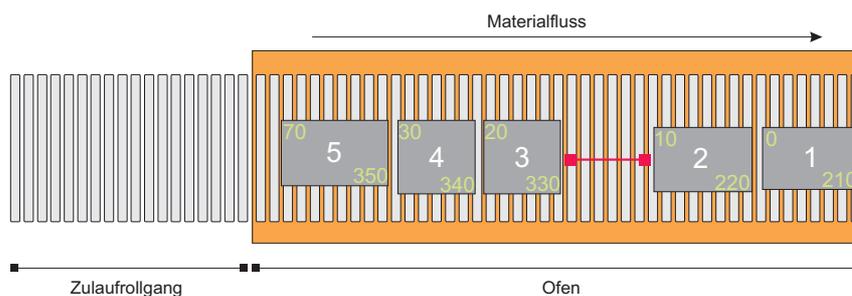


Abbildung 6.5: Beispielhafte Situation bei Berücksichtigung der *Hollomon*-Fahrweise

Das Beispiel zeigt, dass mit der *Hollomon*-Fahrweise prinzipiell Durchsatzsteigerungen erzielt werden können. Vor der Entwicklung eines geeigneten Optimierungsverfahrens, bietet es sich jedoch zunächst an, die unterschiedlichen Temperatur-Zeit-Kombinationen näher zu untersuchen.

6.2.1 Analyse der Temperatur-Zeit-Kombinationen

Anhand des Beispieldatensatzes aus Tabelle 6.5 lässt sich bereits grob erahnen, wie sich die Ofenverweilzeiten bei sinkenden Temperaturen und konstantem *Hollomon*-Parameter entwickeln. Bei genauerer Beobachtung ist zu erkennen,

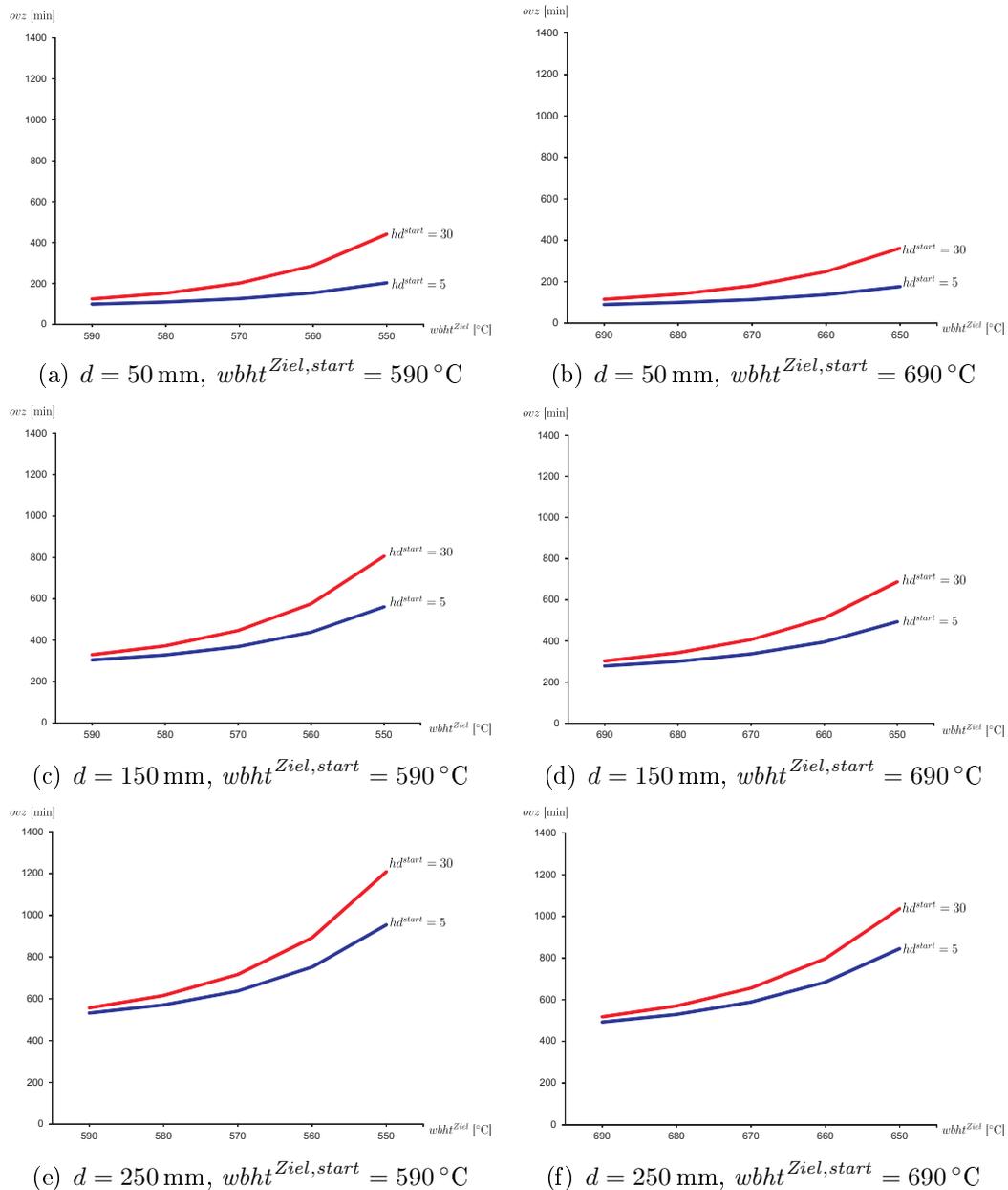
dass die Ofenverweilzeiten unterschiedlich schnell ansteigen. So ist sie nach einer Temperatursenkung von $30\text{ }^{\circ}\text{C}$ gegenüber der Ausgangstemperatur bei Wabe 5 um ca. 44,8 % gestiegen. Die gleiche Temperatursenkung hat z.B. bei Wabe 1 jedoch zu einem Ofenverweilzeitanstieg von 96 % geführt.

Wie bereits erläutert, kommen nur die Haltetemperatur, die Haltezeit und die Wabendicke für die unterschiedlichen Temperatur-Zeit-Entwicklungen in Frage. Welche Auswirkungen diese Parameter auf das Verhalten der Temperatur-Zeit-Kombinationen ausüben, zeigen die in Abbildung 6.6 zusammengefassten Untersuchungen.

Es wurden jeweils drei verschiedene Wabendicken von 50, 150 und 250 mm zugrunde gelegt. Hinsichtlich der Haltedauer wurden Startvorgaben (hd^{start}) von 5 und 30 min herangezogen. Die entsprechenden Startvorgaben für die Haltetemperaturen ($wbht^{Ziel,start}$) lagen bei 590 und $690\text{ }^{\circ}\text{C}$. Von diesen ausgehend wurden Temperatursenkungen von bis zu $40\text{ }^{\circ}\text{C}$ untersucht.

Bei allen Teilabbildungen 6.6(a) - 6.6(f) fällt direkt auf, dass die beiden Starthaldedauern sehr unterschiedlich auf die Temperatursenkungen reagieren. So liegt der aus Temperatursenkungen von $40\text{ }^{\circ}\text{C}$ resultierende durchschnittliche Ofenverweilzeitanstieg mit 356,4 min bei einer Starthaldedauer von 30 min deutlich über den entsprechenden 538,4 min bei einer Starthaldedauer von 5 min. An dieser Stelle sei darauf hingewiesen, dass alle Ofenverweilzeitsteigerungen nahezu ausschließlich auf den jeweiligen Haltedaueranstieg zurückfallen, da sich die entsprechenden Erwärmdauern nur geringfügig erhöhen. Des Weiteren ist den Untersuchungen zu entnehmen, dass die Ofenverweilzeiten bei dickeren Waben wesentlich stärker als bei dünneren Waben erhöht werden müssen, um den ursprünglich vorgegebenen Wärmebehandlungseffekt zu erzielen. So erfordern Temperatursenkungen von $40\text{ }^{\circ}\text{C}$ bei 50 mm dicken Waben im Mittel einen Ofenverweilzeitanstieg von 295,13 min, wohingegen bei 150 bzw. 250 mm dicken Waben bereits 636,42 bzw. 1010,66 min erforderlich sind. Ferner fällt bei paarweiser Betrachtung der Teilabbildungen 6.6(a) mit 6.6(b), 6.6(c) mit 6.6(d) und 6.6(e) mit 6.6(f) auf, dass die Starttemperatur scheinbar auch einen (wenn auch nur relativ geringen) Einfluss auf den resultierenden Ofenverweilzeitanstieg ausübt. So liegt dieser bei einer Temperatursenkung von $40\text{ }^{\circ}\text{C}$ und einer Starttemperatur von $590\text{ }^{\circ}\text{C}$ mit durchschnittlich 695,22 min

Abbildung 6.6: Temperatur-Zeit-Entwicklung für unterschiedliche Ausgangsparameter (bei konstantem *Hollomon*-Parameter)



leicht über dem entsprechenden Ofenverweilzeitanstieg von 599,59 min bei einer Starttemperatur von 690 °C.

Allgemein lässt sich festhalten, dass der mit einer Temperatursenkung gleichzeitig verbundene Ofenverweilzeitanstieg umso geringer ist,

- je geringer die Starthaltedauer,
- je dünner die Wabe und
- je höher die Starttemperatur ist.

Waben mit diesen Eigenschaften eignen sich somit prinzipiell am ehesten für den Einsatz der *Hollomon*-Fahrweise bzw. die Manipulation der ursprünglichen Wärmebehandlungsvorgaben.

6.2.2 Ein Optimierungsverfahren für die *Hollomon*-Fahrweise

Anders als bei der Verband-Fahrweise lässt sich die Optimierung der *Hollomon*-Fahrweise nicht ohne Verlust der Optimalitätsgarantie von der Reihenfolgeoptimierung entkoppeln. Eine optimale Problemlösung ist demnach nur durch ein integriertes bzw. simultanes Optimierungsverfahren möglich. Aus diesem Grund wird im Folgenden das in Kapitel 5.4.2 vorgestellte *HTFSPD-Branch&Bound*-Verfahren um den Aspekt der *Hollomon*-Fahrweise erweitert und dementsprechend als *HTFSPD-HF-Branch&Bound*-Verfahren bezeichnet.

Zunächst gilt es, die Verzweigungsregeln anzupassen. Beim ursprünglichen *HTFSPD-Branch&Bound*-Verfahren entsteht eine Relaxation eines bestimmten Teilproblems nur durch Fixierung einer weiteren Position in der Reihenfolge. Bei Berücksichtigung der *Hollomon*-Fahrweise müssen jedoch noch weitere Verzweigungsmöglichkeiten in Betracht gezogen werden. Im Prinzip kommen hierfür alle möglichen Temperatur-Zeit-Kombinationen der jeweiligen Waben in Frage. Abbildung 6.7 veranschaulicht derartige Verzweigungen an einem Beispiel. Bei dem oberen Knoten handelt es sich um den Ursprungsknoten, bei dem alle Waben noch unfixiert sind. Die Knoten darunter stellen mögliche Verzweigungen dar, wobei nur jeweils Wabe 1 zusätzlich fixiert wird. Da diese jedoch (entsprechend der *Hollomon*-Fahrweise) über variierbare Wärmebehandlungsparameter verfügt, bildet jede mögliche Temperatur-Zeit-Kombination prinzipiell ein eigenes Teilproblem.

Mit Hilfe des an der Wabenummer abgetragenen Indexes z kann auf die jeweilige Kombinationsmöglichkeit bzw. auf die entsprechenden Wärmebehand-

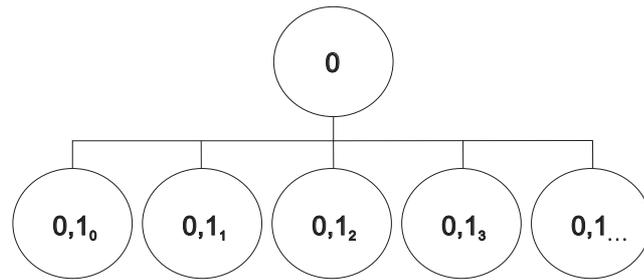


Abbildung 6.7: Verzweigungsmöglichkeiten eines Knotens aufgrund verschiedener Temperatur-Zeit-Kombinationen

lungsparameter geschlossen werden. Für die Haltetemperatur $wbht_{jz}^{Ziel}$ gilt:

$$wbht_{jz}^{Ziel} = wbht_j^{Ziel,start} - z \cdot 10. \quad (6.8)$$

Somit verweist z auf die Anzahl der Temperatursenkungen gegenüber der Starttemperatur. Die dieser Haltetemperatur entsprechende Ofentemperatur wird analog mit $wbht_{jz}^{Ofen}$ und die resultierende Ofenverweilzeit mit ovz_{jz} gekennzeichnet.

Durch die aufgrund der Berücksichtigung der *Hollomon*-Fahrweise neu entstandenen Verzweigungsmöglichkeiten können sich Speicherbedarf und Laufzeit des Verfahrens drastisch erhöhen. Um diesen Effekt etwas abzdämpfen, empfiehlt es sich, nur eine begrenzte Anzahl z^{max} möglicher Temperatur-Zeit-Kombinationen zu betrachten. Wie die Temperatur-Zeit-Verläufe aus Abbildung 6.6 zeigen, liegen die Ofenverweilzeiten bei $z = 4$ teilweise schon über dem Doppelten ihres Ausgangswerts. Die Vermutung liegt nahe, dass derartige Änderungen an den ursprünglichen Wärmebehandlungsparametern wohl nicht zu Lösungsverbesserungen führen werden, sodass eine Wahl von $z^{max} \leq 3$ begründet erscheint. Weiterhin eignet sich die Änderung von Wärmebehandlungsparametern grundsätzlich nur, wenn sich dadurch vorherige Temperaturlücken einsparen oder verkleinern lassen. Aus diesem Grund können Temperatur-Zeit-Kombinationen mit Temperaturen, die unter der Minimaltemperatur der betrachteten Waben ($wbht_{min}^{Ofen}$) liegen, ohne Bedenken vernachlässigt werden.

Neben dem *Branching* muss allerdings auch das *Bounding* leicht angepasst werden, da die Verwendung der ursprünglichen unteren Schrankenberechnung das durch die *Hollomon*-Fahrweise mögliche Verbesserungspotential eventuell verschleiern könnte. Diese Schrankenermittlung berücksichtigt bei der Berech-

nung der Länge der fiktiven Wabe l_j (siehe Algorithmus 3) nämlich nur eine mögliche Temperatur (Startvorgabe) pro unverplanter Wabe. Da bei der *Holomon*-Fahrweise jedoch mehrere Temperaturen in Frage kommen, besteht die Möglichkeit, dass Konstellationen existieren, die zu einem kleineren l_j führen, was wiederum einen kleineren unteren Schrankenwert zur Folge haben könnte. Da daraus gegebenenfalls ein zu frühes Ausloten von Teilproblemen resultieren würde, muss der ursprünglichen unteren Schrankenberechnung eine Temperaturanpassung bezüglich der unverplanten Waben vorgeschaltet werden.

Die beim *HTFSPD-HF-Branch&Bound*-Verfahren für die untere Schrankenberechnung vorgeschlagene Temperaturanpassung ist in Algorithmus 7 beschrieben. Sie bezieht sich auf alle unverplanten Waben, deren ursprünglich vorgegebenen Ofentemperaturen ($wbht_{unvp_{p_0}}^{Ofen}$) größer sind als die, der an letzter Position der fixierten Reihenfolge befindlichen Wabe ($wbht_{folge_{n'}}^{Ofen}$). Die Temperaturen dieser Waben werden dann so angepasst, dass die Temperaturdifferenz zu $wbht_{folge_{n'}}^{Ofen}$ minimal ist (Zeile 3).

Algorithmus 7 Temperaturanpassung für die untere Schrankenberechnung des *HTFSPD-HF-Branch&Bound*-Verfahrens

Require: Wabenmenge J , $wbht_{jz}^{Ofen}$, ovz_{jz} , l_j , mit $j \in J$, z^{max} , $folge'$, n' , $unvp$

Ensure: $wbht_{unvp_p}^{Ofen}$

- 1: **for** $p = 1, \dots, |unvp|$ **do**
- 2: **if** $wbht_{unvp_{p_0}}^{Ofen} > wbht_{folge_{n'}}^{Ofen}$ **then**
- 3: $wbht_{unvp_p}^{Ofen} = \max\{wbht_{folge_{n'}}^{Ofen}, wbht_{unvp_{p_z^{max}}}^{Ofen}\}$
- 4: **end if**
- 5: **end for**

Durch die beschriebene Temperaturanpassung wird gewährleistet, dass bei der anschließenden unteren Schrankenberechnung (speziell bei der Berechnung von l_j) nur die tatsächlich mindestens erforderlichen Temperaturlücken herangezogen werden. Mit Ausnahme dieser vorgeschalteten Temperaturanpassung entsprechen die weiteren Schritte zur Ermittlung des unteren Schrankenwerts beim *HTFSPD-HF-Branch&Bound*-Verfahren denselben Schritten wie beim ursprünglichen *HTFSPD-Branch&Bound*-Verfahren (siehe Algorithmus 3).

Der Ablauf des *HTFSPD-HF-Branch&Bound*-Verfahrens ist in Algorithmus 8

zusammengefasst.

Algorithmus 8 *HTFSPD-HF-Branch&Bound*

Require: Wabenmenge J , $wbht_{j_z}^{Ofen}$, ovz_{j_z} , l_j , mit $j \in J$, z^{max} , $wbht_{min}^{Ofen}$
 f_{templ} , $templ^{min}$, t^{min}

Ensure: C_{max}^{glob} , $folge^{glob}$, z_j^{glob}

- 1: **call** *Rule-based Sorting*
- 2: **call** *Simulation-1* und setze $C_{max}^{akt} = C_{max}$
- 3: erzeuge Ursprungsknoten k mit $folge'_k = ()$, $C_{max,k}^{lb} = 0$, $C_{max,k}^{ub} = C_{max}^{akt}$
- 4: $K \leftarrow K \cup \{k\}$
- 5: **while** $|K| \neq 0$ **do**
- 6: **if** $C_{max,K_1}^{lb} \geq C_{max}^{akt}$ **or** $open_{K_1,j} = \mathbf{true} \forall j \in \{J\}$ **then**
- 7: $K \leftarrow K \setminus \{K_1\}$
- 8: **else**
- 9: **for** $j = 1, \dots, n$ **do**
- 10: **if** $j \notin folge'_{K_1}$ **and** $open_{K_1,j} = \mathbf{false}$ **then**
- 11: **for** $z = 0, \dots, z^{max}$ **do**
- 12: **if** $wbht_{folge_{j_z}}^{Ofen} \geq wbht_{min}^{Ofen}$ **then**
- 13: erzeuge neuen Knoten k
- 14: $open_{K_1,j} = \mathbf{true}$
- 15: $folge'_k \leftarrow folge'_{K_1} \cup \{j_z\}$
- 16: berechne $C_{max,k}^{lb*}$ und $C_{max,k}^{ub}$
- 17: **if** $C_{max,k}^{lb} \geq C_{max}^{akt}$ **then**
- 18: lösche Knoten k
- 19: **else if** $C_{max,k}^{ub} < C_{max}^{akt}$ **then**
- 20: aktualisiere $folge^{akt}$ und C_{max}^{akt}
- 21: **for** $k' \in K$ **do**
- 22: **if** $C_{max,k'}^{lb} \geq C_{max}^{akt}$ **then**
- 23: $K \leftarrow K \setminus \{k'\}$
- 24: **end if**
- 25: **end for**
- 26: $K \leftarrow K \cup \{k\}$
- 27: **else**
- 28: $K \leftarrow K \cup \{k\}$
- 29: **end if**
- 30: **end if**
- 31: **end for**
- 32: **break**
- 33: **end if**
- 34: **end for**
- 35: **end if**
- 36: **end while**
- 37: $C_{max}^{glob} = C_{max}^{akt}$, $folge^{glob} = folge^{akt}$

Die gegenüber dem ursprünglichen *HTFSPD-Branch&Bound*-Verfahren neuen Verzweigungsregeln sind den Zeilen 11 und 12 zu entnehmen. Wenn folglich ein Teilproblem durch Fixierung einer bis dahin unverplanten Wabe j verzweigt werden soll, so werden alle aufgrund der Temperatur-Zeit-Kombinationen z^{max} möglichen Relaxationen direkt gebildet. Voraussetzung hierfür ist allerdings, dass die jeweilige Ofentemperatur $wbht_{folge_j z}^{Ofen}$ nicht unter die Minimaltemperatur $wbht_{min}^{Ofen}$ fällt.

Des Weiteren wird mit $C_{max,k}^{lb*}$ darauf hingewiesen, dass der ursprünglichen Berechnung von $C_{max,k}^{lb}$ (siehe Algorithmus 3) hier noch die in Algorithmus 7 beschriebene Temperaturanpassung vorzuschalten ist. Ansonsten entspricht der Ablauf des *HTFSPD-HF-Branch&Bound*-Verfahrens weitestgehend dem des *HTFSPD-Branch&Bound*-Verfahrens. Es sei jedoch noch erwähnt, dass die in den Zeilen 26 - 31 von Algorithmus 5 beschriebenen Schritte beim *HTFSPD-HF-Branch&Bound*-Verfahren wegfallen. Als Ausgabewerte kommen andererseits die Angaben z_j^{glob} hinzu, welche auf die in der optimalen Lösung gewählten Wärmebehandlungstemperaturen der jeweiligen Waben verweisen.

In Tabelle 6.6 sind Rechenergebnisse verschiedener Durchläufe des *HTFSPD-HF-Branch&Bound*-Verfahrens abgebildet, wobei aus Vergleichsgründen wiederum die Datensätze *dat3* - *dat30* herangezogen wurden. Für jeden dieser Datensätze sind jeweils drei Durchläufe durchgeführt worden, wobei sich diese durch verschiedene Vorgaben für z^{max} voneinander unterscheiden. Bezüglich der Temperaturlücken beruhen die Ergebnisse zunächst wieder auf der Annahme von $f_{templ} = 1000$. Als Suchstrategie wurde wiederum die (BB-PFDN)-Strategie verwendet.

Die Ergebnisse des *HTFSPD-HF-Branch&Bound*-Verfahrens sind in den Spalten 5 - 8 (unter BB-HF-PFDN) eingetragen. Um das rein durch die Berücksichtigung der *Hollomon*-Fahrweise entstehende Optimierungspotential abschätzen zu können, sind in den Spalten 2 - 4 (unter BB-PFDN) zusätzlich die bereits bekannten Ergebnisse des *HTFSPD-Branch&Bound*-Verfahrens aufgeführt.

Ein Vergleich der Lösungswerte zeigt, dass die Berücksichtigung der *Hollomon*-Fahrweise bei fast allen Durchläufen zu Verbesserungen geführt hat. Die mit Abstand größte konnte bei Datensatz *dat8* mit einem um 18,24 % geringeren

Tabelle 6.6: Vergleich der Rechenergebnisse bei *Hollomon*-Fahrweise
(mit $f_{templ} = 1000$)

| Daten- satz | BB-PFDN | | | z^{max} | BB-HF-PFDN | | |
|----------------|-----------|-------|----------|-----------|------------|-------|----------|
| | C_{max} | Gap | Zeit | | C_{max} | Gap | Zeit |
| <i>dat3</i> | 548 | 0.00 | < 1.00 | 1 | 548 | 0.00 | < 1.00 |
| | | | | 2 | 543 | 0.00 | < 1.00 |
| | | | | 3 | 543 | 0.00 | < 1.00 |
| <i>dat5</i> | 846 | 0.00 | < 1.00 | 1 | 846 | 0.00 | < 1.00 |
| | | | | 2 | 846 | 0.00 | < 1.00 |
| | | | | 3 | 846 | 0.00 | < 1.00 |
| <i>dat8</i> | 614 | 0.00 | < 1.00 | 1 | 595 | 0.00 | 10.39 |
| | | | | 2 | 566 | 0.00 | 32.03 |
| | | | | 3 | 502 | 0.00 | 14.43 |
| <i>dat10</i> | 656 | 0.00 | < 1.00 | 1 | 654 | 0.00 | 233.33 |
| | | | | 2 | 651 | 0.00 | 2419.18 |
| | | | | 3 | 641 | 0.00 | 5791.71 |
| <i>dat12</i> | 885 | 0.00 | 135.66 | 1 | 882 | 51.21 | 6487.72* |
| | | | | 2 | 863 | 49.94 | 3886.72* |
| | | | | 3 | 872 | 51.02 | 3784.74* |
| <i>dat15</i> | 843 | 0.00 | 267.43 | 1 | 778 | 0.00 | 12245.01 |
| | | | | 2 | 778 | 36.25 | 18000.00 |
| | | | | 3 | 771 | 35.67 | 18000.00 |
| <i>dat20</i> | 920 | 46.74 | 451.71 | 1 | 920 | 48.59 | 3415.55* |
| | | | | 2 | 920 | 49.57 | 2801.14* |
| | | | | 3 | 920 | 52.83 | 2593.81* |
| <i>dat30</i> | 1306 | 54.82 | 18000.00 | 1 | 1311 | 54.99 | 4646.29* |
| | | | | 2 | 1311 | 55.01 | 3178.75* |
| | | | | 3 | 1311 | 55.01 | 2988.67* |

*Abbruch aufgrund des Speicherplatzbedarfs

Gesamtfertigstellungszeitpunkt erzielt werden. Weiterhin fällt auf, dass gerade bei den größeren Datensätzen *dat20* und *dat30* keine Verbesserungen erzielt werden konnten. Da diese auch schon ohne Berücksichtigung der *Hollomon*-Fahrweise aufgrund des zu hohen Speicherbedarfs oder der Laufzeitbeschränkung nicht optimal gelöst werden konnten, heißt dies keineswegs, dass hier die *Hollomon*-Fahrweise kein Verbesserungspotential bietet.

Allgemein lässt sich hinsichtlich der Laufzeiten und des Speicherbedarfs ein enormer Anstieg im Vergleich zu dem *HTFSPD-Branch&Bound*-Verfahren beobachten. Insbesondere mit steigendem z^{max} wird dieser Effekt besonders deut-

lich. Bei Betrachtung der Ergebnisse bezüglich der unterschiedlichen Startvorgaben für z^{max} fällt zudem auf, dass der Lösungswert für Datensatz *dat12* bei $z^{max} = 3$ schlechter ist als bei $z^{max} = 2$. Dies ist höchstwahrscheinlich dem bei $z^{max} = 3$ wesentlich höheren Speicherbedarf geschuldet. Wie man anhand der Ergebnisse für Datensatz *dat8* sieht, können Temperatursenkungen von 30 °C jedoch zu weiteren Verbesserungen führen und sollten demnach nicht grundsätzlich ausgeschlossen werden.

Der Vollständigkeit wegen wurden die geschilderten Untersuchungen unter Vorgabe von $f_{templ} = 600$ erneut durchgeführt. Aus Tabelle 6.7 sind die erzielten Rechenergebnisse zu entnehmen.

Tabelle 6.7: Vergleich der Rechenergebnisse bei *Hollomon*-Fahrweise
(mit $f_{templ} = 600$)

| Daten- satz | BB-PFDN | | | z^{max} | BB-HF-PFDN | | |
|----------------|-----------|-------|----------|-----------|------------|-------|-----------|
| | C_{max} | Gap | Zeit | | C_{max} | Gap | Zeit |
| <i>dat3</i> | 417 | 0.00 | < 1.00 | 1 | 417 | 0.00 | < 1.00 |
| | | | | 2 | 417 | 0.00 | < 1.00 |
| | | | | 3 | 417 | 0.00 | < 1.00 |
| <i>dat5</i> | 846 | 0.00 | < 1.00 | 1 | 846 | 0.00 | < 1.00 |
| | | | | 2 | 836 | 0.00 | < 1.00 |
| | | | | 3 | 836 | 0.00 | < 1.00 |
| <i>dat8</i> | 543 | 0.00 | < 1.00 | 1 | 484 | 0.00 | 1.34 |
| | | | | 2 | 467 | 0.00 | 8.24 |
| | | | | 3 | 467 | 0.00 | 25.65 |
| <i>dat10</i> | 605 | 0.00 | < 1.00 | 1 | 605 | 0.00 | 413.82 |
| | | | | 2 | 574 | 0.00 | 1300.53 |
| | | | | 3 | 574 | 0.00 | 3479.74 |
| <i>dat12</i> | 723 | 0.00 | 21.70 | 1 | 674 | 0.00 | 1011.76 |
| | | | | 2 | 674 | 0.00 | 3700.81 |
| | | | | 3 | 674 | 0.00 | 11614.31 |
| <i>dat15</i> | 778 | 0.00 | 688.89 | 1 | 778 | 31.24 | 18000.00 |
| | | | | 2 | 768 | 30.99 | 14962.41* |
| | | | | 3 | 768 | 48.59 | 9307.03* |
| <i>dat20</i> | 759 | 37.55 | 514.07 | 1 | 792 | 33.13 | 13022.69* |
| | | | | 2 | 823 | 43.03 | 6149.36* |
| | | | | 3 | 842 | 42.99 | 6493.14* |
| <i>dat30</i> | 1197 | 44.44 | 18000.00 | 1 | 1303 | 53.16 | 3621.55* |
| | | | | 2 | 1303 | 53.95 | 3054.55* |
| | | | | 3 | 1303 | 53.95 | 2379.23* |

*Abbruch aufgrund des Speicherplatzbedarfs

Die Ergebnisse bestätigen die zuvor beobachteten Zusammenhänge größtenteils. Auch hier ließ sich die größte Lösungsverbesserung bei Datensatz *dat8* realisieren. Die Verbesserung ist hier mit ca. 14% zwar etwas geringer als bei $f_{templ} = 1000$, jedoch sind die Lösungen der anderen Datensätze zum Teil entsprechend besser (vgl. Datensatz *dat10* und *dat12*). Auch das Laufzeit- und Speicherbedarfverhalten ist den Ergebnissen aus Tabelle 6.6 ähnlich. Bemerkenswert ist hier jedoch, dass sich Datensatz *dat12* nun im Gegensatz zu vorher für alle betrachteten z^{max} exakt lösen ließ. Weiterhin interessant ist die Tatsache, dass $z^{max} = 3$ kein einziges Mal zu besseren Lösungen als $z^{max} = 2$ bzw. $z^{max} = 1$ geführt hat. Für eine Verallgemeinerung dieser Beobachtung liegt hier jedoch eine zu kleine Stichprobe vor.

Es bleibt festzuhalten, dass sich das rein aufgrund der Reihenfolgeoptimierung bestehende Optimierungspotential durch Berücksichtigung der *Hollomon*-Fahrweise noch zusätzlich erhöht. Daher ist es prinzipiell zu empfehlen, die *Hollomon*-Fahrweise in die Reihenfolgebildung zu integrieren. Die durchgeführten Untersuchungen haben jedoch auch gezeigt, dass sich dadurch die Komplexität der Gesamtproblematik im Vergleich zum ursprünglichen HTFSPD drastisch erhöht. Ein exaktes Lösungsverfahren, wie das vorgestellte *HTFSPD-HF-Branch&Bound*-Verfahren, scheint daher nicht für den betrieblichen Einsatz geeignet zu sein.

6.3 Einbeziehung des gesamtlogistischen Prozesses

Bislang lag der Fokus sowohl bei der einführenden Beschreibung des HTFSPD (siehe Kapitel 5) als auch bei den in diesem Kapitel erläuterten Erweiterungen bzw. Varianten (siehe Unterkapitel 6.1 und 6.2) ausschließlich auf dem Ofen. Da die Reihenfolgebildung des Ofens in der Praxis jedoch auch von vor- und nachgelagerten Prozessen beeinflusst wird, soll die Betrachtungsweise im Folgenden um diese Prozesse erweitert werden. Dies führt, im Gegensatz zu den Problemerkweiterungen durch die Verband- oder die *Hollomon*-Fahrweise, nicht zu zusätzlichem Optimierungspotential, sondern zu Situationen, aus denen bislang noch nicht betrachtete Wartezeiten für die Waben resultieren. Folglich

bedarf es hier entsprechenden Anpassungen an der bisherigen Materialflusssimulation des Ofens (vgl. Algorithmus 2).

6.3.1 Die Stapelproblematik als vorgelagerter Prozess

Bislang wurde davon ausgegangen, dass die Wabe, die als nächste in den Ofen einfahren soll, immer direkt zur Verfügung steht. In der Realität ist dies aufgrund der bereits in Kapitel 2.6 angedeuteten Stapelproblematik allerdings nicht immer der Fall.

Wie die in Kapitel 5.1 dargelegte Aufnahme aus dem Dillinger Walzwerk (Abbildung 5.1) zeigt, sind die Waben in nahe des Ofens angelegten Stapeln übereinander gelagert. Aufgrund des beschränkten Lagerplatzes ist dies unumgänglich, führt allerdings dazu, dass der Kran jeweils nur die oberste Wabe eines Stapels direkt aufnehmen kann. Häufig entspricht diese jedoch nicht der nächsten benötigten Wabe. In diesen Fällen muss der Kran die benötigte Wabe zunächst freilegen, d. h. alle darüber liegenden Waben umstapeln.

Abbildung 6.8 veranschaulicht beispielhaft einen derartigen Stapel. Die Nummern an den Waben deuten auf die jeweilige Reihenfolgeposition hinsichtlich des Ofeneintritts hin. Demnach wird die grün gekennzeichnete Wabe als nächste benötigt. Um diese freizulegen, muss der Kran die darüber liegenden, rot gekennzeichneten Waben zunächst umstapeln. Da der Transport über Magnete erfolgt, muss jede Wabe einzeln umgestapelt werden. Je nachdem wie viele Waben dies betrifft und wie die Krankapazität es zulässt, können derartige Umstapelaktionen sehr zeitintensiv sein, weshalb es sie zu minimieren gilt. Entscheidend ist hierbei jeweils die Wahl des geeigneten Zielstapels für die umzustapelnden Waben.

In der Literatur sind derartige Probleme nicht unbekannt, wenn auch eher mit der Verladung von Containern an entsprechenden Container-Terminals in Zusammenhang gebracht. De Castilho und Daganzo [26] haben sich als eine der ersten mit der Entwicklung von Strategien beschäftigt, um die Anzahl erforderlicher Umstapelaktionen an Hafenanlagen sowohl im Vorhinein abzuschätzen als auch im Nachhinein zu minimieren. Auch in der späteren Arbeit

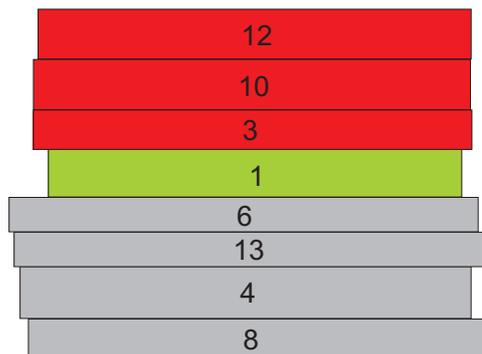


Abbildung 6.8: Ein beispielhafter Stapel mit drei umzustapelnden Waben

von Kim [55] wird eine Methode zur Schätzung der minimalen Anzahl an Umstapelaktionen vorgeschlagen, auf dessen Basis Kim und Hong [56] schließlich eine Heuristik entwickelt haben. Des Weiteren stellen Kim und Hong [56] einen Branch&Bound-Ansatz vor, der allerdings nur für sehr kleine Probleminstanzen in annehmbarer Zeit exakte Lösungen liefert. Caserta et al. [19] bezeichnen die Problemstellung als *Blocks Relocation Problem* (BRP) und schlagen eine auf dynamischer Programmierung basierende Lösungsheuristik vor. Caserta und Voß [17, 18] präsentieren zudem zwei Varianten, mit denen sich das Verfahren weiter verbessern lässt. Caserta et al. [16] entwickeln jedoch noch einen anderen Lösungsansatz. Hierbei handelt es sich um ein iteratives, zufallsbasiertes Verfahren, das auf der Idee einer einfachen *Greedy*-Heuristik basiert, welche später noch näher vorgestellt wird.

In einer kürzlich erschienen Arbeit stellen Forster und Bortfeldt [36] ein Baumsuchverfahren für das BRP vor und vergleichen dessen Ergebnisse mit denen der bereits erwähnten Heuristiken für verschiedene Datensätze. Die Auswertung ergab, dass das Baumsuchverfahren im Durchschnitt gegenüber den Heuristiken von Kim und Hong [56], Caserta et al. [19], Caserta und Voß [17, 18] und Caserta et al. [16] jeweils entsprechend um 47,3 %, 9,6 %, 5,4 % und 0,1 % bessere Lösungen liefert.

Die wenigen Arbeiten aus der Literatur, die sich explizit mit Umstapelproblemen in der Stahlindustrie beschäftigen, beziehen sich nahezu alle auf das von Tang et al. [84] eingeführte *Slab Stack Shuffling Problem* (SSSP). Das Hauptaugenmerk liegt hierbei auf dem Brammenlager, in dem die Brammen vor der Einleitung in den Walzprozess (bzw. in die entsprechenden Öfen) zwischen-

gelagert werden. Auch hier liegen die Brammen aus Platzgründen i. d. R. in Stapeln übereinander. Im Gegensatz zum BRP ist beim SSSP für die Brammen jedoch keine eindeutige Entnahmereihenfolge gegeben. Stattdessen kann an jeder Reihenfolgeposition aus einer bestimmten Kandidatenmenge gewählt werden. Das Optimierungsziel besteht dann darin, die Entnahmereihenfolge so zu wählen, dass die Anzahl der resultierenden Umstapelvorgänge minimiert wird. In den letzten Jahren wurden verschiedene Varianten und Lösungsverfahren für das SSSP veröffentlicht. Hierzu zählen z. B. die Arbeiten von Tang et al. [85], Srinivas und Tiwari [79], Tang und Ren [86] sowie von Cheng und Tang [21].

Das SSSP weist neben der Vorgabe der Entnahmereihenfolge noch weitere grundlegende Unterschiede gegenüber dem BRP auf. So werden beim SSSP die eigentlichen Umstapelaktionen, im Sinne des Transports einer Bramme von einem auf einen anderen Stapel, gar nicht explizit berücksichtigt. Stattdessen werden vereinfachende Annahmen getroffen, wie beispielsweise, dass alle umzustapelnden Brammen nach der Entnahme der benötigten Bramme immer wieder direkt zurück auf ihren Ausgangsstapel abgelegt werden. Ein anderes Beispiel ist die Annahme, dass immer genügend freie Plätze vorhanden sind, um die umzustapelnden Brammen abzulegen, wobei zusätzlich angenommen wird, dass diese dann nicht ein weiteres Mal umgestapelt werden müssen.

Die beim SSSP getroffenen Annahmen sind teilweise sehr realitätsfern. Die Idee, die Entnahmereihenfolge so zu gestalten, dass an jeder Position mehrere Brammen als Kandidaten in Frage kommen ist allerdings ein sinnvoller Ansatz, da in der Praxis gerade bei Brammenlagern i. d. R. viele Brammen nahezu identisch sind. Für den Durchsatz an den Walzgerüsten ist es dann gegebenenfalls irrelevant, in welcher Reihenfolge diese Brammen entnommen werden. Derartige Entnahmegruppen werden beim BRP nicht in Betracht gezogen. Zudem trifft auch das BRP eine Annahme bezüglich der Umstapelvorgänge. So wird davon ausgegangen, dass der Kran immer nur die Container umstapelt, die über dem als nächstes benötigten Container liegen. Diese Annahme ist jedoch weitaus realitätsnäher als die beim SSSP getroffenen Annahmen.

Um den Literaturüberblick abzuschließen sei noch auf die Arbeit von Wang et al. [95] hingewiesen, die sich mit der Optimierung der Schiffsverladung von

Stahlblechen auseinandersetzt. Interessant ist, dass hierin sowohl die Wahl der Entnahmereihenfolge als auch der genauen Zielstapel für die umzustapelnden Bleche Berücksichtigung finden und damit Aspekte des SSSP und des BRP miteinander vereint werden.

Hinsichtlich der in dieser Arbeit betrachteten Problemstellung besteht jedoch im Zusammenhang mit der Umstapelproblematik ein in der Literatur bisher noch nicht explizit betrachteter Sachverhalt: Rein in Bezug auf den Ofen, führt ein zu langes Umstapeln nämlich schlicht und ergreifend zu zusätzlichen Wartezeiten. Somit stehen die Umstapelproblematik und die Durchsatzoptimierung am Ofen in direkter Verbindung zueinander.

Um die durch zu langes Umstapeln eventuell erforderlichen Wartezeiten bei der Reihenfolgeoptimierung zu berücksichtigen, muss die bestehende Materialflusssimulation des Ofens (vgl. Algorithmus 2) erweitert werden. Verschiedene Beobachtungen und Messungen im entsprechenden Betrieb der Dillinger Hüttenwerke haben ergeben, dass der Kran für eine Transport- bzw. Umstapelaktion ca. zwischen 2 und 4 min benötigt. Dabei wurden die folgenden Prozessschritte betrachtet:

1. Fahrt zu dem Stapel, der die als nächstes zum Ofen zu transportierende Wabe enthält.
2. Aufnahme der obersten Wabe.
3. Falls diese der benötigten Wabe entspricht:
 - (a) Fahrt zum Ofen,
 - (b) Ablage der Wabe auf dem Zulaufrollgang des Ofens.
4. Falls diese nicht der benötigten Wabe entspricht:
 - (a) Fahrt zu einem anderen Stapel,
 - (b) Ablage der Wabe auf dem Stapel.

Für die Dauer einer Transport- bzw. Umstapelaktion sind somit hauptsächlich die Fahrtwege sowie die Zeiten für das Aufnehmen und Ablegen der Waben verantwortlich. Dabei haben letztere den wesentlich größeren Anteil an der Gesamtdauer, da die entsprechenden Stapel i. d. R. relativ nahe beieinander liegen. Aus diesem Grund wird im Folgenden von einer konstanten Umstapeldauer $t^u = 3$ min ausgegangen.

Mit der erläuterten Abschätzung der Dauer für das Umstapeln würde die in Abbildung 6.8 grün gekennzeichnete Wabe folglich 12 min benötigen, bis sie auf dem Zulaufrollgang des Ofens liegt. Dies hätte in Anbetracht der Tatsache, dass allein die Mindestwartezeit einer Wabe (t^{min}) bereits 10 min beträgt, wenn überhaupt, nur einen sehr kleinen Einfluss auf den Ofendurchsatz. Müssten jedoch beispielsweise zunächst 20 Waben umgestapelt werden, um die benötigte Wabe freizulegen, so wäre diese erst nach 63 min bereit für die Ofeneinfahrt. Dies hätte höchstwahrscheinlich größere Auswirkungen auf den Ofendurchsatz und damit auch auf die optimale Reihenfolge. Andererseits sollte auch ungeachtet des Ofens grundsätzlich das Ziel verfolgt werden, die zeitliche Belastung der Krane so gering wie möglich zu halten.

Die vorgeschlagene Ermittlung der durch Umstapeln entstehenden Wartezeiten reicht jedoch allein noch nicht aus, um diese in der Ofensimulation zu integrieren. Da sich nach jeder Umstapelaktion die Lager- bzw. Stapelsituation ändert, müsste man im Prinzip vorhersehen können, auf welche Zielstapel jeweils umgestapelt wird. Hierfür gibt es bei den Dillinger Hüttenwerken derzeit jedoch keine Vorgaben, sodass diese Entscheidungen allein dem Kranfahrer überlassen sind. Da diesem teilweise elementare Informationen (wie z. B. die dem aktuellen Ofenprogramm entsprechenden Reihenfolgepositionen) fehlen, ist davon auszugehen, dass nicht immer hinsichtlich zukünftiger Umstapelaktionen optimal umgestapelt wird. Um diese Situation zu verbessern und gleichzeitig das Vorhersehen der Umstapelaktionen zu ermöglichen, wird im Folgenden ein Zielstapelvorschlagssystem entwickelt.

Prinzipiell bietet es sich an, ein für das BRP entwickeltes Lösungsverfahren zu verwenden. Mit Hinblick auf die Integration des Verfahrens in der Simulation, sollte hierbei insbesondere auf möglichst geringe Laufzeiten geachtet werden. Grund hierfür ist, dass sowohl das *HTFSPD-Branch&Bound*-Verfahren als auch der später in Kapitel 7 vorgestellte genetische Algorithmus ein sehr häufiges Aufrufen der Simulation erfordern. Als besonders schnell haben sich die Heuristik von Kim und Hong [56] und die in Caserta et al. [16] verwendete *Greedy*-Heuristik erwiesen. Da letztere, wie in Caserta et al. [15] veröffentlichte Rechenergebnisse zeigen, bessere Lösungen liefert, fällt die Wahl schließlich auf dieses Verfahren. Es wird als *Max-Min*-Heuristik bezeichnet und beruht auf den folgenden Regeln:

Wenn Wabe j umgestapelt werden muss, so setze sie:

1. auf den Stapel $s \in S$, für den s^{fmin} minimal ist, falls mindestens ein Stapel existiert, für den $s^{fmin} > fpos_j$ gilt,
2. andernfalls auf den Stapel(platz) $s \in S$, für den $|s| = 0$ ist, falls mindestens ein Stapel(platz) existiert, für den $|s| = 0$ gilt,
3. andernfalls auf den Stapel $s \in S$, für den s^{fmin} maximal ist.

Dabei bezeichnet s einen bestimmten Stapel aus der Menge aller im Lager befindlichen Stapel S . Zudem wird mit $fpos_j$ auf die Position verwiesen, den Wabe j in der Reihenfolge *folge* einnimmt. Diesbezüglich bezeichnet s^{fmin} die kleinste Reihenfolgeposition der in Stapel s enthaltenen Waben.

Die Vorgehensweise der *Max-Min*-Heuristik soll anhand eines kleinen Beispiels verdeutlicht werden. Angenommen, es befinden sich 15 Waben im Lager, die entsprechend der Darstellung in Abbildung 6.9 (oben) angeordnet sind. Auch hier repräsentieren die Zahlen an den Waben wiederum die jeweilige Reihenfolgeposition. Wabe 1 ist daher als erste zu entnehmen. Um sie freizulegen,

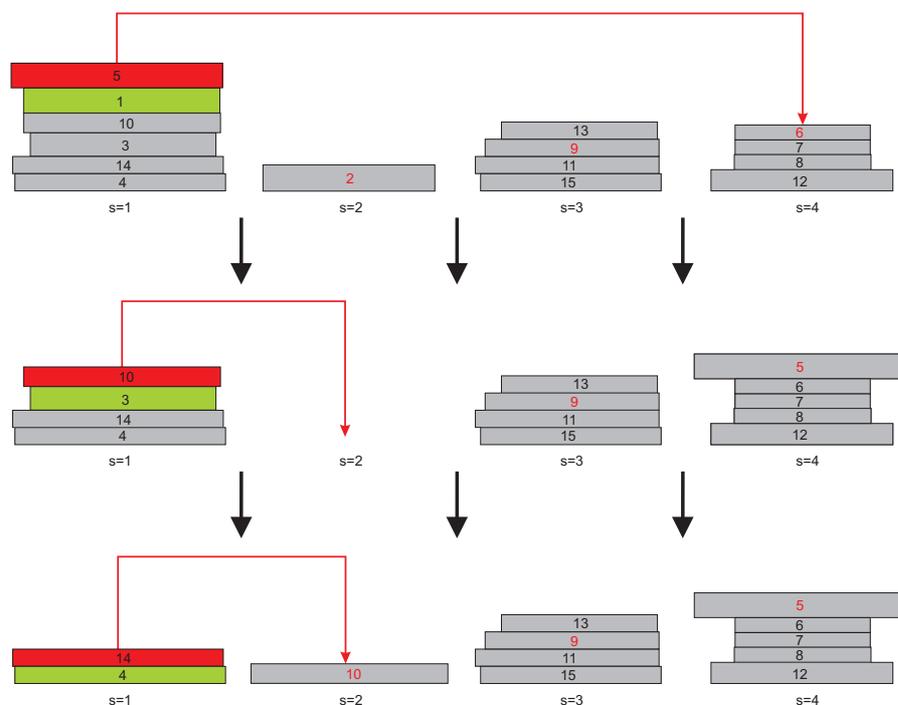


Abbildung 6.9: Beispielhafte Veranschaulichung der *Max-Min*-Heuristik

muss Wabe 5 zunächst umgestapelt werden. Hier greift die erste Regel der *Max-Min*-Heuristik, da mit $s = 3$ und $s = 4$ zwei Stapel existieren, für die $s^{fmin} > fpos_j$ gilt. Da s^{fmin} für Stapel 4 kleiner ist als für Stapel 3, wird Stapel 4 als Zielstapel für Wabe 5 gewählt. Anschließend können die Waben 1 und 2 direkt nacheinander entnommen werden, da sie jeweils bereits freiliegen. Um an Wabe 3 zu gelangen, muss allerdings wieder umgestapelt werden. Für Wabe 10 kann kein Stapel gefunden werden, der die Bedingung $s^{fmin} > fpos_j$ erfüllt. Da mit $s = 2$ jedoch ein freier Stapelplatz existiert, greift hier die zweite Regel der *Max-Min*-Heuristik. Nach der Entnahme von Wabe 3 muss als nächstes Wabe 14 umgestapelt werden. In der vorliegenden Situation werden weder die Bedingungen der ersten noch die der zweiten Regel erfüllt. Somit wird Wabe 14 gemäß der dritten Regel der *Max-Min*-Heuristik auf Stapel 2 umgestapelt, da dieser den maximalen Wert für s^{fmin} annimmt.

Für die betrieblichen Zwecke der Dillinger Hüttenwerke sind die Regeln der *Max-Min*-Heuristik jedoch noch nicht ausreichend. Im Gegensatz zu den einheitlichen Abmessungen von Containern, variieren die Wabenmaße sehr stark. Dies hat vor allem zur Folge, dass beim Umstapeln auch statische Aspekte in Betracht gezogen werden müssen. Zudem ist auf die Einhaltung von Mindestabständen zwischen den Stapeln zu achten. Es werden die folgenden Umstapelbedingungen definiert:

$$\text{Bed. 1: } l_j \leq (1 + \delta^l) \cdot s^{lmin}$$

$$\text{Bed. 2: } b_j \leq (1 + \delta^b) \cdot s^{bmin}$$

$$\text{Bed. 3: } d_j + s^d \leq \delta^d$$

$$\text{Bed. 4a: } s^{x1} \geq s'^{x2} + x^{min} \vee$$

$$s^{x2} \leq s'^{x1} - x^{min} \quad \forall s' \in S \mid (s'^{y1} \geq s^{y1} \wedge s'^{y2} \leq s^{y1} + y^{min}) \vee$$

$$(s'^{y2} \leq s^{y2} \wedge s'^{y1} \geq s^{y2} - y^{min}) \vee$$

$$(s'^{y1} \leq s^{y1} \wedge s'^{y2} \geq s^{y2}) \vee$$

$$(s'^{y1} \geq s^{y1} \wedge s'^{y2} \leq s^{y2})$$

$$\text{Bed. 4b: } s^{y1} \leq s'^{y2} - y^{min} \vee$$

$$s^{y2} \geq s'^{y1} + y^{min} \quad \forall s' \in S \mid (s'^{x1} \leq s^{x1} \wedge s'^{x2} \geq s^{x1} - x^{min}) \vee$$

$$(s'^{x2} \geq s^{x2} \wedge s'^{x1} \leq s^{x2} + x^{min}) \vee$$

$$(s'^{x1} \geq s^{x1} \wedge s'^{x2} \leq s^{x2}) \vee$$

$$(s'^{x1} \leq s^{x1} \wedge s'^{x2} \geq s^{x2})$$

$$\text{Bed. 5a: } s^{x1}, s^{x2} = [lager^{x1} + x^{min}, lager^{x2} - x^{min}]$$

$$\text{Bed. 5b: } s^{y_1}, s^{y_2} = [\text{lager}^{y_1} + y^{\min}, \text{lager}^{y_2} - y^{\min}]$$

Gemäß der ersten Bedingung, darf die Länge einer umzustapelnden Wabe j nicht größer sein als $(1 + \delta^l) \cdot s^{\min}$, wobei δ^l einen festzulegenden Parameter darstellt und s^{\min} der Länge der kürzesten in Zielstapel s enthaltenen Wabe entspricht. Analog dazu darf die Breite einer umzustapelnden Wabe j nach Bedingung 2 nicht größer sein als $(1 + \delta^b) \cdot s^{b\min}$, wobei δ^b den Parameter für die Breitentoleranz darstellt und $s^{b\min}$ auf die Breite der schmalsten in Stapel s enthaltenen Wabe verweist. Bedingung 3 stellt zudem sicher, dass die aktuelle Stapelhöhe s^d zuzüglich der Dicke der umzustapelnden Wabe j nicht größer sein darf als die maximal zulässige Stapelhöhe δ^d . Die Einhaltung der Mindestabstände zwischen den Stapeln wird durch die Bedingungen 4a und 4b gewährleistet, wobei s^{x_1} , s^{y_1} , s^{x_2} und s^{y_2} jeweils Stapelkoordinaten bezeichnen und x^{\min} bzw. y^{\min} den entsprechenden längs- bzw. breitseitigen Mindestabstand repräsentiert. Schließlich sorgen die Bedingungen 5a und 5b dafür, dass kein Stapel aus dem betrachteten Lagerbereich herausragt. Die Randkoordinaten des Lagerbereichs sind mit $(\text{lager}^{x_1}, \text{lager}^{y_1})$ und $(\text{lager}^{x_2}, \text{lager}^{y_2})$ gegeben.

Die Definition bzw. Ermittlung der Stapelkoordinaten wird in Abbildung 6.10 veranschaulicht. Die hierfür erforderlichen Wabenkoordinaten werden mit $x_{1,j}$, $y_{1,j}$, $x_{2,j}$ und $y_{2,j}$ bezeichnet, wobei der Punkt $(x_{1,j}, y_{1,j})$ auf den linken, oberen Rand und der Punkt $(x_{2,j}, y_{2,j})$ auf den rechten, unteren Rand von Wabe j verweist. Die Stapelkoordinaten lassen sich schließlich aus den Wabenkoordinaten ableiten. Für den Beispielstapel aus Abbildung 6.10 ergibt sich s^{x_1} somit aus $x_{1,7}$, s^{x_2} aus $x_{2,5}$ und s^{y_1} sowie s^{y_2} jeweils aus $y_{1,8}$ und $y_{2,8}$. Die Draufsicht aus Teilabbildung 6.10(c) zeigt die Ausmaße des Stapels sowie dessen Positionierung im betrachteten Lagerbereich.

Mit den eingeführten Bedingungen lässt sich die Zulässigkeit der Stapel prüfen, die als potentielle Zielstapel für eine umzustapelnde Wabe in Frage kommen. Folglich können diese in den Regeln 1 und 3 der *Max-Min*-Heuristik integriert werden. Für Regel 2 sind jedoch weitere Anpassungen erforderlich. Das BRP geht von fest definierten Stapelplätzen aus. Sofern keine Container auf einem Stapelplatz liegen, gilt dieser folglich als frei. Aufgrund der stark variierenden Wabenmaße ist diese Betrachtungsweise nicht auf die in dieser Arbeit vorlie-

destabstände) betrachtet. Anschließend wird für diesen Punkt geprüft, ob die abzustapelnde Wabe mit ihrer linken, oberen Ecke an dieser Stelle platziert werden kann. Hierfür müssen die bereits erläuterten Umstapelbedingungen 4a und 4b sowie 5a und 5b erfüllt sein. In dem dargestellten Beispiel ist dies jedoch nicht der Fall, sodass die Vorgehensweise mit den in der entsprechenden Reihe rechts benachbarten Punkten wiederholt wird. Wenn ein Punkt erreicht ist, bei dem der rechte Rand der Wabe über den rechten Randbereich des Lagers hinausgehen würde, wird mit der darunter liegenden Zeile (wieder ganz links) fortgefahren. Die Suche ist beendet, sobald ein zulässiger Punkt für die betrachtete Wabe gefunden wurde (siehe grüner Punkt) oder sobald alle entsprechend des Rasters vorliegenden Punkte ohne Erfolg durchsucht wurden.

Mit Hilfe der erläuterten Erweiterungen lässt sich die *Max-Min*-Heuristik nun auf die betrieblichen Zwecke der Dillinger Hüttenwerke anpassen. Daraus ergibt sich der in Algorithmus 9 zusammengefasste *Umstapelalgorithmus*.

Algorithmus 9 *Umstapelalgorithmus*

Require: Wabenmenge J , l_j , b_j , d_j , $fpos_j$, $x_{1,j}$, $y_{1,j}$, $x_{2,j}$, $y_{2,j}$ mit $j \in J$,
Stapelmenge S , $folge$, x^{min} , y^{min} , δ^l , δ^{l+} , δ^b , δ^{b+} , δ^d , δ^{d+}

Ensure: s^{UZiel}

```

1: initialisiere  $s^{UZiel} = 0$ 
2: while  $s^{UZiel} = 0$  do
3:   initialisiere  $s^{fmin+} = 0$ ,  $s^{fmin-} = n$ 
4:   for  $s = 1, \dots, |S|$  do
5:     if  $s^{fmin} > fpos_{\bar{s}EZiel}$  and  $s^{fmin} < s^{fmin-}$  then
6:       prüfe Umstapelbedingungen ( $\rightarrow zul_s$ )
7:       if  $zul_s = \mathbf{true}$  then
8:          $s^{fmin-} = s^{fmin}$ 
9:          $s^{UZiel} = s$ 
10:      end if
11:    end if
12:  end for
13:  if  $s^{UZiel} = 0$  then
14:    suche nach freien Stapelplätzen für Wabe  $\bar{s}^{EZiel}$  ( $\rightarrow frei_{s'}$ )
15:    if  $frei_{s'} = \mathbf{true}$  then
16:       $S \leftarrow S \cup \{s'\}$ 
17:       $s^{UZiel} = s'$ 
18:    else
19:      for  $s = 1, \dots, |S|$  do
20:        if  $s^{fmin} < fpos_{\bar{s}EZiel}$  and  $s^{fmin} > s^{fmin+}$  then
21:          prüfe Umstapelbedingungen ( $\rightarrow zul_s$ )

```

```

22:           if  $zul_s = \mathbf{true}$  then
23:                $s^{fmin+} = s^{fmin}$ 
24:                $s^{UZiel} = s$ 
25:           end if
26:       end if
27:   end for
28: end if
29: end if
30: if  $s^{UZiel} = 0$  then
31:      $\delta^l \stackrel{+}{=} \delta^{l+}$ 
32:      $\delta^b \stackrel{+}{=} \delta^{b+}$ 
33:      $\delta^d \stackrel{+}{=} \delta^{d+}$ 
34: end if
35: end while

```

Der zunächst mit 0 initialisierte Ausgabewert s^{UZiel} verweist auf den Zielstapel für die umzustapelnde Wabe. Analog wird der Stapel, welcher die nächste zu entnehmende und an den Ofen zu transportierende Wabe enthält, mit s^{EZiel} bezeichnet. Auf dessen oberste Wabe, welche gleichzeitig der nächsten umzustapelnden Wabe entspricht, wird mit \bar{s}^{EZiel} verwiesen. Dabei sei vorausgesetzt, dass diese nicht der als nächstes an den Ofen zu transportierenden Wabe entspricht.

Mit den Zeilen 4 - 12 wird die, um die Prüfung der Umstapelbedingungen erweiterte, erste Regel der *Max-Min*-Heuristik abgedeckt, wobei s^{fmin-} als Hilfsvariable dient. Zudem wird das Ergebnis der Prüfung der Umstapelbedingungen (bzw. die Zulässigkeit des potentiellen Zielstapels) durch die boolesche Variable zul_s beschrieben. Diesbezüglich sei darauf hingewiesen, dass grundsätzlich $zul_{s^{EZiel}} = false$ gilt. Konnte kein Stapel gefunden werden, der die erste Regel erfüllt ($s^{UZiel} = 0$), fährt der Algorithmus mit der zweiten Regel, dem Suchen nach freien Stapelplätzen, fort (Zeilen 14 -18). Wird ein derartiger Platz gefunden ($frei_{s'} = true$), so muss dieser zunächst angelegt werden (Zeile 16), bevor er schließlich als Zielstapel für \bar{s}^{EZiel} gesetzt wird (Zeile 17). War jedoch auch die Suche nach freien Stapelplätzen ohne Erfolg, so tritt die dritte Regel in Kraft (Zeilen 19 - 27). Analog zur ersten Regel, bildet s^{fmin+} eine entsprechende Hilfsvariable. Zudem wird auch hier wieder die Erfüllung der Umstapelbedingungen für die in Betracht kommenden Stapel geprüft. Es sei darauf hingewiesen, dass die Einführung der Stapelbedingungen dazu führen kann, dass auch nach der dritten Regel noch immer kein gültiger Zielstapel

gefunden wird. Um dann dennoch einen Zielstapel ermitteln zu können, müssen die Umstapelbedingungen zunächst aufgeweicht werden (Zeilen 31 - 33). Hierzu werden die Toleranzparameter δ^l , δ^b und δ^d jeweils um δ^{l+} , δ^{b+} und δ^{d+} erhöht. Schließlich werden die in den Zeilen 3 - 34 beschriebenen Schritte dann solange wiederholt, bis ein zulässiger Zielstapel gefunden ist.

Mit dem beschriebenen *Umstapelalgorithmus* ist es nun möglich, geeignete Zielstapelvorschläge für umzustapelnde Waben zu generieren. Des Weiteren können die aufgrund des Umstapelns eventuell erforderlichen Wartezeiten der Waben bezüglich des Ofeneintritts nun explizit in der Bewertung der jeweiligen Reihenfolge berücksichtigt werden. Dies erfordert eine Erweiterung der Ofensimulation. Die Umsetzung wird im späteren Unterkapitel 6.3.3 im Detail präsentiert und erläutert.

6.3.2 Der Einfluss nachgelagerter Prozesse

Neben der Stapelproblematik als vorgelagertem Prozess, wird der Ofen bzw. dessen Fahrweise auch noch von verschiedenen nachgelagerten Prozessen beeinflusst. Zum einen zählt hierzu das erforderliche Abkühlen der Waben auf dem Ofenauslaufrollgang. Zum anderen können die manchen Waben nachgelagerten Brennschritte bzw. die Kapazität der hierfür benötigten Brennanlagen einen Einfluss auf den Ofendurchsatz ausüben. Im Folgenden werden beide Aspekte näher erläutert.

Das Abkühlen auf dem Ofenauslaufrollgang

Das Abkühlen der Waben auf dem Ofenauslaufrollgang ist weniger der Wärmebehandlung an sich, als vielmehr der Tatsache geschuldet, dass zu heiße Waben nicht magnetisch sind und demnach vom Kran nicht aufgenommen werden können. So kann eine Wabe den Auslaufrollgang erst verlassen, sobald ihre Kerntemperatur unter 500°C gefallen ist. Es besteht zwar auch die Möglichkeit, die Waben über eine Fähre abzuführen, jedoch steht diese häufig nicht zur Verfügung, sodass diese Option im Folgenden vernachlässigt wird.

Das zeitweise Verweilen der Waben auf dem Ofenauslaufrollgang kann in be-

stimmten Fällen zu zusätzlichen Wartezeiten vor dem Ofeneingang führen. Eine derartige Situation wird in Abbildung 6.12 veranschaulicht. Man betrachte die Waben 1 und 2. In der dargestellten Situation liegt Wabe 1 auf dem Ofenauslaufrollgang. Sie ist zum Zeitpunkt $t = 30$ in den Ofen ein- und zum Zeitpunkt $t = 315$ aus dem Ofen ausgefahren. Da sie mindestens 30 min auf dem Auslaufrollgang abkühlen muss, kann sie diesen folglich frühestens zum Zeitpunkt $t = 345$ verlassen. Ohne Berücksichtigung der Abkühldauer von Wabe 1, hätte Wabe 2 zum Zeitpunkt $t = 325$ und damit direkt 10 min nach Wabe 1 aus dem Ofen ausfahren können. Zu diesem Zeitpunkt wäre Wabe 1 jedoch noch nicht ausreichend abgekühlt. Da die Kapazität des Auslaufrollgangs nicht für beide Waben ausreicht, kann Wabe 2 folglich erst zum Zeitpunkt $t = 345$ ausfahren. Entsprechend hätte sie dann auch erst zum Zeitpunkt $t = 60$ einfahren dürfen.

Anhand des Beispiels lässt sich erkennen, dass die voraussichtliche Situation am Ofenauslaufrollgang bereits bei der Ermittlung des frühestmöglichen Einfahrzeitpunkts einer Wabe innerhalb der Simulation berücksichtigt werden muss. Aus diesem Grund wird mit $FEZO_j^{ARG}$ im Folgenden der Zeitpunkt bezeichnet, zu dem Wabe j den Auslaufrollgang frühestens verlassen kann. Er berechnet sich wie folgt:

$$FEZO_j^{ARG} = FEZO_j + akd_j^K(wbht_j^{Ziel}, 500), \forall j \in J, \quad (6.9)$$

wobei $akd_j^K(tmp^{beg}, tmp^{end})$ die Dauer bezeichnet, die der Kern von Wabe j benötigt, um von Temperatur tmp^{beg} auf Temperatur tmp^{end} abzukühlen. Analog sei $akd_j^O(tmp^{beg}, tmp^{end})$ für die Abkühlung an der Oberfläche definiert.

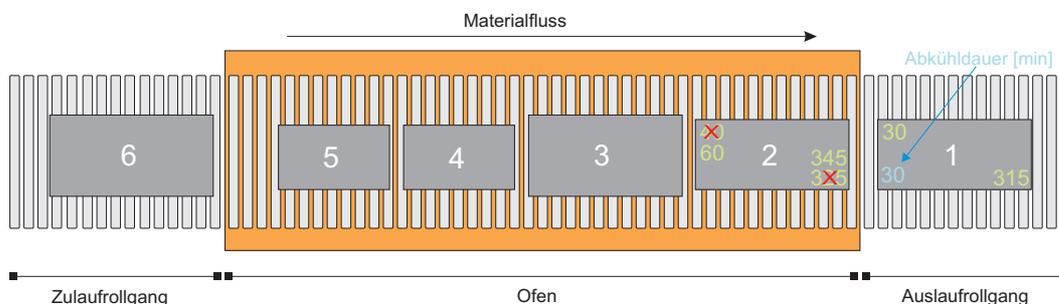


Abbildung 6.12: Beispiel für Wartezeit durch nicht ausreichend freie Kapazität auf dem Ofenauslaufrollgang

Bei den Dillinger Hüttenwerken werden verschiedene Simulationsmodelle eingesetzt, um den Temperaturverlauf sowohl innerhalb des Materials, als auch über die Zeit, abschätzen zu können. Neben materialabhängigen Parametern wie der Wärmekapazität und der Wärmeleitfähigkeit spielt hier insbesondere die Wabendicke eine entscheidende Rolle. In Abbildung 6.13 wird beispielhaft das Abkühlverhalten von drei unterschiedlich dicken Waben (50, 150 und 250 mm) abgebildet. Die Ofenentnahmetemperatur, welche i. d. R. der Halte-temperatur $wbht^{Ziel}$ entspricht, sei mit jeweils 700 °C für alle Waben gleich.

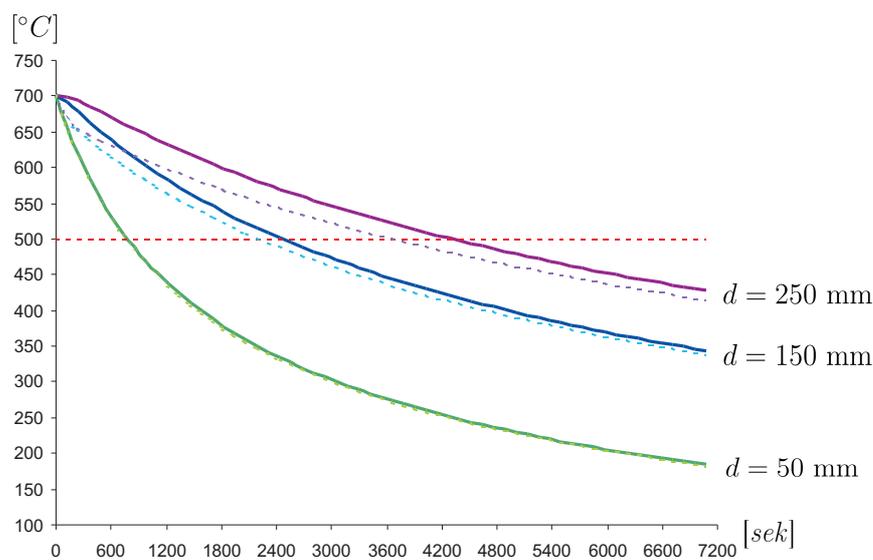


Abbildung 6.13: Das Abkühlverhalten unterschiedlicher Wabendicken

Bei dem in Abbildung 6.13 dargestellten Koordinatensystem ist auf der y-Achse die Temperatur (in $^{\circ}\text{C}$) und auf der x-Achse die Zeit (in Sekunden) abgetragen. Das Abkühlverhalten von 50, 150 und 250 mm dicken Waben wird entsprechend durch die Linien in violett, blau und grün repräsentiert, wobei die durchgezogenen Linien auf die jeweiligen Kerntemperaturen und die gleichfarbig gestrichelten Linien auf die entsprechenden Oberflächentemperaturen verweisen. Die rot gestrichelte Linie kennzeichnet zudem die Temperatur von 500 °C , die für die Ermittlung der erforderlichen Abkühlauern maßgebend ist. Insbesondere beim Vergleich der Kerntemperaturverläufe ist festzustellen, dass die dünneren im Gegensatz zu den dickeren Waben anfangs sehr schnell abkühlen. Während eine 50 mm dicke Wabe bereits nach ca. 12 min von 700 °C

auf 500 °C im Kern abgekühlt ist, benötigt eine 150 mm dicke Wabe dafür ca. 40 min und eine 250 mm Dicke Wabe sogar ca. 71 min. Die Oberflächen kühlen dagegen auch bei dickeren Waben relativ schnell ab und nähern sich erst mit fortschreitendem Abkühlverlauf den Temperaturen des Kerns an. Bei den dünneren Waben weichen Kern- und Oberflächentemperatur nur geringfügig voneinander ab.

Mit der Einführung von *FEZO^{ARG}* können nun zukünftige Kapazitätsengpässe am Ofenauslaufrollgang frühzeitig erkannt und die gegebenenfalls erforderlichen Wartezeiten entsprechend ermittelt werden. Die konkreten Schritte diesbezüglich sind wiederum in der erweiterten Materialflusssimulation des Ofens zu finden, welche in Unterkapitel 6.3.3 vorgestellt wird.

Die Brennanlagen

Wie bereits erwähnt, können auch die dem Ofen nachgelagerten Brennanlagen einen Einfluss auf die Fahrweise des Ofens ausüben. Viele der Waben, die den RHO3 durchlaufen, erfordern direkt nach der Wärmebehandlung ein Brennschneiden aus fallender Hitze. Daraus folgt, dass die jeweiligen Waben noch im warmen bzw. heißen Zustand auf einer Brennanlage bearbeitet werden müssen. Hierbei ist zu beachten, dass die Oberflächentemperatur der Wabe während des kompletten Brennschneidevorgangs nicht unter die vorgegebene Brennendtemperatur fallen darf.

Die Brennendtemperaturen einer Wabe j (nachfolgend durch $brtemp_j$ repräsentiert) liegen i. d. R. zwischen 50 und 300 °C. Ein Brennschnitt kann wenige Minuten bis hin zu über zwei Stunden andauern. Diese Brenndauer einer Wabe j sei im Folgenden mit brd_j bezeichnet. Sie hängt insbesondere von den Wabenmaßen und dem erforderlichen Brennvorgang ab. Mögliche Brennvorgänge sind z. B.:

- das Besäumen der Längskanten,
- das Schopfen der Wabenenden,
- das Längsteilen,
- das Querteilen,
- das Formteilschneiden oder

- das Brennen von Probencoupons.

Prinzipiell stehen dem Bereich, in dem sich auch der RHO3 befindet, zehn Brennportale zur Verfügung. Allerdings muss beachtet werden, dass die aus fallender Hitze zu brennenden Waben nur einen relativ kleinen Anteil an der Gesamtmenge der dort zu brennenden Bleche ausmachen. Daher werden i. d. R. nur einige wenige (ca. zwischen 1 und 3) Brennportale für das Brennen aus fallender Hitze bereitgestellt.

Bezogen auf den Ofen, kann es durch die Berücksichtigung der Brennanlagen, bzw. derer Kapazitäten, zu der folgenden, in Abbildung 6.14 dargestellten Situation kommen. Zusätzlich zu der bereits aus vorherigen Abbildungen bekannten Ofendarstellung sind hier nun auch die Brennanlagen abgebildet, die für das Brennen aus fallender Hitze vorgesehen sind. In der dargestellten Situation betrifft dies zwei Brennanlagen, wobei Wabe 1 aktuell in Brennanlage 1 bearbeitet wird. Ihr Brennvorgang hat zum Zeitpunkt $t = 300$ begonnen und wird, aufgrund ihrer Brenndauer von 100 min, zum Zeitpunkt $t = 400$ abgeschlossen sein. Wabe 2 befindet sich aktuell auf dem Auslaufrollgang, von wo sie zum Zeitpunkt $t = 325$ abgehoben werden kann. Für das Abheben, den Transport zu einer freien Brennanlage und das dortige Ablegen wird mit 10 min eine relativ großzügige Pufferzeit eingeräumt, sodass das Brennen von Wabe 2 geplant zum Zeitpunkt $t = 335$ beginnen kann. Da Brennanlage 1 zu diesem Zeitpunkt belegt ist, wird Wabe 2 auf Brennanlage 2 verplant. Für Wabe 3, die sich in der aktuellen Situation noch im Ofen befindet, diesen zum Zeit-

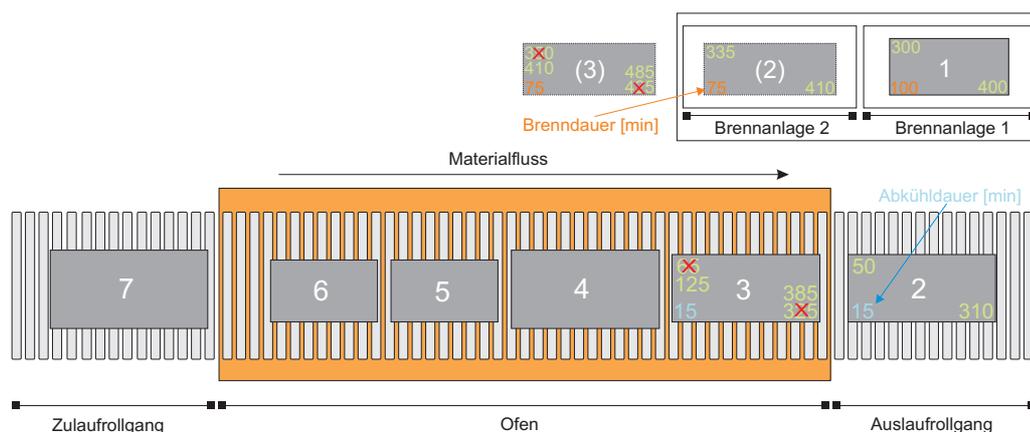


Abbildung 6.14: Beispiel für Wartezeit aufgrund eines Engpasses an den Brennanlagen

punkt $t = 325$ verlassen wird und zum Zeitpunkt $t = 350$ einer Brennanlage zugeführt werden müsste, steht dann jedoch keine freie Brennanlage mehr zur Verfügung. Ihr Brennvorgang könnte folglich frühestens zum Zeitpunkt $t = 410$ (wieder $t^{min} = 10$ min Pufferzeit angenommen) auf Brennanlage 1 eingeleitet werden. Rückgerechnet auf die Ofenzeiten von Wabe 3 folgt daraus, dass sie erst zum Zeitpunkt $t = 125$ hätte in den Ofen einfahren dürfen.

In dem erläuterten Beispiel ergibt sich für Wabe 3 aufgrund eines Engpasses an den Brennanlagen eine Wartezeit von 60 min hinsichtlich der Ofeneinfahrt. Im Prinzip kann das Warten auf eine freie Brennanlage auch nach der Ofenausfahrt erfolgen, jedoch besteht dann die Gefahr, dass die Temperatur der Wabe vor oder während des Brennvorgangs unter ihre Brennendtemperatur fällt. Aus diesem Grund wird diese Option im Folgenden nicht betrachtet.

Abbildung 6.14 zeigt, dass für die Ermittlung eventuell erforderlicher Wartezeiten aufgrund von Brennanlagenengpässen zusätzliche Planzeiten benötigt werden. Demzufolge werden $FAZB_j$ und $FEZB_j$ eingeführt, wobei $FAZB_j$ den Zeitpunkt bezeichnet, zu dem das Brennen von Wabe j frühestens beginnen kann. Dies ist prinzipiell zum Zeitpunkt $FEZO_j^{ARG}$ der Fall, wobei bereits erwähnt wurde, dass hier zusätzlich $t^{min} = 10$ min Pufferzeit für den Kran eingeräumt werden. Von größerer Bedeutung für die Ermittlung der Wartezeiten aufgrund von Brennanlagenengpässen ist mit $FEZB_j$ jedoch der frühestmögliche Endzeitpunkt des Brennvorgangs von Wabe j . Er berechnet sich wie folgt:

$$FEZB_j = FAZB_j + brd_j. \quad (6.10)$$

Neben den Waben, die im RHO3 angelassen und anschließend gebrannt werden, stehen Engpässe an den Brennanlagen insbesondere mit den Waben in Verbindung, die den RHO3 nur durchlaufen, um für das Brennen vorgewärmt zu werden. Solche, sogenannten „Vorwärmer“, müssen folglich prinzipiell nur soweit aufgeheizt werden, bis gewährleistet werden kann, dass der nachfolgende Brennvorgang noch bei Temperaturen über der Brennendtemperatur abgeschlossen sein wird. Die Zieltemperatur von Vorwärmern ist demnach individuell zu ermitteln und vom Abkühlverhalten, der Brenndauer und der Brennendtemperatur der betroffenen Wabe abhängig. Bislang trifft der Steuermann des RHO3 diese Entscheidung nach eigenem Ermessen. Die Zieltemperatur von

Vorwärmern lässt sich aber auch folgendermaßen errechnen:

$$wbht_j^{Ziel} = \operatorname{argmin}_{tmp^{beg} \in \bar{T}} \{ akd_j^O(tmp^{beg}, (1 + \varepsilon) \cdot brtemp_j) > brd_j + t^{min} \}. \quad (6.11)$$

Dabei steht tmp^{beg} für eine bestimmte, in \bar{T} enthaltene Starttemperatur, wobei $\bar{T} = \{250, 260, \dots, 540\}$ gilt. Mit $akd_j^O(tmp^{beg}, (1 + \varepsilon) \cdot brtemp_j)$ ist dann die Dauer gegeben, die Wabe j benötigt, um an ihrer Oberfläche von $tmp^{beg} \text{ }^\circ\text{C}$ auf ihre Brennendtemperatur abzukühlen. Der Parameter ε stellt in diesem Zusammenhang einen Sicherheitspuffer dar, der im Folgenden grundsätzlich auf 0,05 gesetzt wird.

Da bestimmte Gefügeumwandlungen beim Vorwärmen vermieden werden sollen, dürfen die entsprechenden Waben maximal bis $540 \text{ }^\circ\text{C}$ aufgeheizt werden. Zudem dürfen die Ofentemperaturen nicht zu hoch sein. Aktuell werden Vorwärmer, bis sie ca. $170 \text{ }^\circ\text{C}$ erreicht haben, zunächst bei einer Ofentemperatur von $630 \text{ }^\circ\text{C}$ aufgeheizt. Anschließend werden die betroffenen Temperaturzonen auf $580 \text{ }^\circ\text{C}$ heruntergeregelt. Diese Temperaturregelung erfordert eine Anpassung der Temperaturlückenberechnung für Vorwärmer. Da das Vorwärmen keine richtige Wärmebehandlung darstellt, fallen unterschiedliche Temperatureinflüsse hier nicht so stark ins Gewicht, sodass zwischen zwei Vorwärmern i. d. R. eine Lücke von $f_{templ} \cdot 10 \text{ m}$ (also zwischen 6 und 10 m) gelassen wird. Wenn allerdings eine anzulassende Wabe auf einen Vorwärmer folgt oder umgekehrt, so wird folgende Lückenberechnung vorgeschlagen:

$$templ_j = \max\{|wbht_{Ofen}^{VW1} - wbht_{j'}^{Ofen}|, |wbht_{Ofen}^{VW2} - wbht_{j'}^{Ofen}|\} \cdot f_{templ}, \quad (6.12)$$

wobei j einem Vorwärmer und j' einer anzulassenden Wabe und gleichzeitig der Vorgänger- bzw. Nachfolgerwabe von j entspricht. Zudem stellt $wbht_{Ofen}^{VW1}$ ($= 630 \text{ }^\circ\text{C}$) die Ofentemperatur dar, die für einen Vorwärmer gilt, solange dessen Temperatur weniger als $170 \text{ }^\circ\text{C}$ beträgt. Die anschließend geltende Ofentemperatur wird analog mit $wbht_{Ofen}^{VW2}$ ($= 580 \text{ }^\circ\text{C}$) bezeichnet.

Abschließend lässt sich festhalten, dass die Erweiterung des Blickwinkels auf die Brennanlagen zusätzliche Situationen liefert, in denen Waben über die Mindestwartezeit hinaus auf ihre Ofeneinfahrt warten müssen. Demnach ist auch dieser Aspekt, zusammen mit den erforderlichen Anpassungen, in der

aktuellen Materialflusssimulation des Ofens zu integrieren.

6.3.3 Anpassung der bestehenden Simulation und Analyse der Auswirkungen

Wie bereits an mehreren Stellen angedeutet, bedarf es aufgrund der Einbeziehung der erläuterten vor- und nachgelagerten Prozesse entsprechenden Anpassungen an der bisherigen Ofensimulation. *Simulation-2* wird dieser Forderung nun gerecht. Der Algorithmus stellt sich wie folgt dar:

Algorithmus 10 *Simulation-2*

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , akd_j^K , akd_j^O , brd_j mit $j \in J$,
Stapelmenge S , $folge$, f_{templ} , $templ^{min}$, t^{min} , $AnzBR$

Ensure: $FAZO_j$, $FEZO_j$, $FEZO_j^{ARG}$, $FEZB_j$ mit $j \in J$, C_{max}

```

1: for  $p = 1, \dots, n$  do
2:    $t^{trans} = t^u$ 
3:   while  $folge_p \neq \bar{s}^{EZiel}$  do
4:     call Umstapelalgorithmus
5:     aktualisiere Stapelmenge  $S$ 
6:      $t^{trans} \pm t^u$ 
7:   end while
8:   if  $p = 1$  then
9:      $templ_{folge_p} = 0$ 
10:     $FAZO_{folge_p} = t^{trans}$ 
11:  else
12:    if  $wbh_{folge_p} = VW$  and  $wbh_{folge_{p-1}} = VW$  then
13:       $tmpd = 10$ 
14:    else if  $wbh_{folge_p} = VW$  and  $wbh_{folge_{p-1}} \neq VW$  then
15:       $tmpd = \max\{|630 - wbht_{folge_{p-1}}^{Ofen}|, |580 - wbht_{folge_{p-1}}^{Ofen}|\}$ 
16:    else if  $wbh_{folge_p} \neq VW$  and  $wbh_{folge_{p-1}} = VW$  then
17:       $tmpd = \max\{|wbht_{folge_p}^{Ofen} - 630|, |wbht_{folge_p}^{Ofen} - 580|\}$ 
18:    else
19:       $tmpd = |wbht_{folge_p}^{Ofen} - wbht_{folge_{p-1}}^{Ofen}|$ 
20:    end if
21:     $templ_{folge_p} = \max\{tmpd \cdot f_{templ}, templ^{min}\}$ 
22:     $FAZO_{folge_p} = FAZO_{folge_{p-1}} + \max\{t^{min}, t^{trans}\}$ 
23:  end if
24:   $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 

```

```

25:  if  $|W^{Ofen}| > 0$  then
26:      if  $FEZO_{W_1^{Ofen}} + t^{min} \geq FEZO_{folge_p}$  then
27:           $FAZO_{folge_p} \stackrel{\pm}{=} FEZO_{W_1^{Ofen}} + t^{min} - FEZO_{folge_p}$ 
28:           $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
29:      end if
30:      while  $FEZO_{W_{|W^{Ofen}|}^{Ofen}} \leq FAZO_{folge_p}$  do
31:          if  $|W^{Ofen}| \leq 1$  then
32:               $l_{rest}^{Ofen} = l^{Ofen}$ 
33:          else
34:               $l_{rest}^{Ofen} \stackrel{\pm}{=} l_{W_{|W^{Ofen}|}^{Ofen}} + templ_{W_{|W^{Ofen}|-1}^{Ofen}}$ 
35:          end if
36:           $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}^{Ofen}\}$ 
37:      end while
38:      while  $l_{rest}^{Ofen} \leq l_{folge_p} + templ_{folge_p}$  do
39:          if  $|W^{Ofen}| = 0$  then
40:               $l_{rest}^{Ofen} = l^{Ofen}$ 
41:          end if
42:          if  $FEZO_{W_{|W^{Ofen}|}^{Ofen}} > FAZO_{folge_p}$  then
43:               $wl_{folge_p} = FEZO_{W_{|W^{Ofen}|}^{Ofen}} - FAZO_{folge_p}$ 
44:          end if
45:          if  $|W^{Ofen}| = 1$  then
46:               $l_{rest}^{Ofen} = l^{Ofen}$ 
47:          else
48:               $l_{rest}^{Ofen} \stackrel{\pm}{=} l_{W_{|W^{Ofen}|}^{Ofen}} + templ_{W_{|W^{Ofen}|-1}^{Ofen}}$ 
49:          end if
50:           $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}^{Ofen}\}$ 
51:      end while
52:       $FAZO_{folge_p} \stackrel{\pm}{=} wl_{folge_p}$ 
53:       $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
54:  end if
55:   $l_{rest}^{ARG} = l^{ARG}$ 
56:   $W^{ARG} = ()$ 
57:  for  $q = 1, \dots, p - 1$  do
58:      if  $FEZO_{folge_q}^{ARG} > FEZO_{folge_p} + t^{min}$  then
59:           $W^{ARG} \leftarrow W^{ARG} \cup \{folge_q\}$ 
60:           $l_{rest}^{ARG} = l_{folge_q} + templ^{min}$ 
61:      end if
62:  end for
63:  while  $l_{rest}^{ARG} < l_{folge_p}$  do
64:       $FEZO_{folge_p} = FEZO_{W_{|W^{ARG}|}^{ARG}}$ 

```

```

65:     FAZOfolgep = FEZOfolgep - tmin - ovzfolgep
66:     lrestARG ≐ l|WARG| + templmin
67:     WARG ← WARG \ {W|WARG|ARG}
68: end while
69:     FEZOfolgepARG = FEZOfolgep + tmin + akdfolgepK(wbhtfolgepZiel, 500)
70: if brdfolgep > 0 then
71:     WBR = ()
72:     for q = 1, ..., p - 1 do
73:         if FEZBfolgeq > FEZOfolgeqARG + tmin then
74:             WBR ← WBR ∪ {folgeq}
75:         end if
76:     end for
77:     if |WBR| = AnzBR then
78:         j* = argminj ∈ WBR{FEZBj}
79:         FEZOfolgepARG = FEZBj*
80:         FEZOfolgep = FEZOfolgepARG - tmin - akdfolgepK(wbhtfolgepZiel, 500)
81:         FAZOfolgep = FEZOfolgep - tmin - ovzfolgep
82:     end if
83:     FEZBfolgep = FEZOfolgepARG + tmin + brdfolgep
84: end if
85: if |WOfen| = 0 then
86:     lrestOfen ≐ lfolgep - templfolgep
87: else
88:     lrestOfen ≐ lfolgep
89: end if
90:     WOfen ← WOfen ∪ {folgep}
91: end for
92: Cmax = FEZOfolgen

```

Die der Stapel- bzw. Umstapelproblematik geschuldeten Erweiterungen sind in den Zeilen 2 - 10, sowie in Zeile 22 zu finden. Mit t^{trans} wird die Zeitdauer bezeichnet, die der Kran voraussichtlich benötigt, um die betrachtete Wabe aus dem Lagerbereich zu entnehmen und schließlich an den Ofen zu transportieren. Da diese Bereitstellungszeit mindestens t^u min erfordert (für den Fall, dass die betrachtete Wabe bereits freiliegt), wird t^{trans} zunächst grundsätzlich mit t^u initialisiert (Zeile 2). Falls die betrachtete Wabe jedoch zunächst freigelegt werden muss, wird der in Algorithmus 9 beschriebene *Umstapelalgorithmus* aufgerufen, die entsprechende Umstapelaktion durchgeführt und t^{trans} um t^u erhöht. Diese Schritte werden solange wiederholt, bis keine Wabe mehr über

der betrachteten Wabe liegt und diese folglich entnommen werden kann. Da aus den erforderlichen Bereitstellungszeiten zusätzliche Wartezeiten für den Ofen resultieren können, gilt es schließlich noch die Initialisierung der geplanten Ofeneinfahrzeitpunkte anzupassen (Zeilen 10 und 22).

An dieser Stelle sei nochmals explizit auf einige Annahmen hingewiesen, die der geschilderten Vorgehensweise bezüglich des Umgangs mit der Stapel- bzw. Umstapelproblematik zugrunde liegen. Zum einen handelt es sich hierbei, wie auch bei dem BRP und dem SSSP, um eine statische Sichtweise, d.h. über die Zeit im Lager neu ankommende Waben werden nicht in Betracht gezogen. Dies ist relativ unrealistisch, jedoch kaum zu ändern, da die Ankunft zukünftiger Waben nicht genau genug vorherzusagen ist. Des Weiteren wird bezüglich der geplanten Umstapelaktionen davon ausgegangen, dass das Abstapeln der Waben immer mittig auf dem betreffenden Stapel erfolgt. Dies ist in der Praxis für den Kranfahrer zwar nur schwer realisierbar, aber die einzig sinnvolle Annahme. Zudem liegt *Simulation-2* die Annahme zugrunde, dass der Kranfahrer sich erst der Entnahme der nächsten Wabe widmet, wenn die entsprechende Vorgängerwabe in den Ofen einfährt. Somit wird vernachlässigt, dass der Kranfahrer prinzipiell vorarbeiten bzw. vorsortieren könnte. Da der Kranfahrer neben der Bereitstellung der Waben noch viele andere Aufgaben zu bewerkstelligen hat und zudem nur über eingeschränkte Informationen bezüglich der geplanten Ofenreihenfolge verfügt, entspricht diese Annahme jedoch weitestgehend der Realität.

Neben der Umstapelproblematik werden in *Simulation-2* nun auch Vorwärmer berücksichtigt. Wie bereits erläutert, ist diesbezüglich die Temperaturlückenberechnung anzupassen. Die entsprechenden Änderungen sind dem Zeilenbereich 12 - 21 zu entnehmen, wobei *tmpd* als Hilfsvariable dient. Zudem wird der Parameter *wbh_j* eingeführt, um auf das Wärmebehandlungsverfahren von Wabe *j* zu verweisen (VW = Vorwärmen, A = Anlassen). Das Kantenentspannen und das Effusionsglühen seien an dieser Stelle vernachlässigt.

Die in den Zeilen 24 - 54 durchgeführten Schritte sind analog in *Simulation-1* vorzufinden. Der Zeilenbereich 55 - 69 stellt dagegen wiederum eine Erweiterung dar. Hier werden die Wartezeiten ermittelt, die gegebenenfalls eingehalten werden müssen, um zukünftige Kapazitätsengpässe am Ofenauslaufrollgang zu

verhindern. Zunächst wird die Kapazität bzw. Restkapazität des Auslaufrollgangs l_{rest}^{ARG} mit dessen Gesamtlänge ($l^{ARG} = 30000$) gleichgesetzt (Zeile 55). Zudem wird die geordnete Menge W^{ARG} als leere Menge initialisiert (Zeile 56). Dieser wird anschließend jede Wabe zugeordnet, die sich zum Zeitpunkt der bisher geplanten Ofenausfahrt der betrachteten Wabe noch auf dem Auslaufrollgang befindet (Zeilen 57 - 62). Dabei wird gleichzeitig die zu diesem Zeitpunkt zur Verfügung stehende Restkapazität des Auslaufrollgangs ermittelt. Falls diese nicht ausreicht, um die betrachtete Wabe aufzunehmen, muss gewartet werden, bis die nächste Wabe den Auslaufrollgang verlässt. Die Ofenfahrzeiten der betrachteten Wabe sowie l_{rest}^{ARG} und W^{ARG} sind dementsprechend anzupassen. Dieses Vorgehen wird solange wiederholt, bis der Auslaufrollgang über ausreichend Kapazität verfügt (Zeilen 63 - 68). Aus dem ermittelten Ofenausfahrzeitpunkt ergibt sich schließlich der frühestmögliche Zeitpunkt, zu dem die betrachtete Wabe den Ofenauslaufrollgang verlassen wird (Zeile 69).

Die zusätzliche Berücksichtigung der Brennanlagenkapazitäten (Zeilen 70 - 84) verläuft auf ähnliche Weise, wobei diese Schritte nur erforderlich sind, wenn die betrachtete Wabe nach ihrer Wärmebehandlung aus fallender Hitze gebrannt werden soll. Zunächst wird analog zu W^{ARG} eine geordnete Menge W^{BR} eingeführt und als leere Menge initialisiert (Zeile 71). Im Anschluss wird dieser Menge wiederum jede Wabe zugeordnet, die sich zum Zeitpunkt des bisher geplanten Verlassens des Auslaufrollgangs der betrachteten Wabe noch auf einer Brennanlage befindet (Zeilen 72 - 76). Im folgenden Schritt wird geprüft, ob zu diesem Zeitpunkt alle Brennanlagen belegt sind, wobei $AnzBR$ die vorgegebene Anzahl prinzipiell zur Verfügung stehender Brennanlagen bezeichnet (Zeile 77). Sollte dies der Fall sein, so wird zunächst die Wabe j^* ermittelt, deren Brennvorgang als nächstes beendet sein wird (Zeile 78). Diesem Zeitpunkt entsprechend, müssen dann die Planzeiten $FEZO^{ARG}$, $FEZO$ und $FAZO$ für die betrachtete Wabe aktualisiert werden (Zeilen 79 - 81). In Zeile 83 wird dann schließlich der Zeitpunkt ermittelt, zu dem der Brennvorgang der betrachteten Wabe frühestens abgeschlossen sein wird. Der restliche Zeilenbereich 85 - 92 entspricht gleichermaßen den bereits aus *Simulation-1* bekannten, abschließenden Berechnungen.

Wie bereits bei der Einführung der Verband- bzw. der *Hollomon*-Fahrweise sollen auch die isolierten Auswirkungen der neuen Materialflusssimulation des

Ofens anhand einiger Rechentests demonstriert werden. Hierzu werden zum einen wiederum die Datensätze *dat3* - *dat30* herangezogen. Da diese jedoch nur anzulassende Waben beinhalten und eine Erweiterung aus *Simulation-2* darin besteht, Vorwärmer zu berücksichtigen, werden mit den Datensätzen *dat10_{vw}*, *dat16_{vw}* und *dat25_{vw}* zum anderen drei weitere Datensätze eingeführt, welche ausschließlich Vorwärmer beinhalten. Zudem sei darauf hingewiesen, dass allen Testrechnungen hinsichtlich der Stapelbedingungen folgende Parameterwerte zugrunde liegen:

$$\begin{aligned} \delta^l &= 0,6 & \delta^{l+} &= 0,05 \\ \delta^b &= 0,9 & \delta^{b+} &= 0,075 \\ \delta^d &= 5000 & \delta^{d+} &= 50 \\ x^{\min} &= 500 & y^{\min} &= 1000 \end{aligned}$$

Die Ergebnisse der Testrechnungen für die Datensätze *dat3* - *dat30* unter Annahme von $f_{templ} = 1000$ und $AnzBR = 2$ sind in Tabelle 6.8 zusammengefasst. Dabei sind sowohl die Ergebnisse der *Rule-based Sorting*-Heuristik als auch die des *HTFSPD-Branch&Bound*-Verfahrens jeweils für beide Bewertungsmethoden einander gegenübergestellt.

Der Vergleich zeigt, dass sich die für den Ofen relevanten vor- und nachgelagerten Prozesse, insbesondere bei den etwas größeren Datensätzen, erheblich

Tabelle 6.8: Vergleich der Rechenergebnisse für *Simulation-2*
(mit $f_{templ} = 1000$ und $AnzBR = 2$)

| Daten- satz | <i>Simulation-1</i> | | | | <i>Simulation-2</i> | | | |
|----------------|---------------------|-----------|-------|----------|---------------------|-----------|--------|----------|
| | RBS C_{max} | BB-PFDN | | Zeit | RBS C_{max} | BB-PFDN | | Zeit |
| | C_{max} | C_{max} | Gap | Zeit | C_{max} | C_{max} | Gap | Zeit |
| <i>dat3</i> | 548 | 548 | 0.00 | < 1.00 | 554 | 551 | 0.00 | < 1.00 |
| <i>dat5</i> | 846 | 846 | 0.00 | < 1.00 | 980 | 849 | 0.00 | < 1.00 |
| <i>dat8</i> | 704 | 614 | 0.00 | < 1.00 | 710 | 620 | 0.00 | < 1.00 |
| <i>dat10</i> | 713 | 656 | 0.00 | < 1.00 | 744 | 721 | 0.00 | 20.56 |
| <i>dat12</i> | 931 | 885 | 0.00 | 135.66 | 1079 | 929 | 0.00 | 5742.91 |
| <i>dat15</i> | 871 | 843 | 0.00 | 267.43 | 1171 | 919 | 31.77 | 18000.00 |
| <i>dat20</i> | 920 | 920 | 46.74 | 451.71* | 1721 | 1176 | 100.00 | 18000.00 |
| <i>dat30</i> | 1311 | 1306 | 54.82 | 18000.00 | 2793 | 2279 | 100.00 | 18000.00 |

*Abbruch aufgrund des Speicherplatzbedarfs

auf den Ofendurchsatz auswirken. So ergibt sich beispielsweise für Datensatz *dat30* bei Anwendung der *Rule-based Sorting*-Heuristik mit *Simulation-2* eine Gesamtfertigstellungszeit, die mehr als doppelt so hoch ist, wie die entsprechend mit *Simulation-1* bewertete Lösung. Es sei darauf hingewiesen, dass in diesem Fall sogar identische Reihenfolgen zugrunde liegen. Auch die Ergebnisse des *HTFSPD-Branch&Bound*-Verfahrens weichen bezogen auf die verwendete Bewertungsmethode teilweise stark voneinander ab. Ausschlaggebend für die reine Reihenfolgeoptimierung ist nun jedoch, ob auch die ermittelten Reihenfolgen voneinander abweichen. Diesbezüglich sei darauf hingewiesen, dass dies, außer bei den kleineren Datensätzen (*dat3* - *dat8*), immer der Fall war. Daher ist *Simulation-2* als Bewertungsgrundlage *Simulation-1* vorzuziehen.

Eine weitere Beobachtung, die aus den Ergebnissen zu entnehmen ist, besteht darin, dass *Simulation-2* jetzt auch ein höheres Optimierungspotential liefert. So liegt die Lösung der *Rule-based Sorting*-Heuristik z. B. bei Datensatz *dat12* nun um ca. 16 % von der optimalen Lösung entfernt. Bei *Simulation-1* waren es dagegen lediglich ca. 5 %. Mit der Wahl von *Simulation-2* scheint somit auch die Komplexität der Reihenfolgebildung zuzunehmen. Es gilt jedoch auch zu beachten, dass *Simulation-2* längere Rechenzeiten als *Simulation-1* erfordert.

Für $f_{templ} = 600$ ergeben sich die in Tabelle 6.9 abgebildeten Ergebnisse. Auch hier lassen sich die gleichen, bereits erläuterten Beobachtungen ableiten. Das

Tabelle 6.9: Vergleich der Rechenergebnisse für *Simulation-2*
(mit $f_{templ} = 600$ und $AnzBR = 2$)

| Daten- satz | <i>Simulation-1</i> | | | | <i>Simulation-2</i> | | | |
|----------------|---------------------|-----------|-------|----------|---------------------|-----------|--------|----------|
| | RBS C_{max} | BB-PFDN | | Zeit | RBS C_{max} | BB-PFDN | | Zeit |
| | C_{max} | C_{max} | Gap | Zeit | C_{max} | C_{max} | Gap | Zeit |
| <i>dat3</i> | 417 | 417 | 0.00 | < 1.00 | 458 | 420 | 0.00 | < 1.00 |
| <i>dat5</i> | 846 | 846 | 0.00 | < 1.00 | 980 | 849 | 0.00 | < 1.00 |
| <i>dat8</i> | 612 | 543 | 0.00 | < 1.00 | 736 | 572 | 0.00 | 2.04 |
| <i>dat10</i> | 703 | 605 | 0.00 | < 1.00 | 852 | 659 | 0.00 | 41.88 |
| <i>dat12</i> | 768 | 723 | 0.00 | 21.70 | 930 | 787 | 0.00 | 1272.34 |
| <i>dat15</i> | 798 | 778 | 0.00 | 688.89 | 1136 | 878 | 33.03 | 18000.00 |
| <i>dat20</i> | 900 | 759 | 37.55 | 514.07* | 1721 | 1135 | 100.00 | 18000.00 |
| <i>dat30</i> | 1303 | 1197 | 44.44 | 18000.00 | 2793 | 2279 | 100.00 | 18000.00 |

*Abbruch aufgrund des Speicherplatzbedarfs

Optimierungspotential gegenüber der *Rule-based Sorting*-Heuristik ist sogar noch ein wenig höher als bei $f_{templ} = 1000$.

Da Vorwärmer mit *Simulation-1* nicht berücksichtigt werden können, ist diesbezüglich kein direkter Vergleich zwischen den beiden Bewertungsmethoden möglich. Um dennoch den Einfluss von Vorwärmern zu demonstrieren, werden für die Datensätze $dat10_{vw}$, $dat16_{vw}$ und $dat25_{vw}$ jeweils drei Testrechnungen mit unterschiedlichen Vorgaben für $AnzBR$ (2, 4 und 6) durchgeführt. Die mit $f_{templ} = 1000$ erzielten Ergebnisse sind in Tabelle 6.10 dargestellt.

Tabelle 6.10: Vergleich der Rechenergebnisse für Vorwärmer (mit *Simulation-2* und $f_{templ} = 1000$)

| Daten- satz | $AnzBR$ | RBS | BB-PFDN | | |
|----------------|---------|-----------|-----------|--------|----------|
| | | C_{max} | C_{max} | Gap | Zeit |
| $dat10_{vw}$ | 2 | 499 | 386 | 0.00 | 845.84 |
| | 4 | 343 | 274 | 0.00 | 322.66 |
| | 6 | 320 | 274 | 0.00 | 317.37 |
| $dat16_{vw}$ | 2 | 872 | 792 | 100.00 | 18000.00 |
| | 4 | 602 | 419 | 100.00 | 18000.00 |
| | 6 | 602 | 329 | 100.00 | 18000.00 |
| $dat25_{vw}$ | 2 | 1504 | 1292 | 100.00 | 18000.00 |
| | 4 | 1354 | 906 | 100.00 | 18000.00 |
| | 6 | 1354 | 870 | 100.00 | 18000.00 |

Man stellt fest, dass nur Datensatz $dat10_{vw}$ innerhalb des vorgegebenen Zeitrahmens optimal gelöst werden konnte. Trotzdem zeigt sich auch hier wiederum ein großes Optimierungspotential gegenüber der *Rule-based Sorting*-Heuristik. Aus der Tatsache, dass die optimalen Gesamtfertigstellungszeiten mit steigendem $AnzBR$ weiter abnehmen, lässt sich weiterhin folgern, dass die Kapazität der Brennanlagen häufig eher einen Engpass darstellt als die des Ofens. Dieser Effekt ist bei Vorwärmern stärker ausgeprägt als bei anzulassenden Waben, da die Ofenverweilzeiten beim Vorwärmen i. d. R. wesentlich kürzer sind als beim Anlassen und teilweise sogar unter den jeweils erforderlichen Brennzeiten liegen.

Auch für die Vorwärmer werden die Testrechnungen wieder mit der Vorgabe $f_{templ} = 600$ wiederholt. Tabelle 6.11 zeigt die erzielten Ergebnisse. Sie stützen

Tabelle 6.11: Vergleich der Rechenergebnisse für Vorwärmer
(mit *Simulation-2* und $f_{templ} = 600$)

| Daten- satz | AnzBR | RBS | | BB-PFDN | |
|---------------------------|-------|-----------|-----------|---------|-----------|
| | | C_{max} | C_{max} | Gap | Zeit |
| <i>dat10_{vw}</i> | 2 | 499 | 386 | 0.00 | 852.59 |
| | 4 | 343 | 252 | 0.00 | 133.57 |
| | 6 | 297 | 252 | 0.00 | 87.14 |
| <i>dat16_{vw}</i> | 2 | 872 | 792 | 100.00 | 18000.00 |
| | 4 | 602 | 419 | 100.00 | 18000.00 |
| | 6 | 602 | 306 | 100.00 | 18000.00 |
| <i>dat25_{vw}</i> | 2 | 1504 | 1292 | 100.00 | 18000.00 |
| | 4 | 1354 | 872 | 100.00 | 18000.00 |
| | 6 | 1354 | 809 | 100.00 | 14201.20* |

*Abbruch aufgrund des Speicherplatzbedarfs

die Aussage, dass die Brennanlagen häufig den Engpass darstellen. So ergeben sich für $AnzBR = 2$ die identischen Ergebnisse wie zuvor in Tabelle 6.10 mit $f_{templ} = 1000$. Erst wenn mehr Brennanlagen zur Verfügung stehen, beginnen die Ergebnisse zwischen beiden Tabellen abzuweichen. Aufgrund der kleineren Temperaturlücken, liegen die Gesamtfertigstellungszeiten bei $f_{templ} = 600$ dann leicht unter den entsprechend mit $f_{templ} = 1000$ resultierenden Zeiten.

Es ist festzuhalten, dass mit *Simulation-2* nun eine Bewertungsmethode vorliegt, die im Vergleich zu *Simulation-1* eine zweckgerichtetere Reihenfolgeoptimierung des Ofens gewährleistet. Neben der reinen Optimierung bringt diese Bewertungsmethode jedoch noch weitere Verbesserungen mit sich. Mit dem *Umstapelalgorithmus* als Nebenprodukt ist beispielsweise ein Verfahren integriert worden, welches gezielt Zielvorschläge für umzustapelnde Waben generiert und somit den Kran um einige Umstapelaktionen entlastet. Des Weiteren können für Vorwärmer nun auch Zieltemperaturen rückgerechnet werden, die dem Ofensteuerer als Richtwert dienen, um die entsprechenden Waben ausreichend aufzuheizen.

Es hat sich herausgestellt, dass die Verwendung von *Simulation-2*, ebenso wie die Einbindung der Verband- und der *Hollomon*-Fahrweise, eine Erweiterung gegenüber der in Kapitel 5 vorgestellten Problematik darstellt. Während die Parallel-Fahrweise durch dessen Entkopplung jedoch eher in einer Problemre-

duzierung resultiert, erhöht sich die Komplexität der Reihenfolgebildung dagegen sowohl bei Berücksichtigung der *Hollomon*-Fahrweise als auch bei Verwendung von *Simulation-2* als Bewertungskriterium.

Kapitel 7

Konzeption eines gesamtheitlichen Lösungsverfahrens

Das folgende Kapitel setzt die einzelnen, in den vorangegangenen Kapiteln erläuterten Teilprobleme zusammen und befasst sich mit der Konzeption eines gesamtheitlichen Lösungsverfahrens. Unterkapitel 7.1 schildert Idee, Aufbau und Struktur des vorgeschlagenen Verfahrens, ehe dessen einzelne Bausteine in den darauffolgenden Unterkapiteln im Detail erläutert werden. So widmet sich Unterkapitel 7.2 zunächst dem eigentlichen Kern des gesamtheitlichen Lösungskonzepts, den eine hybride Metaheuristik bildet. Daraufhin befasst sich Unterkapitel 7.3 mit der Problematik, welche Anpassungen erforderlich sind, um den sogenannten *Online*-Aspekt zu berücksichtigen. Anschließend wird in Unterkapitel 7.4 eine Optimierung der Temperaturlückenvorgaben vorgestellt. Da beide Aspekte aus den Unterkapiteln 7.3 und 7.4 die Bewertungsmethode betreffen bzw. darin zu integrieren sind, widmet sich Unterkapitel 7.5 zum Abschluss der Beschreibung einer wiederum erweiterten Materialflusssimulation des Ofens.

7.1 Idee, Aufbau und Struktur

Ein effizientes Lösungskonzept für die vorliegende Problemstellung erfordert die Zusammenführung der bislang meist separat betrachteten Teilprobleme. Hierbei gilt es, sowohl voneinander unabhängige Teilprobleme ausfindig zu machen, als auch bestehende Interdependenzen zu ermitteln. So wurde bereits in Kapitel 6 festgestellt, dass die Verband-Fahrweise nur indirekt bzw. einseitig mit der reinen Reihenfolgeoptimierung verbunden ist, während zwischen der *Hollomon*-Fahrweise und der reinen Reihenfolgeoptimierung eine direkte bzw. wechselseitige Beziehung besteht. Diese Beziehungen sind für das gesamtheitliche Lösungskonzept dahingehend entscheidend, dass sie Aufschluss darüber geben, welche Teilprobleme sequentiell und welche simultan zu lösen sind.

Wie bereits das in Kapitel 5.4.2 vorgestellte *Branch&Bound*-Verfahren (vgl. Algorithmus 5) basiert auch das im Rahmen der vorliegenden Arbeit erarbeitete, gesamtheitliche Lösungsverfahren auf der Kopplung von Simulation und Optimierung. Das Gesamtkonzept lässt sich anhand von Abbildung 7.1 erläutern. In einem ersten Schritt wird mit der WBH-Blockbildung eine Problemzerlegung vorgenommen, die darin besteht, die verschiedenen Waben zunächst hinsichtlich ihres Wärmebehandlungsverfahrens in Gruppen (WBH-Blöcke) einzuteilen. Die anschließende Reihenfolgeoptimierung wird für alle Blöcke separat durchgeführt. Der Hintergedanke dieser Vorgehensweise ist, dadurch die Komplexität der Gesamtproblematik zu reduzieren. Da die Temperaturen und Ofenverweilzeiten zwischen den einzelnen Wärmebehandlungsverfahren z. T. stark voneinander abweichen, ist es sehr unwahrscheinlich, dass häufige Verfahrenswechsel zu einem hohen Gesamtdurchsatz führen. Aus diesem Grund erscheint die WBH-Blockbildung als gerechtfertigt. Für die Reihenfolge, in der die einzelnen WBH-Blöcke abgearbeitet werden, sind die Sollproduktionswochen der Waben entscheidend. Es wird folgende Sortierregel vorgeschlagen:

- Ermittle für jeden WBH-Block die Wabe mit der kleinsten Sollproduktionswoche und sortiere die Blöcke anschließend aufsteigend nach dieser kleinsten Sollproduktionswoche.
- Sortiere alle WBH-Blöcke, deren kleinste Sollproduktionswoche gleich ist, aufsteigend nach dem Durchschnitt aus allen im jeweiligen WBH-Block enthaltenen Sollproduktionswochen.

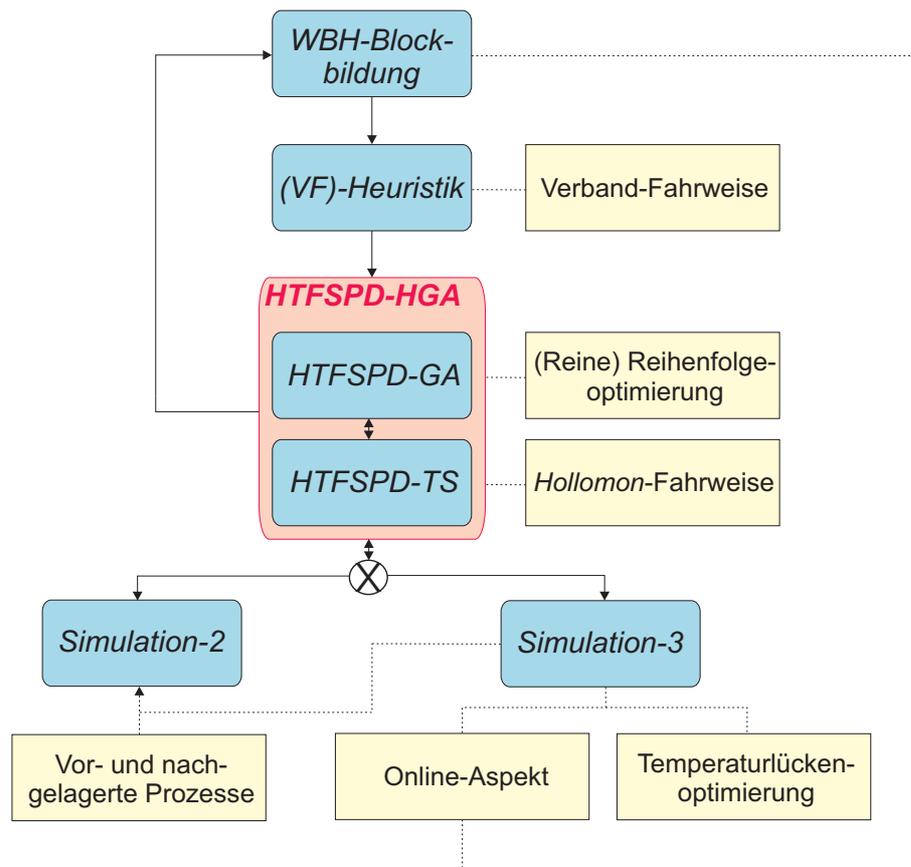


Abbildung 7.1: Gesamtheitliches Lösungskonzept

Der nächste Schritt besteht darin, die (VF)-Heuristik für den aktuell betrachteten WBH-Block auszuführen. Es sei darauf hingewiesen, dass Vorwärmer hier jedoch ausgeschlossen werden müssen, da nach dem Ofenaustritt eines Vorwärmers ein möglichst schneller Transport auf eine Brennanlage sicherzustellen ist. Im Falle mehrerer nebeneinander liegender Vorwärmer kann dies jedoch nicht gewährleistet werden.

Im darauffolgenden Schritt beginnt die tatsächliche Reihenfolgeoptimierung. Wie die bisher durchgeführten Auswertungen (vgl. Kapitel 5 und 6) zeigen, kann das *HTFSPD-Branch&Bound*-Verfahren im Realbetrieb der Dillinger Hüttenwerke aufgrund zu langer Rechenzeiten nicht eingesetzt werden. In Kapitel 4.3.2 wurden bereits verschiedene Metaheuristiken vorgestellt. Dabei wurde zwischen einzellösungs- und populationsbasierten Metaheuristiken unterschieden, wobei i. d. R. einzellösungsbasierte Heuristiken eher ein intensives und populationsbasierte Heuristiken eher ein diversifizierendes Suchverhalten

aufweisen. Mit hybriden Metaheuristiken wird versucht die Vorteile beider Verfahrensgruppen auszuschöpfen, indem man sie auf bestimmte Weise kombiniert. In Talbi [83] werden diesbezüglich einige grundsätzliche Konstellationen vorgeschlagen.

Auch wenn Metaheuristiken prinzipiell als universell einsetzbar gelten, sollten die Eigenschaften der vorliegenden Problemstellung bei der Wahl des Verfahrens immer in Betracht gezogen werden. Für Reihenfolgeprobleme empfehlen Wang und Wu [96] beispielsweise die Verwendung eines genetischen Algorithmus in Verbindung mit einer iterativen Nachbarschaftssuche, welche als eine Art Lernprozess jeweils der Mutation nachfolgend durchgeführt wird. Für die dieser Arbeit zugrundeliegende Problemstellung wird ein ähnlicher Ansatz gewählt. Anstatt der iterativen Nachbarschaftssuche, wird dagegen ein *Tabu Search* (HTFSPD-TS) gewählt und in einen genetischen Algorithmus (HTFSPD-GA) integriert. Das kombinierte Verfahren wird als *HTFSPD Hybrid Genetic Algorithm* (HTFSPD-HGA) bezeichnet und in Unterkapitel 7.2 näher erläutert. An dieser Stelle sei jedoch bereits darauf hingewiesen, dass beim HTFSPD-HGA die Optimierung der reinen Reihenfolge und der *Holomon*-Fahrweise simultan erfolgt.

In Kapitel 6.3.3 wurde mit *Simulation-2* durch die Einbindung für den Ofen relevanter, vor- und nachgelagerter Prozesse bereits eine Erweiterung der bis dahin geltenden Materialflusssimulation des Ofens (*Simulation-1*) vorgestellt. Während der Implementierungsphase im Realbetrieb der Dillinger Hüttenwerke konnte jedoch festgestellt werden, dass weitere Anpassungen und Verbesserungen an der Simulation bzw. Bewertungsmethode erforderlich sind, die insbesondere den sogenannten *Online*-Aspekt und die Temperaturlückenoptimierung betreffen. Beide Aspekte werden in den Unterkapiteln 7.3 und 7.4 näher erläutert. Die durch deren Einbindung neu entstandene Bewertungsmethode (*Simulation-3*) wird schließlich in Unterkapitel 7.5 vorgestellt. Es sei jedoch bereits erwähnt, dass sich die Betrachtung des *Online*-Aspekts neben der Reihenfolgebewertung auch direkt auf die WBH-Blockbildung auswirkt.

7.2 Der hybride genetische Algorithmus für das HTFSPD (HTFSPD-HGA)

In Kapitel 4.3.2 wurde der grundlegende Ablauf von genetischen Algorithmen und *Tabu Search* bereits vorgestellt. Beide Verfahren besitzen ihre spezifischen Stärken und Schwächen. Genetische Algorithmen durchsuchen den Lösungsraum i. d. R. sehr weiträumig. Sie versuchen dadurch dem Verharren in einem lokalen Optimum entgegenzuwirken und so bestenfalls das globale Optimum zu finden. Der Nachteil dieser Strategie ist häufig, dass bestimmte Bereiche des Lösungsraums nicht ausführlich genug durchsucht werden, sodass sich die einzelnen lokalen Optima gar nicht erst erschließen. *Tabu Search* sucht dagegen i. d. R. sehr intensiv. So können lokale Optima zwar eher aufgespürt werden als mit einem genetischen Algorithmus, jedoch ist der Suchraum dann meist auf einen relativ kleinen Bereich des Lösungsraums beschränkt. Aus diesem Grund ist der erfolgreiche Einsatz von *Tabu Search* häufig stark von der gewählten Startlösung abhängig.

Mit einem hybriden Ansatz, wie dem HTFSPD-HGA, wird versucht, sich die jeweiligen Stärken von genetischen Algorithmen und *Tabu Search* zu Nutze zu machen und dabei gleichzeitig die bestehenden Schwächen zu relativieren. So soll der genetische Algorithmus (HTFSPD-GA) für eine weiträumige Durchsuchung des Lösungsraums sorgen und der *Tabu Search* (HTFSPD-TS) hin und wieder die Suche in bestimmten Bereichen intensivieren.

Der Ablauf des HTFSPD-HGA ist in Abbildung 7.2 veranschaulicht. Es handelt sich, wie bereits erwähnt, um ein Zusammenspiel zwischen dem genetischen Algorithmus (HTFSPD-GA) und dem *Tabu Search* (HTFSPD-TS), wobei der *Tabu Search*, im Sinne eines Lernprozesses, der Mutation des genetischen Algorithmus nachgeschaltet ist.

Wenn der HTFSPD-TS am Ende jeder Iteration des HTFSPD-GA aufgerufen wird, dürfen die Rechenzeiten des HTFSPD-TS nicht zu hoch sein. Ansonsten würde der genetische Algorithmus nur sehr langsam von Iteration zu Iteration fortschreiten und gegebenenfalls mit relativ kleiner Iterationsanzahl abbrechen. In einem solchen Fall wären Sinn und Zweck des Verfahrens verfehlt.

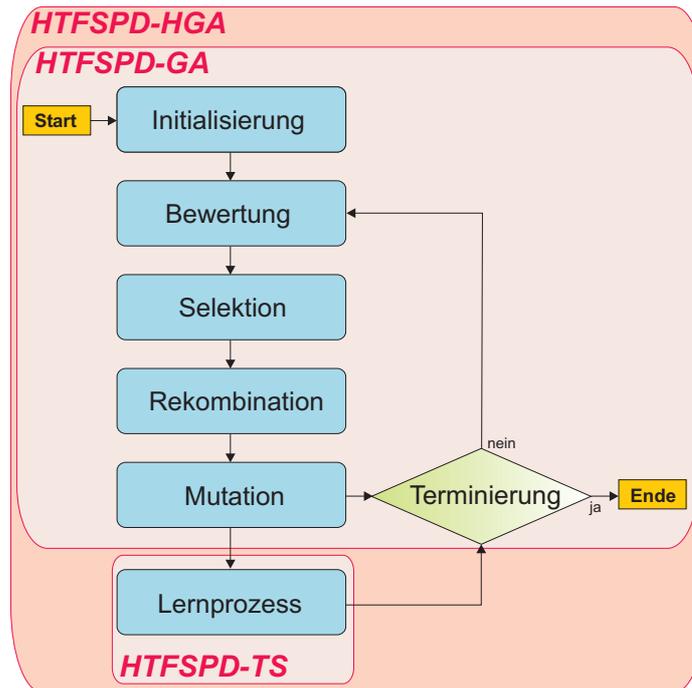


Abbildung 7.2: Der HTFSPD-HGA

Auf der anderen Seite geht eine zu starke Beschränkung der Rechenzeiten des HTFSPD-TS zulasten der erwünschten Suchintensivierung.

Um die beiden dargelegten Szenarien möglichst zu vermeiden, empfiehlt es sich, dem HTFSPD-TS zwar ausreichend Rechenzeit für eine relativ intensive Suche zur Verfügung zu stellen, ihn jedoch nicht in jeder Iteration des HTFSPD-GA, sondern nur in regelmäßigen Abständen aufzurufen. Im Folgenden wird dieser Turnus mit $turnus^{TS}$ bezeichnet (d. h. der Lernprozess wird regelmäßig jeweils nach $turnus^{TS}$ Iterationen durchgeführt). Zudem wird immer nur eine Lösung aus der aktuellen Population des HTFSPD-GA dem Lernprozess unterzogen, wobei sich die Auswahl dieser Lösung an dessen Güte orientiert. In den nächsten beiden Unterkapiteln 7.2.1 und 7.2.2 werden die Abläufe des HTFSPD-GA und des HTFSPD-TS im Detail vorgestellt.

7.2.1 Der genetische Algorithmus (HTFSPD-GA)

In Kapitel 4.3.2 wurde der grundlegende Ablauf genetischer Algorithmen bereits grob geschildert. Mit ihrem auf Methoden und Erkenntnissen der biologi-

schen Genetik, insbesondere der Evolutionstheorie, beruhenden Prinzip gehören sie bis heute in der Literatur zu den meist verbreitetsten Metaheuristiken. Einen guten Ein- bzw. Überblick in dieses Feld bieten Sivanandam und Deepa [78], Eiben und Smith [34] sowie das Werk von Goldberg [42]. Im Folgenden werden die einzelnen Teilschritte eines genetischen Algorithmus sowie deren jeweilige Realisierung im HTFSPD-GA detailliert erläutert.

Zunächst sind einige Begrifflichkeiten einzuführen. Im Zusammenhang mit genetischen Algorithmen spricht man des Öfteren von Populationen, Individuen und Generationen. In Kapitel 4.3.2 wurde eine populationsbasierte Metaheuristik bereits als eine auf mehreren Lösungen basierenden Heuristik definiert. Bei genetischen Algorithmen bezeichnet man eine einzelne Lösung i. d. R. als Individuum. Je nach Problemstellung werden diese durch bestimmte Zeichenfolgen repräsentiert. Bei einem wie in dieser Arbeit vorliegenden Reihenfolgeproblem entspricht diese Zeichenfolge einer bestimmten Reihenfolge (beispielsweise einer Produktionsreihenfolge). Ein Individuum ist somit als Lösungskandidat für die betrachtete Problemstellung zu verstehen, dessen Güte mit Hilfe einer Bewertungsfunktion ermittelt wird. Zudem gehört es immer einer aus mehreren Individuen bestehenden Population an. Da mit jeder Iteration neue Individuen entstehen und alte verworfen werden, ist eine Population im Laufe des Verfahrens regelmäßigen Änderungen unterworfen. Eine zu einem bestimmten Zeitpunkt vorliegende Population wird daher auch als Generation bezeichnet.

Auch wenn sie sich als universell einsetzbar erweisen, so ist die Effizienz eines genetischen Algorithmus sowohl von der Wahl der in den jeweiligen Phasen angewandten Verfahren bzw. Operatoren, als auch von deren Parametrierung abhängig. Mit diesen Steuerungselementen lässt sich dann auch das Suchverhalten beeinflussen. Im Folgenden werden die Verfahren, die beim HTFSPD-GA zum Einsatz kommen, näher vorgestellt. Zudem wird auf deren Bedeutung und auf aus der Parameterwahl jeweils resultierende Auswirkungen hingewiesen.

Initialisierung

Bereits dem ersten Schritt eines genetischen Algorithmus, der Initialisierung der ersten Population, kommt eine wichtige Rolle zu. Zunächst ist die Popula-

tionsgröße, also die Anzahl an Individuen, die eine Population enthalten soll, festzulegen. Dabei ist zu beachten, dass eine zu kleine Population die Gefahr einer vorzeitigen Konvergenz erhöht, d. h. die Individuen sich so sehr ähneln, dass das Verfahren in einem lokalen Optimum verharrt. Je größer die Population, desto eher lässt sich diese Gefahr vermeiden bzw. verzögern. Andererseits kann eine zu große Population kostbare Rechenzeit verschwenden.

In der Literatur werden verschiedene Populationsgrößen vorgeschlagen. Generell herrscht jedoch Einigkeit darüber, dass diese Größe problemspezifisch festzulegen ist. Es ist ebenfalls denkbar, die Populationsgröße variabel zu gestalten, d. h. sie im Laufe des Verfahrens regelmäßig anzupassen. Das HTFSPD-GA arbeitet, wie die meisten genetischen Algorithmen, jedoch mit einer über alle Iterationen fixen Populationsgröße.

Neben der Anzahl zu initialisierender Individuen ist auch die Art und Weise der Initialisierung von großer Bedeutung. So könnten prinzipiell alle Individuen zufällig erzeugt werden. Dies hat den Vorteil den Suchraum möglichst weiträumig zu gestalten, birgt jedoch die Gefahr, dass es, wenn überhaupt, sehr lange dauert, bis gute Lösungen gefunden werden. Es empfiehlt sich daher, zumindest einige Individuen der ersten Population mit bestimmten Startheuristiken zu initialisieren. Es sollte sich dabei um einfache, schnelle Verfahren handeln, da die Initialisierung sonst zu viel Rechenaufwand benötigt. Um auch hier wiederum die Gefahr einer vorzeitigen Konvergenz zu vermeiden, sollte ein Großteil der ersten Population jedoch auch aus zufällig erzeugten Individuen bestehen.

Im Folgenden wird eine Population durch die Menge Pop repräsentiert. Mit Pop^{it} wird dementsprechend die nach it Iterationen vorliegende Population bezeichnet, was der it -ten Generation entspricht. Auf die einzelnen Individuen wird dann mit Pop_{ind}^{it} verwiesen, wobei $ind = 1, \dots, |Pop^{it}|$ gilt und $|Pop^{it}|$ somit die Populationsgröße kennzeichnet. Ein Individuum ind repräsentiert einerseits eine bestimmte Wabenreihenfolge $folge^{ind}$. Um auch die *Hollomon*-Fahrweise zu integrieren, werden jeder Reihenfolgeposition $p = 1, \dots, n$ zudem bestimmte Temperatur-Zeit-Kombinationen $z_{folge_p^{ind}}$ zugeordnet, die einen Wert zwischen 0 und dem vorgegebenen z^{max} annehmen können.

An die bisherigen Erläuterungen schließt sich nun die Beschreibung der kon-

Tabelle 7.1: Vier Sortierverfahren zur Initialisierung des HTFSPD-GA

| | <i>Sort-1</i> | <i>Sort-2</i> | <i>Sort-3</i> | <i>Sort-4</i> |
|-------------|---------------|---------------|---------------|---------------|
| <i>wbht</i> | ↑ (1) | ↓ (1) | ↑ (2) | ↓ (2) |
| <i>ovz</i> | ↑ (2) | ↑ (2) | ↑ (1) | ↑ (1) |
| <i>lvo</i> | ↑ (3) | ↑ (3) | ↑ (3) | ↑ (3) |

kreten Initialisierung des HTFSPD-GA an. Zunächst kommen vier einfache Sortierverfahren zum Einsatz, die in Tabelle 7.1 zusammengefasst sind. Es handelt sich um die Verfahren *Sort-1*, *Sort-2*, *Sort-3* und *Sort-4*, denen allen die Sortierkriterien *wbht*, *ovz* und *lvo* gemein sind, wobei *lvo_j* auf die Position verweist, welche Wabe *j* im Stapel einnimmt (Lage von oben im Stapel). Die Verfahren unterscheiden sich lediglich in Richtung und Reihenfolge der Sortierung. Auf die Sortierrichtung weisen die Pfeile hin. Die Sortierreihenfolge ergibt sich aus den daneben in Klammern dargestellten Zahlen. Für *Sort-1* ergibt sich damit als erstes Kriterium eine aufsteigende Sortierung nach *wbht* und als zweites Kriterium eine aufsteigende Sortierung nach *ovz*. Bei *Sort-4* wird dagegen zunächst aufsteigend nach *ovz* und anschließend absteigend nach *wbht* sortiert. Die aufsteigende Sortierung nach *lvo* als drittes Kriterium ist allen Verfahren gemein. Es wird darauf hingewiesen, dass *Sort-1* und *Sort-2* (abgesehen von der Sortierung nach *lvo*) Bestandteil der (RBS)-Heuristik (vgl. Algorithmus 1) sind.

Der HTFSPD-GA sieht vor, jeweils ein Individuum mit einem der vier Sortierverfahren zu initialisieren. Aus bereits erläuterten Gründen werden zudem 96 Individuen zufällig erzeugt, sodass für die Populationsgröße $|Pop^0| = 100$ gilt. Abschließend sei erwähnt, dass die dargelegte Initialisierung nur die Wabenreihenfolgen $folge^{ind}$ (mit $ind = 1, \dots, |Pop^0|$) der Individuen betreffen. Die Temperatur-Zeit-Kombinationen der Waben werden bei jedem Individuum zunächst mit ihren ursprünglichen Vorgaben initialisiert, d.h. für alle $ind = 1, \dots, |Pop^0|$ und $p = 1, \dots, n$ gilt $z_{folge_p^{ind}} = 0$.

Bewertung

Die Bewertung bildet im Prinzip den einzigen problemspezifischen Schritt klassischer genetischer Algorithmen. Bezüglich der dieser Arbeit zugrunde-

liegenden Problemstellung wurde bereits ausführlich auf entsprechende Bewertungsmethoden hingewiesen. Prinzipiell können sowohl *Simulation-1* als auch *Simulation-2* oder die in Kapitel 7.5 noch vorzustellende *Simulation-3* im HTFSPD-GA verwendet werden. Als Bewertungsgrundlage dient in allen Fällen der Gesamtfertigstellungszeitpunkt C_{max} . Bei der Wahl der Bewertungsmethode sollte allerdings immer auch auf die jeweiligen Rechenzeiten geachtet werden, da ein genetischer Algorithmus regelmäßig und sehr häufig auf Bewertungen angewiesen ist.

Selektion

Die Selektion beinhaltet die Auswahl der Individuen, die „überleben“, d. h. in die nächste Generation übernommen werden sollen und somit prinzipiell für die anschließenden Schritte der Rekombination und Mutation in Frage kommen. Die Überlebenswahrscheinlichkeiten orientieren sich dabei i. d. R. an dem *Survival of the fittest* Prinzip. Demnach werden gute Individuen bei der Selektion bevorzugt behandelt, in der Hoffnung durch Weitervererbung der guten Gene bzw. Bestandteile bei der anschließenden Rekombination möglichst noch bessere Individuen zu generieren.

Die Wahl der Selektionsstrategie hat einen großen Einfluss auf das Suchverhalten eines genetischen Algorithmus. Je höher der Selektionsdruck, d. h. je eher die besseren Individuen überleben, desto eingeschränkter gestaltet sich der Suchraum. Umgekehrt bewirkt ein geringerer Selektionsdruck ein weiträumigeres Suchverhalten. Auf Vor- und Nachteile beider Aspekte wurde bereits eingegangen.

In der Literatur (siehe z. B. Sivanandam und Deepa [78]) finden sich verschiedene klassische Selektionsverfahren wie beispielsweise *Roulette Wheel Selection*, *Rank Selection*, *Tournament Selection* oder *Boltzmann Selection*. Für den HTFSPD-GA wird dagegen eine Variante des von Karci [53] vorgeschlagenen Verfahrens verwendet, welches auf dem Fortpflanzungsprinzip von Bienen Schwärmen basiert. Das Paarungsverhalten von Bienen ist meist so gestaltet, dass sich die Bienenkönigin mit mehreren Drohnen paart. Bezogen auf die Selektion eines genetischen Algorithmus bedeutet dies, dass das beste Individuum einer Generation (die Bienenkönigin) jeweils mit einem anderen In-

dividuum rekombiniert wird. Eine derartige Selektionsmethode schränkt den Suchraum dahingehend ein, dass die bei der Rekombination erzeugten Individuen immer bestimmte Bestandteile mit der Bienenkönigin gemein haben. Für viele Problemstellungen wäre diese Selektionsmethode, die im Folgenden als Bienen-Selektion bezeichnet wird, wahrscheinlich eine zu starke Einschränkung, für das HTFSPD erweist sie sich jedoch als gut geeignet. Grund ist die Tatsache, dass gute Lösungen beim HTFSPD immer einem aufsteigenden oder absteigenden Temperaturverlauf zumindest ähneln. Es kann aufgrund der Existenz der Temperaturlücken nicht sinnvoll sein häufig zwischen Temperaturen hin und her zu springen. Aufgrund dieser Problembeschaffenheit lässt sich der Suchraum hier relativ früh eingrenzen und demzufolge der Einsatz der Bienen-Selektion rechtfertigen. Für weitere Ansätze und Ideen, die im Zusammenhang mit Optimierungsverfahren und dem Verhalten von Bienenschwärmen stehen, sei auf die Studie von Karaboga und Akay [52] verwiesen.

Nach den bisherigen Erläuterungen lässt sich die Selektion des HTFSPD-GA nun im Detail zusammenfassen. Da das jeweils beste Individuum der aktuellen Generation immer als ein Elternpaar feststeht, sind nur noch die restlichen $\frac{|Pop^{it}|}{2}$ Elternpaare auszuwählen. Hierzu wird eine Turnierselektion (*Tournament Selection*) angewandt, die sich wie folgt darstellt:

1. Wähle zufällig $tsize^{GA}$ Individuen aus der aktuellen Generation (die Bienenkönigin ist nicht erlaubt).
2. Speichere das Individuum mit der besten Güte in einem Selektionspool.
3. Falls der Selektionspool weniger als $\frac{|Pop^{it}|}{2}$ Individuen enthält, fahre mit 1. fort, ansonsten ist die Selektion beendet.

Bei diesem Verfahren treten somit immer $tsize^{GA}$ Individuen gegeneinander an, wobei nur jeweils das beste in den Selektionspool aufgenommen wird und damit einen Elternteil für die Rekombination bildet. Der Parameter $tsize^{GA}$, der auch als Turniergröße bezeichnet wird, dient demnach als Werkzeug zur Steuerung des Selektionsdrucks. Da dieser durch die Bienenkönigin-Selektion bereits recht hoch ist, wird $tsize^{GA} = 2$ empfohlen.

Rekombination

Im Anschluss an die Selektion erfolgt die Rekombination (engl. *Crossover*) der ausgewählten Elternteile. Dabei werden jeweils zwei Elternteile auf bestimmte Weise miteinander gekreuzt, wodurch i. d. R. zwei neue Individuen (Kinder) entstehen. Die Rekombination stellt somit eine Art Informationsaustausch dar.

Die meisten Rekombinationsoperatoren, die in der Literatur vertreten sind, arbeiten mit sogenannten *crossover-points*, deren Auswahl zufallsbasiert erfolgt. Diese *crossover-points* dienen als Anhaltspunkte für die darauffolgende Kreuzung und bestimmen somit, welche Gene bzw. Bestandteile eines Elternteils vererbt werden. Bei der Wahl des Rekombinationsoperators ist darauf zu achten, dass nicht alle Verfahren auf jede Problemstellung anwendbar sind. So muss bei Reihenfolge- bzw. Permutationsproblemen beispielsweise gewährleistet sein, dass infolge der Kreuzung keine unzulässigen Individuen entstehen. In der Literatur haben sich daher spezifische Rekombinationsoperatoren für diese Problemklasse herausgebildet.

Kellegöz, Toklu und Wilson [54] haben die bekanntesten Rekombinationsoperatoren für Reihenfolgeprobleme auf ihre Effizienz hin untersucht, wobei ein *One Machine Total Weighted Tardiness Problem* zugrunde gelegt wurde. Die folgenden Verfahren wurden betrachtet:

- *Position-based Crossover Operator*,
- *Order-based Crossover Operator*,
- *One Point Crossover*,
- *Cycle Crossover Operator*,
- *Order Crossover*,
- *Linear Order Crossover Operator*,
- *Partially Mapped Crossover Operator*,
- *Edge Recombination Crossover Operator*,
- *Two Point Crossover Operator* (3 Varianten).

Die besten Ergebnisse wurden mit dem *Order-based Crossover Operator* (OBX) und dem *Position-based Crossover Operator* (PBX) erzielt. Hinsichtlich der Rechenzeiten ergaben sich nur geringfügige und somit vernachlässigbare Un-

terschiede zwischen den Verfahren. Aufgrund dieser Ergebnisse werden im HTFSPD-GA die Operatoren (OBX) und (PBX) zur Rekombination eingesetzt.

Abbildung 7.3 veranschaulicht die Vorgehensweise des *Order-based Crossover Operators* anhand eines kleinen Beispiels. Auf der linken Seite ist die erste Kreuzung der beiden Elternpaare, die Erzeugung des ersten Kindes, dargestellt. Auf der rechten Seite ist dementsprechend die zweite Kreuzung abgebildet. Elternteil 1 wird durch die Reihenfolge $folge^1 = (1, 2, 3, 4, 5, 6, 7, 8, 9)$ und Elternteil 2 durch die Reihenfolge $folge^2 = (2, 3, 5, 4, 1, 9, 8, 7, 6)$ repräsentiert. Für die Erzeugung des ersten Kindes wird zunächst Elternteil 1 betrachtet. Jedes Gen (Reihenfolgeposition) wird mit 50 %-iger Wahrscheinlichkeit ausgewählt bzw. (rot) markiert (siehe Waben 1, 4, 5 und 8). Anschließend werden diese Waben in Elternteil 2 gestrichen. Die restlichen Waben von Elternteil 2 werden eins zu eins in die Reihenfolge von Kind 1 übernommen. Die noch freien Positionen werden abschließend mit den in Elternteil 1 markierten Waben der entsprechenden Reihenfolge nach belegt. Die Erzeugung des zweiten Kindes verläuft analog, nur dass nun mit der Markierung der Gene von Elternteil 2 begonnen wird.

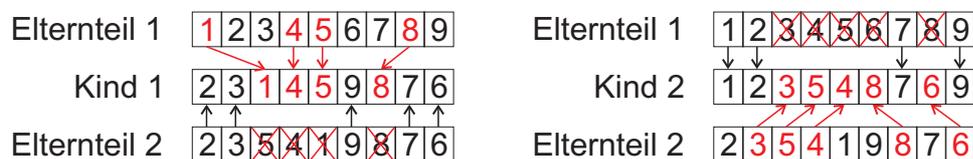


Abbildung 7.3: Der *Order-based Crossover Operator*

In Abbildung 7.4 wird der *Position-based Crossover Operator* anhand des gleichen Beispiels demonstriert. Auch hier werden die Gene von Elternteil 1 zunächst mit 50 %-iger Wahrscheinlichkeit markiert und anschließend an ihren jeweiligen Positionen in Elternteil 2 gestrichen. Aus Vergleichsgründen seien auch die Markierungen identisch wie zuvor. Die markierten Waben werden

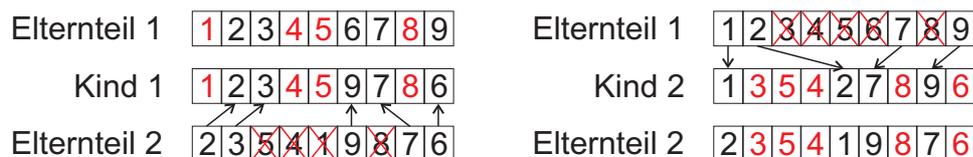


Abbildung 7.4: Der *Position-based Crossover Operator*

dann direkt an die jeweiligen Positionen in die Reihenfolge von Kind 2 übertragen. Schließlich werden die noch freien Positionen mit den noch in Elternteil 2 existierenden Waben belegt, wobei sich die Einsatzreihenfolge nach der Reihenfolge von Elternteil 2 richtet. Die Erzeugung des zweiten Kindes erfolgt auch hier analog, wobei nun wiederum zunächst mit Elternteil 2 begonnen wird.

Für den Einsatz im HTFSPD-GA müssen die beiden Operatoren aufgrund der Existenz der *Hollomon*-Fahrweise leicht angepasst werden. Es stellt sich nämlich die Frage welche z -Werte die neu erzeugten Individuen annehmen sollen, insbesondere wenn diese für gleiche Waben bei beiden Elternteilen voneinander abweichen. Als Antwort darauf wird eine einfache Regel implementiert, die besagt, dass die z -Werte der Waben für das i -te Kind sich nach den z -Werten der Waben von Elternteil i richten bzw. diesen entsprechen.

Konkret sieht die Rekombination des HTFSPD-GA für jedes Elternpaar vor, zwischen den Operatoren (OBX) und (PBX) jeweils zufällig zu wählen, wobei beiden gleich hohe Eintrittswahrscheinlichkeiten wk^{OBX} und wk^{PBX} zugrunde gelegt werden. Da es sich bei genetischen Algorithmen grundsätzlich empfiehlt, nie den kompletten Selektionspool einer Rekombination zu unterziehen, wird zudem mit wk^{NOX} eine Wahrscheinlichkeit dafür vorgegeben, dass keine Rekombination stattfindet. In diesem Fall werden die neuen Individuen quasi asexuell gebildet, d. h. Kind 1 entspricht einer Kopie von Elternteil 1 und Kind 2 analog von Elternteil 2. Als gute Vorgabe für die jeweiligen Eintrittswahrscheinlichkeiten hat sich bei ersten Experimenten die Konstellation $wk^{OBX} = wk^{PBX} = 45\%$ und demzufolge $wk^{NOX} = 10\%$ erwiesen.

Mutation

Die Mutation bildet den letzten Schritt klassischer genetischer Algorithmen. Unter ihr wird die zufällige Veränderung des Erbguts von Individuen verstanden. Während die Rekombination im Prinzip nur eine Weitervererbung bestehenden Erbguts darstellt, so sorgt die Mutation für die Ausbildung neuer Merkmale.

Erst durch das Zusammenspiel von Rekombination und Mutation können ge-

netische Algorithmen überhaupt effizient arbeiten. Ohne Mutation würde ein derartiges Verfahren nur die aus der Startpopulation bekannten Informationen nach und nach miteinander kombinieren. Der Suchbereich wäre in diesem Fall stark eingeschränkt. Ohne Rekombination artet der Suchvorgang dagegen in einer kompletten Zufallssuche aus. Allgemein sollten in einer Iteration nicht zu viele Individuen einer Mutation unterzogen werden, da ansonsten die Qualität der Population zu stark darunter leidet. Aus dem gleichen Grund empfiehlt es sich, im Rahmen der Mutation nur eher kleine Veränderungen an den Individuen vorzunehmen.

Auch für die Mutation bietet die Literatur eine Reihe von Operatoren, wobei einige davon wiederum nur speziell auf Reihenfolge- bzw. Permutationsprobleme anwendbar sind. Eiben und Smith [34] stellen diesbezüglich vier klassische Mutationsoperatoren vor:

- *Swap Mutation*,
- *Insert Mutation*,
- *Inversion Mutation*,
- *Scramble Mutation*.

Die Abbildungen 7.5(a) - 7.5(d) veranschaulichen das Prinzip der jeweiligen Verfahren anhand eines einheitlichen Beispiels. Vor der Mutation sei ein Individuum *ind* mit der Reihenfolge $folge^{ind} = (2, 3, 5, 4, 1, 9, 8, 7, 6)$ gegeben. Allen Verfahren ist im ersten Schritt die zufällige Auswahl zweier unterschiedlicher Reihenfolgepositionen (in den Abbildungen mit *a* und *b* gekennzeichnet) gemein. Aus Vergleichsgründen sei im Folgenden $a = 4$ und $b = 8$ für alle

Abbildung 7.5: Vier klassische Mutationsoperatoren



Operatoren angenommen.

Beim *Swap Mutation Operator* (SWM) (vgl. Abbildung 7.5(a)) werden die Waben an den Positionen a und b einfach getauscht. Die restliche Reihenfolge ist davon nicht betroffen. Der *Insert Mutation Operator* (INSM) (vgl. Abbildung 7.5(b)) entnimmt zunächst die Wabe, die sich an Reihenfolgeposition b befindet. Diese setzt er dann vor die Reihenfolgeposition a , falls $a < b$ gilt oder hinter die Reihenfolgeposition a , falls $a > b$ gilt. Die Waben, die sich vor der Mutation zwischen a und b befanden, müssen dementsprechend einrücken. An ihrer Reihenfolge ändert sich jedoch nichts. Beim *Inversion Mutation Operator* (INVM) (vgl. Abbildung 7.5(c)) ist dies jedoch der Fall. Hier wird die Reihenfolge der Waben, die sich an und zwischen den Positionen a und b befinden, einfach gedreht bzw. gespiegelt. Die gleichen Positionen werden auch beim *Scramble Mutation Operator* (SCRM) (vgl. Abbildung 7.5(d)) verändert, wobei hier die Reihenfolge einfach zufällig gemischt wird.

Man stellt sofort fest, dass die Operatoren (INVM) und (SCRM) wesentlich größere Veränderungen zur Folge haben als dies bei den Operatoren (SWM) und (INSM) der Fall ist. Prinzipiell sind (SWM) und (INSM) daher eher geeignet. Da die Mutation jedoch auch ein Instrument darstellt, um eine vorzeitige Konvergenz zu verhindern, können größere Veränderungen an den Individuen gelegentlich von Vorteil sein. Aus diesem Grund kommen beim HTFSPD-GA alle vier vorgestellten Mutationsoperatoren zum Einsatz. Deren Eintrittswahrscheinlichkeiten seien mit wk^{SWM} , wk^{INSM} , wk^{INVM} und wk^{SCRM} definiert. Zusätzlich wird für die Wahrscheinlichkeit, dass keine Mutation stattfindet, der Parameter wk^{NOM} eingeführt.

Es lässt sich schnell feststellen, dass sich die vier vorgestellten Mutationsoperatoren alle nur auf die Wabenreihenfolgen der Individuen beziehen, dagegen die z -Werte davon unberührt bleiben. Aus diesem Grund wird mit dem *Holomon Mutation Operator* (HOM) zusätzlich ein problemspezifisches Verfahren integriert mit dem sich auch diese Werte variieren lassen. Beim (HOM) wird im ersten Schritt eine Reihenfolgeposition zufällig ausgewählt. Anschließend wird für die zugehörige Wabe zufällig ein Wert zwischen 0 und z^{max} ermittelt, welcher dann den neuen z -Wert dieser Wabe bildet. Dabei wird jedoch wie bereits beim *HTFSPD-HF-Branch&Bound*-Verfahren (vgl. Algorithmus 8) im-

mer darauf geachtet, dass die daraus resultierende Ofentemperatur nicht unter die Minimaltemperatur $wbht_{min}^{Ofen}$ der im betrachteten Datensatz enthaltenen Waben fallen darf. Abbildung 7.6 zeigt dies anhand eines kleinen Beispiels. Anstatt der Wabenreihenfolge sind hier die den jeweiligen z -Werten entsprechenden Ofentemperaturen abgebildet. Die rote Markierung verweist auf die zufällig ausgewählte Reihenfolgeposition, für dessen Wabe ein neuer z -Wert ermittelt werden soll. In diesem Fall darf die Ofentemperatur dieser Wabe nicht auf unter $600\text{ }^\circ\text{C}$ ($= wbht_{min}^{Ofen}$) geändert werden, auch wenn mit der Vorgabe von z^{max} prinzipiell größere Temperatursenkungen möglich wären.

| | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| vor Mutation | 640 | 620 | 610 | 600 | 600 | 600 | 600 | 610 | 610 |
| nach Mutation | 640 | 620 | 610 | 600 | 600 | 600 | 600 | 600 | 610 |

Abbildung 7.6: Der *Hollomon Mutation Operator*

Erst mit der Einführung des *Hollomon Mutation Operators* ist die *Hollomon-Fahrweise* im HTFSPD-GA integriert. Die Eintrittswahrscheinlichkeit dieses Operators wird mit wk^{HOM} bezeichnet und sollte prinzipiell nicht zu hoch gewählt werden, da dieses Verfahren teilweise zu sehr starken Veränderungen am jeweiligen Individuum führen kann. Die besten Ergebnisse konnten im Rahmen einiger Versuchsexperimente mit der Parameterkonstellation $wk^{NOM} = 25\%$, $wk^{SWM} = wk^{INSM} = 30\%$ und $wk^{INVM} = wk^{SCRM} = wk^{HOM} = 5\%$ erzielt werden.

Terminierung

Die Terminierung, bzw. die Festlegung der diesbezüglichen Kriterien, ist allgemein ein sehr wichtiger und meist schwer zu entscheidender Teilschritt bei der Entwicklung von Metaheuristiken. Talbi [83] unterscheidet in diesem Zusammenhang statische und adaptive Kriterien. Ein klassisches Beispiel für ein statisches Kriterium ist der Abbruch nach Ablauf eines vorher festgelegten Zeitlimits. Auch die Vorgabe einer maximalen Iterationsanzahl oder eines Limits für die Anzahl durchgeführter Bewertungen gehören dieser Gruppe an. Im Gegensatz zu statischen sind adaptive Kriterien nicht unabhängig vom Suchverlauf des jeweiligen Verfahrens. Hier setzt ein Abbruch beispielsweise ein, sobald die letzte erzielte Verbesserung eine bestimmte Zeit oder Iterations-

anzahl zurückliegt. Falls eine untere Grenze für die jeweilige Problemstellung vorliegt, ist auch ein Abbruch bei Erreichen einer bestimmten Nähe zu diesem Grenzwert denkbar.

Prinzipiell können statische und adaptive Abbruchkriterien auch gemeinsam implementiert werden. So auch beim HTFSPD-GA, dem ein statisches Zeitlimit $time^{GA}$ und eine maximale Anzahl an Iterationsschritten ohne Verbesserung $impit^{max,GA}$ zugrunde gelegt wird. An dieser Stelle sei darauf hingewiesen, dass die Abbruchkriterien für den HTFSPD-HGA mit $time^{HGA}$ und $impit^{max,HGA}$ analog definiert werden. Auf die Belegung dieser Parameter und die dabei zu beachtenden Aspekte wird bei der Vorstellung einiger Rechenergebnisse in Unterkapitel 7.2.3 eingegangen.

7.2.2 Der *Tabu Search* (HTFSPD-TS)

Das auf Glover [39] zurückgehende Konzept des *Tabu Search* Algorithmus stellt bis heute im Umgang mit kombinatorischen Optimierungsproblemen eine viel verwendete Metaheuristik dar. Wie bereits in Kapitel 4.3.2 beschrieben, handelt es sich bei *Tabu Search* um eine einzellösungsbasierte Metaheuristik, die auf dem Prinzip der Nachbarschaftssuche mit integriertem Gedächtnis basiert. Einen guten Einblick in dieses Feld liefern beispielsweise Glover und Laguna [40] oder auch Gendreau [38].

Es wurde bereits darauf hingewiesen, dass *Tabu Search* als einzellösungsbasierte Metaheuristik eher einer intensivierenden als einer diversifizierenden Suche nachkommt. Grundsätzlich lässt sich das Suchverhalten, wie auch bei genetischen Algorithmen, jedoch durch bestimmte Parametereinstellungen steuern. Auf diese Möglichkeiten wird in den weiteren Ausführungen näher eingegangen.

Der Ablauf eines *Tabu Search* stellt sich i. d. R. wie folgt dar:

1. Initialisierung,
2. Nachbarschaftssuche,
3. Prüfung und Bewertung der Nachbarschaft,

4. Auswahl einer neuen Lösung,
5. Gedächtnisverwaltung,
6. Terminierung, falls Abbruchkriterium erfüllt; sonst weiter mit Schritt 2.

Im Folgenden wird die allgemeine Bedeutung der einzelnen Phasen sowie deren jeweilige Ausgestaltung im HTFSPD-TS im Detail vorgestellt.

Initialisierung

Die Initialisierung hat bei *Tabu Search* Algorithmen einen ausschlaggebenden Charakter. Da das Verfahren auf einer einzelnen Lösung basiert, wird der Suchraum hier zu Beginn viel stärker in einen bestimmten Bereich gelenkt als dies z. B. bei genetischen Algorithmen der Fall ist. Bei schlechten Startlösungen kann es somit, wenn überhaupt, sehr lange dauern, bis ein Bereich mit guten Lösungen gefunden wird. Im Umkehrschluss bedeutet dies nicht, dass bessere Startlösungen immer auch zu letztendlich besseren Lösungen führen.

Prinzipiell eignen sich die im Zusammenhang mit dem HTFSPD-GA vorgestellten Sortierverfahren *Sort-1* - *Sort-4* (vgl. Tabelle 7.1) ebenso für den HTFSPD-TS. Da dieser im Rahmen des HTFSPD-HGA jedoch im Sinne eines Lernprozesses dem HTFSPD-GA nachgeschaltet angeordnet ist, stellt sich die Frage der Initialisierung für den *Tabu Search* hier nur indirekt. Es gilt lediglich ein Individuum aus der einem Lernprozess zu unterziehenden Generation auszuwählen, welches dann als Startlösung für den HTFSPD-TS dient.

Die Auswahl des Individuums sollte sich aufgrund der vorherigen Erläuterungen an der Güte der einzelnen Individuen orientieren. Im HTFSPD-TS wird daher wiederum eine Turnierselektion (vgl. Kapitel 7.2.1) angewandt. Der entsprechende Parameter für die Turniergröße sei mit $tsize^{TS}$ bezeichnet. Nach einigen Testläufen hat sich ergeben, dass mit $tsize^{TS} = 4$ gute Ergebnisse erzielt werden.

Nachbarschaftssuche

Die Nachbarschaftssuche stellt den eigentlichen Suchprozess dar. Der Einsatz von *Tabu Search* setzt grundsätzlich die Existenz einer Nachbarschaftsstruk-

tur im gesamten Lösungsraum voraus, d. h. für jede beliebige zulässige Lösung muss eine nicht leere Menge an Nachbarschaftslösungen existieren. Ein elementarer Schritt bei der Entwicklung eines *Tabu Search* ist daher die Definition geeigneter Nachbarschaftsbeziehungen. In diesem Zusammenhang stellt sich bereits die Frage nach der Suchstrategie, da die Nachbarschaft einer Lösung im Prinzip den Suchraum der aktuellen Iteration bildet. Aufgrund der Bedeutung der Nachbarschaftsdefinition ist es an dieser Stelle grundsätzlich sinnvoll, problemspezifisches Wissen (falls vorhanden) einzubinden.

Der HTFSPD-TS basiert auf einer in Abbildung 7.7 anhand eines Beispiels veranschaulichten Nachbarschaftsdefinition. Die aktuelle Lösung *ind* sei durch Reihenfolge $folge^{ind} = (2, 7, 5, 1, 6, 9, 8, 3, 4)$ und den zugehörigen Temperatur-Zeit-Kombinationen $z_{folge^{ind}} = (0, 2, 0, 0, 1, 0, 0, 1, 1)$ gegeben, wobei $z^{max} = 2$ gelten soll. Um die Nachbarschaft zu bestimmen, wird zunächst eine Reihenfolgeposition ausgewählt. Im abgebildeten Beispiel sei dies die 5. Position mit

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| aktuelle Lösung | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>6</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 6 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 6 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nachbarschaft | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>6</td><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 6 | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>6</td><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>1</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 6 | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>6</td><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>2</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 6 | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| 6 | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>6</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 6 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>6</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>1</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 6 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>6</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 6 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 6 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 6 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 6 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>6</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 6 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>6</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 6 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>6</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 6 | 5 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 7 | 6 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 6 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 6 | 5 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>6</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 6 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>6</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 6 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>6</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 6 | 1 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 1 | 1 |
| 2 | 7 | 5 | 6 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 6 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 6 | 1 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 2 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>6</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 6 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>6</td><td>9</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 6 | 9 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 6 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 6 | 9 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>6</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 6 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>6</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 6 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>6</td><td>8</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>2</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 6 | 8 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 1 |
| 2 | 7 | 5 | 1 | 9 | 6 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 6 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 6 | 8 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>6</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 6 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>6</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 6 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>6</td><td>3</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 6 | 3 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 1 |
| 2 | 7 | 5 | 1 | 9 | 8 | 6 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 8 | 6 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 8 | 6 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>6</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 6 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>6</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 6 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>6</td><td>4</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 6 | 4 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| 2 | 7 | 5 | 1 | 9 | 8 | 3 | 6 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 8 | 3 | 6 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 8 | 3 | 6 | 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td><td>6</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 6 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td><td>6</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 6 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>2</td><td>7</td><td>5</td><td>1</td><td>9</td><td>8</td><td>3</td><td>4</td><td>6</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>2</td></tr> </table> | 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 6 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 7 | 5 | 1 | 9 | 8 | 3 | 4 | 6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 2 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Abbildung 7.7: Nachbarschaftsbeziehungen im HTFSPD-TS (Beispiel)

$folge_5^{ind} = 6$ und $z_{folge_5^{ind}} = 1$. Die Nachbarschaftslösungen stellen sich dann so dar, dass Wabe 6 jeweils an allen Positionen $p = \{1, \dots, n\}$ mit jeweils allen z -Werten $z = \{0, \dots, z^{max}\}$ angeordnet bzw. belegt wird. Um jedoch keine Kopie der Ausgangslösung zu erstellen, darf der neue Wert für $z_{folge_5^{ind}}$ nicht dem ursprünglichen Wert entsprechen. Zudem ist auch hier wiederum darauf zu achten, dass keine z -Werte zugeordnet werden dürfen, die zu Ofentemperaturen unter $wbht_{min}^{Ofen}$ führen würden.

Es wurde bereits darauf hingewiesen, problemspezifisches Wissen in der Nachbarschaftsdefinition berücksichtigen zu wollen. Dies geschieht im HTFSPD-TS im Zusammenhang mit der Auswahl der Reihenfolgeposition. Hier erscheint es sinnvoll, ursprünglich eher schlecht platzierte Waben mit einer höheren Wahrscheinlichkeit zu wählen. Die Frage ist nun, wie „schlecht platziert“ definiert werden kann. Hierfür bieten sich die Wartezeiten der Waben vor dem Ofen an, die sich aus den in der jeweiligen Materialflusssimulation des Ofens ermittelten Ofeneinfahrzeiten ergeben. Die Wartezeit $warte_j$ einer Wabe j mit Reihenfolgeposition p ergibt sich wie folgt:

$$warte_{folge_p^{ind}} = \begin{cases} FAZO_{folge_p^{ind}} - FAZO_{folge_{p-1}^{ind}}, & p = \{2, \dots, n\} \\ FAZO_{folge_p^{ind}}, & p = 1 \end{cases} \quad (7.1)$$

Die letztendliche Auswahl der Reihenfolgeposition im Rahmen der Nachbarschaftssuche wird dann wiederum anhand einer Turniererlektion ermittelt, wobei Waben mit höheren Wartezeiten gegen Waben mit niedrigeren Wartezeiten „gewinnen“. Die entsprechende Turniergröße sei mit $tsize^{nbsuch}$ bezeichnet. Es empfiehlt sich, diesen Parameter problemgrößenabhängig festzulegen, wobei folgende Formel vorgeschlagen wird:

$$tsize^{nbsuch} = \left\lceil \frac{n}{nbsuch^*} \right\rceil. \quad (7.2)$$

Mit $nbsuch^* = 4$ wurden im Rahmen einiger Parametertests die besten Ergebnisse erzielt.

Prüfung und Bewertung der Nachbarschaft

Im nächsten Schritt werden die erzeugten Nachbarschaftslösungen zunächst einer Tabu-Prüfung und einer anschließenden Bewertung unterzogen. Für die Tabu-Prüfung ist die sogenannte Tabu-Liste ausschlaggebend. Sie stellt einen Informationsspeicher des bisherigen Suchverlaufs über einen bestimmten Zeitraum dar. Die Tabu-Liste enthält vorangegangene Lösungen, Teile davon oder auch nur bestimmte getätigte Züge. Trifft für eine Nachbarschaftslösung das Tabu-Kriterium zu, d.h. ist die Lösung, ein Teil davon oder der getätigte Zug in der Tabu-Liste enthalten, so darf diese Nachbarschaftslösung zu diesem Zeitpunkt nicht gewählt werden. Bei manchen *Tabu Search* Algorithmen werden jedoch bestimmte Aspirationskriterien festgelegt, bei deren Erfüllung auch prinzipiell tabuisierte Lösungen erlaubt werden. Auf die Bedeutung und Gestaltungsmöglichkeiten der Tabu-Liste wird bei der späteren Erläuterung der Gedächtnisverwaltung noch näher eingegangen.

Nach Abschluss der Tabu-Prüfung werden die verbliebenen Nachbarschaftslösungen schließlich bewertet. Wie auch bei den genetischen Algorithmen steckt hierin der eigentliche problemspezifische Teil des Verfahrens. Auch für den HTFSPD-TS können hier sowohl *Simulation-1* als auch *Simulation-2* oder die in Kapitel 7.5 noch vorzustellende *Simulation-3* als Bewertungsgrundlage verwendet werden, wobei ebenfalls auf die erforderlichen Rechenzeiten zu achten ist.

Auswahl einer neuen Lösung

Nachdem Gültigkeit und Güte der einzelnen Nachbarschaftslösungen nun bekannt sind, ist die Auswahl der Lösung erforderlich, mit welcher die Suche fortgesetzt werden soll. Häufig wird an dieser Stelle prinzipiell die beste gefundene Lösung gewählt. Diese Strategie birgt jedoch die Gefahr, dass die Suche sich immer wieder im Kreis dreht und der Algorithmus somit in einem lokalen Optimum stagniert. Es kann daher strategisch durchaus sinnvoll sein, bei der Auswahl der neuen Lösung auch schlechtere Kandidaten aus der Nachbarschaftsmenge in Betracht zu ziehen. Grundsätzlich sollten bessere Lösungen jedoch bevorzugt werden, da die Suche ansonsten schnell in zu schlechte Bereiche gelenkt wird.

Beim HTFSPD-TS wird die Auswahl der neuen Lösung aus der Nachbarschaft wieder einer Turnierselektion überlassen. Da die Nachbarschaft nicht immer die gleiche Größe besitzt, ist es auch hier wiederum sinnvoll, eine variable Turniergröße zu verwenden. Diese sei mit $tsize^{nbwahl}$ bezeichnet und wie folgt berechnet:

$$tsize^{nbwahl} = \left\lceil \frac{|nb|}{nbwahl^*} \right\rceil. \quad (7.3)$$

Dabei stellt nb die Nachbarschaftsmenge dar, sodass $|nb|$ die Anzahl der prinzipiell auswählbaren Nachbarschaftslösungen widerspiegelt. Mit $nbwahl^*$ lässt sich dann der gewünschte Selektionsdruck einstellen, wobei sich die Vorgabe $nbwahl^* = 4$ im Rahmen einiger Testläufe als vielversprechend erwiesen hat.

Gedächtnisverwaltung

Mit der Verwaltung eines Gedächtnisses unterscheidet sich *Tabu Search* von anderen Nachbarschaftssuchverfahren. Den Kern dieses Gedächtnisses bilden die Informationen aus der Tabu-Liste. Da sich dieser Speicher, je nach Größe, immer auf eine gewisse Anzahl zuletzt getätigter Schritte bezieht, wird die Tabu-Liste häufig auch mit einem Kurzzeitgedächtnis verglichen. Heute tendieren viele *Tabu Search* Algorithmen dazu, zusätzlich ein Mittel- und ein Langzeitgedächtnis zu integrieren (vgl. z. B. Talbi [83]). Im Folgenden werden die mit den einzelnen Gedächtnistypen verbundenen Zielsetzungen sowie deren konkrete Ausgestaltung im HTFSPD-TS vorgestellt.

Das Kurzzeitgedächtnis bzw. die Tabu-Liste soll hauptsächlich Zyklen im Suchprozess und damit auch in gewissem Maße das Verharren in lokalen Optima verhindern. Wie bereits erwähnt, kann eine Tabu-Liste beispielsweise ganze Lösungen, Teillösungen oder getätigte Züge enthalten. Folglich können prinzipiell auch mehrere Tabu-Listen gleichzeitig verwaltet werden. Allgemein gilt, dass je mehr Informationen im Kurzzeitgedächtnis gespeichert werden, desto weiträumiger wird der Suchraum durchsucht. Umgekehrt hat ein kleines Kurzzeitgedächtnis i. d. R. ein intensiveres Suchverhalten zur Folge.

Die Tabu-Liste des HTFSPD-TS ist so gestaltet, dass immer ganze Lösungen

gespeichert werden. Somit sind potentielle Nachbarschaftslösungen nur tabu, wenn sie identisch mit einer in der Tabu-Liste existierenden Lösung sind. Im Folgenden sei die Tabu-Liste durch die Menge $tabulist$ repräsentiert. Die maximale Größe der Tabu-Liste $tabulist^{max}$ wird abhängig von der Größe des vorliegenden Problems mit Hilfe von Formel 7.4 ermittelt:

$$tabulist^{max} = n \cdot (1 + z^{max}). \quad (7.4)$$

Die zusätzliche Integration eines mittelfristigen Gedächtnisses dient der Suchintensivierung. Es stellt eine Art Qualitätsspeicher dar, da es i. d. R. die besten gefundenen Lösungen verwaltet. Die Idee dahinter besteht darin, den Suchprozess mit fortschreitender Zeit des Verfahrens zu intensivieren, indem man wieder zu guten bereits gefundenen Lösungen „zurückkehrt“ und deren Nachbarschaften erneut untersucht.

Im HTFSPD-TS wird ein solcher Qualitätsspeicher ebenfalls integriert. Er wird durch die Menge $bestlist$ repräsentiert, in der jeweils die 10 besten bisher gefundenen Lösungen gespeichert werden ($bestlist^{max} = 10$). Zudem wird mit $restart^{best}$ ein Parameter eingeführt, der den Zeitpunkt bestimmt, wann die Suche mit einer Lösung aus der Menge $bestlist$ fortgesetzt bzw. neu gestartet wird. Dieser Fall tritt genau dann ein, wenn die letzte gefundene Verbesserung $restart^{best}$ Iterationen zurückliegt. Die Entscheidung, welche Lösung aus $bestlist$ gewählt werden soll, erfolgt jeweils zufällig gleichverteilt. Der Parameter $restart^{best}$ wird zunächst mit 100 (Iterationen) initialisiert. Um die Suche im Laufe des Verfahrens stetig zu intensivieren werden einige Parameter des HTFSPD-TS bei jedem Neustart leicht angepasst. So wird

- $tabulist^{max}$ auf $\max\{2, tabulist^{max} - (1 + z^{max})\}$ und
- $restart^{best}$ auf $\max\{10, restart^{best} - 10\}$

gesetzt. Zudem wird die Tabu-Liste geleert.

Die Integration eines Langzeitgedächtnisses verfolgt im Gegensatz zum mittelfristigen Gedächtnis das Ziel, den Suchprozess zu diversifizieren. Die Idee besteht darin, im Anschluss an eine Suchintensivierung die Suche wieder in noch nicht betrachtete Bereiche des Lösungsraums zu leiten. Die Verfolgung dieser

Strategie ist bei der Verwendung von *Tabu Search* grundsätzlich zu empfehlen. Da der HTFSPD-TS im Verlauf des HTFSPD-HGA jedoch bereits mehrmals mit jeweils unterschiedlichen Individuen aufgerufen wird, ist die Integration eines Langzeitgedächtnisses hier nicht mehr erforderlich. Zudem würden daraus höhere Rechenzeiten für den HTFSPD-TS resultieren, welche aus bereits erläuterten Gründen möglichst vermieden werden sollten.

Terminierung

Die im Rahmen der Beschreibung des HTFSPD-GA möglichen Kriterien bezüglich der Terminierung des Verfahrens eignen sich im Prinzip alle auch für den Einsatz im HTFSPD-TS. Aus diesem Grund wird auch diesem ein statisches Zeitlimit $time^{TS}$ und eine maximale Anzahl an Iterationsschritten ohne Verbesserung $impit^{max,TS}$ zugrunde gelegt. Die Belegung der Terminierungsparameter wird im Rahmen der Präsentation einiger Rechenergebnisse im folgenden Unterkapitel vorgestellt.

7.2.3 Rechenergebnisse

Mit der Entwicklung des HTFSPD-HGA steht nun ein heuristischer Ansatz für die betrachtete Problemstellung zur Verfügung. Dessen Effizienz in Bezug auf die reine Reihenfolgeoptimierung lässt sich am besten durch einen direkten Vergleich mit dem *HTFSPD-Branch&Bound*-Verfahren nachweisen. Die Ergebnisse dieses Vergleichs für die bereits mehrfach herangezogenen Datensätze sind in Tabelle 7.2 dargestellt.

Die Ergebnisse des *HTFSPD-Branch&Bound*-Verfahrens sind bereits aus vorherigen Untersuchungen bekannt und nur wieder der Übersicht wegen zusammengetragen. Es sei darauf hingewiesen, dass es sich hierbei um die Ergebnisse handelt, die ohne Berücksichtigung der *Hollomon*-Fahrweise entstanden. Da der HTFSPD-HGA an vielen Stellen stochastischen Einflüssen ausgesetzt ist, wurde der Algorithmus für jeden Datensatz jeweils fünfmal durchgeführt. Die in den letzten beiden Spalten dargestellten Ergebnisse des HTFSPD-HGA entsprechen demnach Durchschnittswerten. Die Parameter wurden weitestgehend so gewählt, wie sie aus der Beschreibung des HTFSPD-HGA (bzw.

Tabelle 7.2: Vergleich der Rechenergebnisse des HTFSPD-HGA mit denen des *HTFSPD-Branch&Bound*-Verfahrens

| Daten- satz | f_{templ} | AnzBR | BB-PFDN | | | HTFSPD-HGA | |
|---------------------------|-------------|-------|-----------|--------|-----------|---------------------|------------------|
| | | | C_{max} | Gap | Zeit | $\emptyset C_{max}$ | \emptyset Zeit |
| <i>dat3</i> | 1000 | 2 | 551 | 0.00 | < 1.00 | 551.00 | 30.00 |
| | 600 | 2 | 420 | 0.00 | < 1.00 | 420.00 | 30.20 |
| <i>dat5</i> | 1000 | 2 | 849 | 0.00 | < 1.00 | 849.00 | 79.00 |
| | 600 | 2 | 849 | 0.00 | < 1.00 | 849.00 | 78.80 |
| <i>dat8</i> | 1000 | 2 | 620 | 0.00 | < 1.00 | 620.00 | 208.20 |
| | 600 | 2 | 572 | 0.00 | 2.04 | 572.00 | 208.40 |
| <i>dat10</i> | 1000 | 2 | 721 | 0.00 | 20.56 | 721.00 | 437.20 |
| | 600 | 2 | 659 | 0.00 | 41.88 | 659.00 | 411.60 |
| <i>dat12</i> | 1000 | 2 | 929 | 0.00 | 5742.91 | 929.00 | 582.20 |
| | 600 | 2 | 787 | 0.00 | 1272.34 | 787.00 | 600.00 |
| <i>dat15</i> | 1000 | 2 | 919 | 31.77 | 18000.00 | 919.00 | 600.00 |
| | 600 | 2 | 878 | 33.03 | 18000.00 | 888.20 | 600.00 |
| <i>dat20</i> | 1000 | 2 | 1176 | 100.00 | 18000.00 | 1170.00 | 600.00 |
| | 600 | 2 | 1135 | 100.00 | 18000.00 | 988.20 | 600.00 |
| <i>dat30</i> | 1000 | 2 | 2279 | 100.00 | 18000.00 | 1416.00 | 600.00 |
| | 600 | 2 | 2279 | 100.00 | 18000.00 | 1323.20 | 600.00 |
| <i>dat10_{vw}</i> | 1000 | 2 | 386 | 0.00 | 845.84 | 386.00 | 407.80 |
| | 600 | 2 | 386 | 0.00 | 852.59 | 386.00 | 415.80 |
| | 1000 | 4 | 274 | 0.00 | 322.66 | 274.00 | 302.40 |
| | 600 | 4 | 252 | 0.00 | 133.57 | 252.00 | 312.20 |
| | 1000 | 6 | 274 | 0.00 | 317.37 | 274.00 | 280.40 |
| | 600 | 6 | 252 | 0.00 | 87.14 | 252.00 | 281.60 |
| <i>dat16_{vw}</i> | 1000 | 2 | 792 | 100.00 | 18000.00 | 766.40 | 600.00 |
| | 600 | 2 | 792 | 100.00 | 18000.00 | 765.40 | 600.00 |
| | 1000 | 4 | 419 | 100.00 | 18000.00 | 387.00 | 600.00 |
| | 600 | 4 | 419 | 100.00 | 18000.00 | 387.00 | 600.00 |
| | 1000 | 6 | 329 | 100.00 | 18000.00 | 322.00 | 600.00 |
| | 600 | 6 | 306 | 100.00 | 18000.00 | 295.00 | 600.00 |
| <i>dat25_{vw}</i> | 1000 | 2 | 1292 | 100.00 | 18000.00 | 1232.20 | 600.00 |
| | 600 | 2 | 1292 | 100.00 | 18000.00 | 1220.20 | 600.00 |
| | 1000 | 4 | 906 | 100.00 | 18000.00 | 879.20 | 600.00 |
| | 600 | 4 | 872 | 100.00 | 18000.00 | 841.80 | 600.00 |
| | 1000 | 6 | 870 | 100.00 | 18000.00 | 828.80 | 600.00 |
| | 600 | 6 | 809 | 100.00 | 14201.20* | 769.60 | 600.00 |

*Abbruch aufgrund des Speicherplatzbedarfs

des HTFSPD-GA und des HTFSPD-TS) hervorgehen. Ausgenommen hiervon ist die Festlegung von $w_k^{HOM} = 0$, $w_k^{NOM} = 30$ und $z^{max} = 0$, da diese Tests zunächst nur für die Bewertung der reinen Reihenfolgebildung ausge-

legt sind. Die Zeitlimits wurden mit $time^{GA} = 600$ und $time^{TS} = 30$ sowie $impit^{max,GA} = 2000$ und $impit^{max,TS} = 1000$ auf die betrieblichen Möglichkeiten bzw. Gegebenheiten angepasst. Des Weiteren wurde $turnus^{TS} = 20$ gesetzt.

Aus dem Vergleich wird ersichtlich, dass der HTFSPD-HGA bei allen Instanzen, für die das *HTFSPD-Branch&Bound*-Verfahren die optimale Lösung garantieren kann, diese Lösungen ebenfalls gefunden hat (vgl. *dat3* - *dat12* und *dat10_{vw}*). Der wahre Vorteil des HTFSPD-HGA zeigt sich jedoch bei den größeren Datensätzen. Die Lösungen, die das *HTFSPD-Branch&Bound*-Verfahren nach fünfstündiger Rechenzeit ermittelt hat, liegen hier teilweise deutlich (bei Datensatz *dat30* und $f_{templ} = 600$ um ca. 72,2 %) über denen, die der HTFSPD-HGA nach 10 min ausgibt. Alles in allem hat der HTFSPD-HGA mit Ausnahme von einer Instanz die Testläufe dominiert. Lediglich bei Datensatz *dat15* und $f_{templ} = 600$ lagen die Lösungen im Mittel um ca. 1,1 % über der mit dem *HTFSPD-Branch&Bound*-Verfahren ermittelten Lösung. Letztere wurde von dem HTFSPD-HGA zwar prinzipiell auch gefunden, jedoch nur in 3 der 5 Fälle.

Die Effizienz des HTFSPD-HGA in Bezug auf die reine Reihenfolgeoptimierung konnte nachgewiesen werden. Es stellt sich nun die Frage, wie sich die Heuristik bei zusätzlicher Berücksichtigung der *Hollomon*-Fahrweise verhält. Hierzu bietet sich ein Vergleich mit dem in Kapitel 6.2.2 entwickelten *HTFSPD-HF-Branch&Bound*-Verfahren an. Da zu diesem Zeitpunkt noch *Simulation-1* zugrunde lag, wurde der HTFSPD-HGA für die folgenden Tests ebenfalls mit dieser Bewertungsmethode ausgelegt. Zudem wurde sich auf den Fall $z^{max} = 3$ beschränkt. Die Ergebnisse sind aus Tabelle 7.3 zu entnehmen.

Auch bei diesem Vergleich lässt sich feststellen, dass der HTFSPD-HGA mit Ausnahme von einer Instanz die Testläufe dominiert. Lediglich bei Datensatz *dat12* unter Annahme von $f_{templ} = 1000$ ergaben sich im Mittel um ca. 1,5 % abweichende Lösungen gegenüber dem *HTFSPD-HF-Branch&Bound*-Verfahren. Die mit dem HTFSPD-HGA erzielten Verbesserungen fallen bei dieser im Vergleich zu der vorherigen Auswertung (vgl. Tabelle 7.2) zwar geringer aus, jedoch ist dies hier eher der zugrunde gelegten Bewertungsmethode geschuldet. Gleichwohl zeigen diese Rechenergebnisse, dass sich der HTFSPD-

Tabelle 7.3: Vergleich der Rechenergebnisse des HTFSPD-HGA mit denen des *HTFSPD-HF-Branch&Bound*-Verfahrens (mit $z^{max} = 3$)

| Daten- satz | f_{templ} | BB-HF-PD | | | HTFSPD-HGA | |
|----------------|-------------|-----------|-------|----------|---------------------|------------------|
| | | C_{max} | Gap | Zeit | $\emptyset C_{max}$ | \emptyset Zeit |
| <i>dat3</i> | 1000 | 543 | 0.00 | < 1.00 | 543.00 | 34.20 |
| | 600 | 417 | 0.00 | < 1.00 | 417.00 | 33.80 |
| <i>dat5</i> | 1000 | 846 | 0.00 | < 1.00 | 846.00 | 71.00 |
| | 600 | 836 | 0.00 | < 1.00 | 836.00 | 71.80 |
| <i>dat8</i> | 1000 | 502 | 0.00 | 14.43 | 502.00 | 224.80 |
| | 600 | 467 | 0.00 | 25.65 | 467.00 | 198.20 |
| <i>dat10</i> | 1000 | 641 | 0.00 | 5791.71 | 641.00 | 282.60 |
| | 600 | 574 | 0.00 | 3479.74 | 574.00 | 332.80 |
| <i>dat12</i> | 1000 | 872 | 51.02 | 3784.74* | 885.00 | 583.00 |
| | 600 | 674 | 0.00 | 11614.31 | 674.00 | 600.00 |
| <i>dat15</i> | 1000 | 771 | 35.67 | 18000 | 771.00 | 600.00 |
| | 600 | 768 | 48.59 | 9307.03* | 732.60 | 600.00 |
| <i>dat20</i> | 1000 | 920 | 52.83 | 2593.81* | 908.00 | 600.00 |
| | 600 | 842 | 42.99 | 6493.14* | 723.40 | 600.00 |
| <i>dat30</i> | 1000 | 1311 | 55.01 | 2988.67* | 1105.20 | 600.00 |
| | 600 | 1303 | 53.95 | 2379.23* | 1042.00 | 600.00 |

*Abbruch aufgrund des Speicherplatzbedarfs

HGA bezüglich der simultanen Optimierung von reiner Reihenfolgebildung und *Hollomon*-Fahrweise als durchaus geeignet erweist.

7.3 Der *Online*-Aspekt

Bis hierhin wurde die vorliegende Problemstellung ausschließlich im Sinne einer *Offline*-Optimierung betrachtet und dementsprechend modelliert. Bei dieser klassischen Betrachtungsweise wird davon ausgegangen, dass die relevanten Daten jeder Probleminstanz vollständig gegeben sind. Ein entsprechender *Offline*-Algorithmus berechnet dann auf Basis dieses vollständigen Wissens eine optimale bzw. approximative Lösung. In der Praxis liegt diese vollständige Information allerdings selten vor. Häufig müssen Entscheidungen ohne Wissen über zukünftige Ereignisse getroffen werden. Derartige Problemstellungen werden i. d. R. als *Online*-Probleme und die hierfür entwickelten Algorithmen

entsprechend als *Online*-Algorithmen bezeichnet. Näheres hierzu ist z. B. in Borodin und El-Yaniv [12] nachzulesen.

Auch die Problemstellung, der sich die Dillinger Hüttenwerke im Rahmen der Belegungsoptimierung des RHO3 gegenübersehen, erweist sich im Realbetrieb als *Online*-Problem. So stehen dem Ofensteuerer z. B. für die Reihenfolgeplanung nur die Waben zur Verfügung, die sich bereits im Lager befinden und deren nächster Fertigungsschritt durch eine Wärmebehandlung im RHO3 gekennzeichnet ist. Für die Erstellung des Produktionsprogramms wäre es jedoch auch interessant zu wissen, welche Waben wann zukünftig hinzukommen. In Abhängigkeit der Eigenschaften und Bereitstellungszeitpunkte dieser Waben hätten diese Informationen zum Teil großen Einfluss auf die Gestaltung des Produktionsprogramms. Zur Veranschaulichung dient das folgende Beispiel.

Angenommen das aktuelle Produktionsprogramm sei so festgelegt worden, wie in Abbildung 7.8 blau dargestellt. Dieses Programm entspricht der mit dem *HTFSPD-Branch&Bound*-Verfahren unter Verwendung von *Simulation-2* ermittelten optimalen Lösung für Datensatz *dat10* und $f_{templ} = 1000$. Weiterhin sei nun angenommen, dass zum Zeitpunkt der Erstellung des Produktionsprogramms (d. h. zum Zeitpunkt 0) die grün dargestellten Informationen voll-

| Wabe | Ofentemperatur [°C] | Ofenverweilzeit [min] | FAZO [min] | FEZO [min] |
|------|---------------------|-----------------------|------------|------------|
| 3 | 650 | 164 | 3 | 177 |
| 5 | 640 | 277 | 15 | 302 |
| 2 | 620 | 299 | 39 | 348 |
| 7 | 610 | 386 | 63 | 459 |
| 9 | 610 | 386 | 87 | 483 |
| 6 | 610 | 386 | 177 | 573 |
| 10 | 630 | 319 | 302 | 631 |
| 8 | 630 | 280 | 381 | 671 |
| 1 | 630 | 275 | 405 | 690 |
| 4 | 650 | 205 | 506 | 721 |

| Wabe | Ofentemperatur [°C] | Ofenverweilzeit [min] | Bereitstellungszeitpunkt [min] |
|------|---------------------|-----------------------|--------------------------------|
| 11 | 650 | 164 | 30 |
| 12 | 610 | 386 | 90 |
| 13 | 600 | 311 | 310 |

Abbildung 7.8: Die Belegungsoptimierung des RHO3 als *Online*-Problem

ständig vorliegen. Einfachheitshalber sind hier mit der Ofentemperatur und der Ofenverweilzeit nur beispielhaft die beiden wichtigsten Wabendaten aufgeführt. Unter dem Bereitstellungszeitpunkt ist der Zeitpunkt zu verstehen, zu dem die betreffende Wabe in den Lagerbereich eintritt und ab dann somit prinzipiell für die Einfahrt in den Ofen zur Verfügung steht.

Die Frage ist nun, ob und wie sich das optimale Produktionsprogramm in diesem Beispiel durch die zusätzlichen Informationen verändert. Die Antwort darauf liefert wieder das *HTFSPD-Branch&Bound*-Verfahren. Hierzu muss in der Ofensimulation lediglich gewährleistet werden, dass eine Wabe erst ab ihrem Bereitstellungszeitpunkt zur Verfügung steht. Das resultierende Produktionsprogramm ist aus Abbildung 7.9 zu entnehmen.

| Wabe | Ofentemperatur [°C] | Ofenverweilzeit [min] | FAZO [min] | FEZO [min] |
|------|---------------------|-----------------------|------------|------------|
| 3 | 650 | 164 | 3 | 177 |
| 11 | 650 | 164 | 33 | 207 |
| 4 | 650 | 205 | 43 | 258 |
| 5 | 640 | 277 | 58 | 345 |
| 8 | 630 | 280 | 73 | 363 |
| 10 | 630 | 319 | 86 | 417 |
| 1 | 630 | 275 | 142 | 427 |
| 2 | 620 | 299 | 171 | 480 |
| 6 | 610 | 386 | 258 | 654 |
| 7 | 610 | 386 | 345 | 741 |
| 13 | 600 | 311 | 462 | 783 |
| 12 | 610 | 386 | 472 | 868 |
| 9 | 610 | 386 | 482 | 878 |

Abbildung 7.9: Die Belegungsoptimierung bei vollständiger Information

Wie man sieht, ist Wabe 11 nun an zweiter Stelle eingeplant. Da sie erst zum Zeitpunkt 30 zur Verfügung steht, kann sie frühestens zum Zeitpunkt 33 (3 min Transportzeit) in den Ofen einfahren. Interessant hierbei ist, dass es sich scheinbar dennoch lohnt, auf das Eintreffen von Wabe 11 zu warten. Des Weiteren sind die Waben 12 und 13 im neuen Produktionsprogramm entsprechend auf die Positionen 12 und 11 verteilt. Bei einem Vergleich mit dem ursprünglichen Produktionsprogramm (vgl. Abbildung 7.8) fällt auf, dass die „neuen“ Waben nicht nur einfach an bestimmten Stellen eingefügt wurden, sondern sich auch das restliche Gesamtbild verändert hat. Die Gesamtfertigstellungszeit beträgt

bei dieser Produktionsfolge 876 min. Es sei nochmals erwähnt, dass es sich hierbei um die optimale Lösung für den Fall vollständiger Information bezüglich der Waben 11, 12 und 13 handelt (*Offline*-Wissen).

Dieses kleine Beispiel soll verdeutlichen, dass Informationen über zukünftige Ereignisse großen Einfluss auf die Erstellung des Produktionsprogramms ausüben können. Wie bereits angedeutet, sind im Realbetrieb der Dillinger Hüttenwerke derartige Informationen wie in dem vorgestellten Beispiel nicht verfügbar. Zukünftig wärmezubehandelnde Waben sind zwar gewissermaßen bekannt, jedoch können bezüglich des Bereitstellungszeitpunkts größtenteils nicht einmal taggenaue Angaben gemacht werden. Diese Tatsache ist zum Teil den zahlreichen, komplexen und anlageintensiven Produktionsprozessen geschuldet. Die vorliegenden Informationen über zukünftig wärmezubehandelnde Waben sind demnach zu ungenau, um sie im Rahmen der Belegungsoptimierung des RHO3 gezielt einzusetzen.

Aus den aufgeführten Gründen ist es nur möglich, auf bekannt werdende Bereitstellungszeitpunkte entsprechend zu reagieren, d. h. das Produktionsprogramm gegebenenfalls umzuplanen. Welche Folgen sich daraus ergeben, soll wieder anhand des oben eingeführten Beispiels verdeutlicht werden. Dabei wird zum einen vorausgesetzt, dass der Ofensteuerer direkt von den Bereitstellungszeitpunkten erfährt, sobald diese vorliegen. Zum anderen wird im Folgenden angenommen, dass die Reaktionen auf derartige Informationen immer unmittelbar erfolgen und daraus jeweils das für den aktuellen Kenntnisstand optimale Produktionsprogramm resultiert.

Zum Zeitpunkt 0 sind nur die zehn Waben aus Datensatz *dat10* bekannt. Das Produktionsprogramm entspricht folglich zunächst dem aus Abbildung 7.8 (blaue Tabelle). Ab dem Zeitpunkt 30 stehen die Informationen über Wabe 11 zur Verfügung. Die bei dieser Informationsmenge optimale Reaktion ist in Abbildung 7.10 veranschaulicht (wieder mit dem *HTFSPD-Branch&Bound*-Verfahren ermittelt). Man stellt fest, dass es zu diesem Planungszeitpunkt hinsichtlich der Minimierung der Gesamtfertigstellungsdauer am besten ist, Wabe 11 der letzten Reihenfolgeposition zuzuordnen. Bei einem Vergleich mit dem zum Zeitpunkt 0 festgelegten Produktionsprogramm fällt zudem auf, dass sich auch die Reihenfolge der anderen Waben wiederum geändert hat.

| Wabe | Ofentemperatur [°C] | Ofenverweilzeit [min] | FAZO [min] | FEZO [min] |
|------|---------------------|-----------------------|------------|------------|
| 3 | 650 | 164 | 3 | 177 |
| 5 | 640 | 277 | 15 | 302 |
| 10 | 630 | 319 | 42 | 371 |
| 6 | 610 | 386 | 60 | 456 |
| 9 | 610 | 386 | 78 | 474 |
| 7 | 610 | 386 | 177 | 573 |
| 2 | 620 | 299 | 302 | 611 |
| 1 | 630 | 275 | 413 | 698 |
| 8 | 630 | 280 | 434 | 724 |
| 4 | 650 | 205 | 519 | 734 |
| 11 | 650 | 164 | 573 | 747 |

Abbildung 7.10: Die Belegungsoptimierung zum Zeitpunkt 30

Die nächste Information im Sinne der Bereitstellung von Wabe 12 wird zum Zeitpunkt 90 bekannt. Mittlerweile sind die Waben 10, 6 und 9 in den Ofen eingefahren. Abbildung 7.11 zeigt das bei Berücksichtigung der neuen Informationen resultierende, optimale Produktionsprogramm. Es zeigt sich, dass zunächst, wie zuvor geplant, mit den Waben 7 und 2 fortgesetzt wird. Die darauffolgende Position nimmt nun jedoch Wabe 12 ein. Die Reihenfolge der restlichen vier Waben bleibt unverändert. Man kann bereits jetzt (auch wenn erst 12 Waben verplant sind) feststellen, dass diese Produktionsfolge mit einer Gesamtfertigungszeit von 883 min eine längere Zeit in Anspruch nimmt als

| Wabe | Ofentemperatur [°C] | Ofenverweilzeit [min] | FAZO [min] | FEZO [min] |
|------|---------------------|-----------------------|------------|------------|
| 3 | 650 | 164 | 3 | 177 |
| 5 | 640 | 277 | 15 | 302 |
| 10 | 630 | 319 | 42 | 371 |
| 6 | 610 | 386 | 60 | 456 |
| 9 | 610 | 386 | 78 | 474 |
| 7 | 610 | 386 | 177 | 573 |
| 2 | 620 | 299 | 302 | 611 |
| 12 | 610 | 386 | 371 | 767 |
| 1 | 630 | 275 | 492 | 777 |
| 8 | 630 | 280 | 573 | 863 |
| 4 | 650 | 205 | 658 | 873 |
| 11 | 650 | 164 | 709 | 883 |

Abbildung 7.11: Die Belegungsoptimierung zum Zeitpunkt 90

die mit vollkommener Information ermittelte Variante.

Schließlich fehlt noch Wabe 13, deren Bereitstellung jedoch erst zum Zeitpunkt 310 bekannt wird. Eine erneute Belegungsoptimierung führt zu dem in Abbildung 7.12 dargestellten Produktionsprogramm.

Die Gesamtfertigstellungsdauer liegt mit 1006 min ca. 15% über der Zeit, die benötigt gewesen wäre, wenn zum Zeitpunkt 0 vollkommene Information über die Waben 11, 12 und 13 geherrscht hätte.

Es bleibt festzuhalten, dass sich Informationen über zukünftig wärmezubehandelnde Waben bei gezielter Verwendung positiv auf den Ofendurchsatz auswirken können. Da derartige Informationen bei den Dillinger Hüttenwerken allerdings nicht bzw. zu ungenau vorhanden sind, bleibt nur die Möglichkeit durch entsprechende Umplanungen immer wieder auf Änderungen zu reagieren. In diesem Fall muss dann aber zumindest gewährleistet werden, dass dieses Umplanen unmittelbar nach Eingang der entsprechenden Information erfolgt. Bislang ist dies nur bedingt der Fall. Der Ofensteuermann kann zwar den aktuellen Lagerbestand manuell abrufen, jedoch erstellt er sein Produktionsprogramm i. d. R. zu Beginn seiner Schicht und ändert es währenddessen nur selten wieder ab.

| Wabe | Ofentemperatur [°C] | Ofenverweilzeit [min] | FAZO [min] | FEZO [min] |
|------|---------------------|-----------------------|------------|------------|
| 3 | 650 | 164 | 3 | 177 |
| 5 | 640 | 277 | 15 | 302 |
| 10 | 630 | 319 | 42 | 371 |
| 6 | 610 | 386 | 60 | 456 |
| 9 | 610 | 386 | 78 | 474 |
| 7 | 610 | 386 | 177 | 573 |
| 2 | 620 | 299 | 302 | 611 |
| 12 | 610 | 386 | 371 | 767 |
| 13 | 600 | 311 | 456 | 777 |
| 8 | 630 | 280 | 573 | 863 |
| 1 | 630 | 275 | 611 | 896 |
| 11 | 650 | 205 | 767 | 941 |
| 4 | 650 | 205 | 791 | 1006 |

Abbildung 7.12: Die Belegungsoptimierung zum Zeitpunkt 310

Aus den erläuterten Gründen ist es im Rahmen der Belegungsoptimierung sinnvoll, den verfügbaren Lagerbestand in regelmäßigen Abständen zu aktualisieren. Mit jeder Aktualisierung wird dann anschließend das in dieser Arbeit entwickelte gesamtheitliche Verfahren zur Ofenbelegungsoptimierung (vgl. Abbildung 7.1) aufgerufen, um eine eventuelle Neu- oder Umplanung des Produktionsprogramms vorzunehmen. Die Implementierung und Umsetzung im Realbetrieb der Dillinger Hüttenwerke wird in Kapitel 8 noch näher vorgestellt.

Der regelmäßige Aufruf der Belegungsoptimierung offenbart einige Schwachstellen der bisherigen Verfahrensweise. Der Grund hierfür ist, dass bislang von einer reinen *Offline*-Optimierung ausgegangen wurde. Auch wenn Informationen über zukünftige Ereignisse nicht gezielt zum Einsatz kommen, so hat der *Online*-Aspekt in Form der regelmäßigen Durchführung der Optimierung durchaus einen dynamischen Effekt, den es zu berücksichtigen gilt und der Anpassungen erfordert. So wurde bislang z. B. angenommen, dass der Ofen zu Beginn des Verfahrens immer leer ist. Es ist offensichtlich, dass dies nicht immer bzw. sogar nur äußerst selten der Fall sein kann. Es gilt jedoch nicht nur die aktuelle Ofenbelegung, sondern auch die Belegungen der Rollgänge am Zu- und Auslauf sowie der Brennanlagen zu ermitteln. All diese Startbelegungen bilden dann entsprechend fixe Ausgangspunkte für die jeweilige Optimierung.

Eine weitere Anpassung, welche durch Betrachtung des *Online*-Aspekts erforderlich wird, betrifft die eigentliche Zielfunktion. Betrachtet man wieder das für Datensatz *dat10* ermittelte, optimale Produktionsprogramm aus Abbildung 7.8, so könnte man sich über die Positionierung von Wabe 4 wundern. Diese belegt die letzte Stelle, erfordert aber die gleiche Ofentemperatur wie Wabe 3, die direkt zu Beginn eingeplant wurde. Natürlich existieren neben der Ofentemperatur noch weitere Einflussfaktoren. Dennoch werfen derartige Temperaturverläufe Fragen auf, zumal diese auch in anderen Testrechnungen immer wieder beobachtet wurden. Ein kleines Beispiel soll im Folgenden Antworten liefern.

Abbildung 7.13 zeigt die Ofenbelegung des optimalen Produktionsprogramms für Datensatz *dat10*, wie sie zum Zeitpunkt 550 aussehen würde. Mit Wabe 4 ist bereits die letzte Wabe eingefahren. Ihre Ausfahrt zum Zeitpunkt 721 stellt gleichzeitig die minimale Gesamtfertigstellungsdauer des Datensatzes und da-

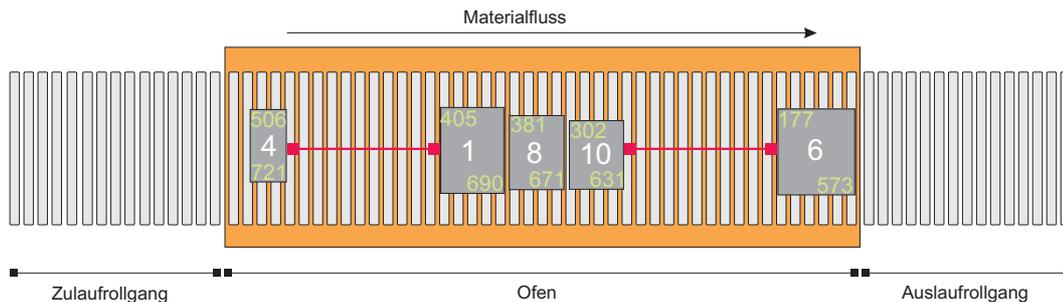


Abbildung 7.13: Die Ofenbelegung zum Zeitpunkt 550 des aus statischer Sicht optimalen Produktionsprogramms

mit die optimale Lösung dar. Dies gilt jedoch nur für die aktuell betrachteten Waben. Berücksichtigt man jedoch zukünftige Ereignisse bzw. im Lager eintreffende Waben (auch wenn man sie nicht kennt), so muss dieses Produktionsprogramm hinterfragt werden. Der Vergleich mit einem leicht abgeänderten Produktionsprogramm soll dies verdeutlichen. Im Folgenden sei Wabe 4 nun an der zweiten, statt an der letzten Position eingeplant. Abbildung 7.14 stellt die daraus resultierende Ofenbelegung dar, wie sie sich wiederum zum Zeitpunkt 550 ergeben würde.

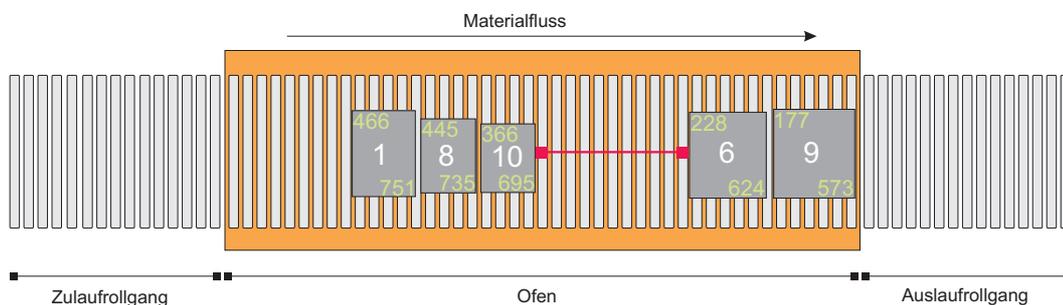


Abbildung 7.14: Die Ofenbelegung zum Zeitpunkt 550 des aus dynamischer Sicht vorgeschlagenen Produktionsprogramms

Man stellt fest, dass die Gesamtfertigstellungsdauer nun mit 751 min um 30 min höher ist als die der optimalen Lösung für *dat10*. Dennoch findet sich ein wesentliches Argument, welches für die Wahl dieses abgeänderten Produktionsprogramms spricht und sich auf die Restkapazität des Ofens bezieht. Diese liegt bei dem aus statischer Sicht optimalen Produktionsprogramm nach der Einfahrt der letzten Wabe bei 3733 mm. Bei Wahl des zweiten Produktionsprogramms hätte man dagegen 15266 mm an freier Restkapazität. Betrachtet man nur die 10 Waben des Datensatzes, so ist diese Information hinsichtlich

des Ofendurchsatzes prinzipiell irrelevant. Weicht man von dieser statischen Betrachtung allerdings ab und bezieht den dynamischen Aspekt mit ein, so ändert sich der Blickwinkel auf die im Anschluss an das aktuelle Produktionsprogramm zur Verfügung stehende Kapazität im Ofen. Denn, je mehr Restkapazität, desto eher können anschließend wärmezubehandelnde Waben in den Ofen einfahren. Bei einer Ofenbelegung, wie in Abbildung 7.14 dargestellt, ist dieses Potential wesentlich höher als bei der Ofenbelegung aus Abbildung 7.13.

Das Beispiel hat gezeigt, dass die rein statische Zielsetzung der Minimierung der Gesamtfertigstellungsdauer mit Hinsicht auf die im Zeitverlauf regelmäßige Durchführung der Belegungsoptimierung aufgehoben bzw. modifiziert werden muss. Die Frage ist nun, wie die abschließend zur Verfügung stehende Restkapazität im Ofen sinnvoll in die Bewertung integriert werden kann. Die diesbezüglich vorgeschlagene Idee besteht darin, einfach einige fiktive Waben am Ende des Produktionsprogramms hinzuzufügen. Hier bietet es sich an, bezüglich der Längen viele sehr kleine Waben zu nehmen, um die Restkapazität möglichst gut verteilen zu können.

Im Folgenden seien den aus den beiden Abbildungen 7.13 und 7.14 zugrundeliegenden Produktionsprogrammen jeweils 20 fiktive Waben hinzugefügt, deren Längen 2000 mm und deren Ofenverweilzeiten 277 min betragen, wobei letzteres dem Durchschnittswert anzulassender Waben entspricht. Nun werden diese erweiterten Produktionsfolgen erneut hinsichtlich ihrer Gesamtfertigstellungsdauer bewertet. Das Ergebnis stellt sich wie folgt dar. Das aus statischer Sicht optimale Produktionsprogramm benötigt nun 1316 min, das aus dynamischer Sicht vorgeschlagene Produktionsprogramm dagegen nur 1196 min. Bei dieser Bewertungsmethode ist das zweite Produktionsprogramm daher dem ersten vorzuziehen. Die Erweiterung der ursprünglichen Bewertungsmethode um die fiktiven Waben scheint somit für die Berücksichtigung des dynamischen Effekts geeignet zu sein. An dieser Stelle sei darauf hingewiesen, dass bei der Simulation der fiktiven Waben bewusst nur aufgrund nicht ausreichend freier Ofenkapazität oder unterschiedlicher Ofenverweilzeiten entstehende Wartezeiten berücksichtigt wurden. Temperaturlücken wurden für die fiktiven Waben vernachlässigt (es wurde lediglich mit dem Mindestabstand $temp^{min}$ gerechnet).

Um zu prüfen, ob das Abbildung 7.14 zugrundeliegende Produktionsprogramm auch der aus dynamischer Sicht optimalen Produktionsfolge entspricht, wird das *HTFSPD-Branch&Bound*-Verfahren erneut mit der erweiterten Bewertung durchgeführt. Dabei stellt sich heraus, dass die optimale Produktionsfolge mit $folge = (3, 9, 7, 6, 2, 10, 1, 8, 5, 4)$ eine andere ist. Die optimale Gesamtfertigungsdauer einschließlich der fiktiven Waben beträgt 1067 min. Die letzte nicht fiktive Wabe fährt zwar erst nach 768 min aus dem Ofen aus, jedoch sind hier im Anschluss an ihre Einfahrt noch 28868 mm Restkapazität im Ofen zur Verfügung.

Es bleibt zunächst festzuhalten, dass die aufgeführten Beispiele gezeigt haben, dass Informationen über zukünftige Ereignisse (insbesondere über im Lager eintreffende, wärmezubehandelnde Waben) die Erstellung des Produktionsprogramms positiv beeinflussen können. Da diese jedoch i. d. R. zu ungenau vorliegen, um sie explizit in die Planung mit einzubeziehen, bietet sich eine regelmäßige Aktualisierung der Belegungsoptimierung an. Somit wird gewährleistet, dass neu im Lager eintreffende, wärmezubehandelnde Waben schnellstmöglich erkannt und in die Planungen aufgenommen werden. Zudem erscheint es sinnvoll, die ursprünglich statische Bewertung durch Hinzufügen fiktiver Waben um eine eher dynamische Sichtweise zu erweitern. Diesbezüglich sei jedoch noch darauf hingewiesen, dass die Eigenschaften der fiktiven Waben nicht zwingend wie vorgestellt gewählt werden müssen. Wenn beispielsweise geplant ist, im Anschluss an das aktuelle Produktionsprogramm von einer Anlass- auf eine Vorwärmbehandlung zu wechseln, so macht es Sinn die fiktiven Waben mit Vorwärmern entsprechenden Eigenschaften zu versehen. Bei einem solchen Verfahrenswechsel ist es dann auch denkbar, Temperaturlücken mit einzubeziehen.

Die bisherigen Ausführungen bezüglich des *Online*-Aspekts betrafen hauptsächlich die Reihenfolgeoptimierung. Wie in Abbildung 7.1 gekennzeichnet, hat der *Online*-Aspekt zudem Auswirkungen auf die WBH-Blockbildung. Die beiden in Kapitel 7.1 vorgeschlagenen Regeln bezüglich der Bearbeitungsreihenfolge der WBH-Blöcke beruhen nämlich ebenfalls nur auf einer statischen Sichtweise. Aus dynamischer Sicht stellt sich jedoch die Frage, wie bzw. wo im Zeitverlauf neu hinzukommende Waben integriert werden sollen. Hierfür müssen die Regeln der WBH-Blockbildung erweitert werden. Die diesbezüglichen

Überlegungen sollen wiederum anhand eines kleinen Beispiels demonstriert werden.

Angenommen, die aktuelle WBH-Blockbildung sei wie in Abbildung 7.15 dargestellt geplant. Die Darstellung beschränkt sich auf den Verweis auf das jeweils erforderliche WBH-Verfahren und die entsprechende Sollproduktionswoche, wobei diese als Differenz zur aktuellen Produktionswoche angegeben ist. Es handelt sich hierbei um eine Konstellation, welche sich bei Anwendung der statischen Regeln ergeben würde. Die acht anzulassenden (A), die acht vorzuwärmenden (VW) und die vier Waben, die eine Kantenentspannung (KE) benötigen, werden jeweils in Gruppen zusammengefasst. Da die insgesamt kleinste Sollproduktionswoche unter den anzulassenden Waben zu finden ist, wird dieser Block an erster Stelle eingeplant. Die kleinsten Sollproduktionswochen des VW- und des KE-Blocks sind zwar identisch, jedoch sind die vorzuwärmenden Waben im Durchschnitt eiliger. Somit folgt auf den A-Block zunächst der VW- und anschließend der KE-Block.

| Wabe | WBH-Verfahren | Sollproduktionswoche |
|------|---------------|----------------------|
| 10 | A | +1 |
| 5 | A | -1 |
| 1 | A | -1 |
| 6 | A | -5 |
| 3 | A | +1 |
| 4 | A | -5 |
| 2 | A | -2 |
| 11 | A | -1 |
| 7 | VW | -3 |
| 8 | VW | -3 |
| 12 | VW | -3 |
| 16 | VW | -1 |
| 14 | VW | -1 |
| 13 | VW | -1 |
| 9 | VW | -1 |
| 17 | VW | -1 |
| 15 | KE | 0 |
| 18 | KE | 0 |
| 19 | KE | -3 |
| 20 | KE | -2 |

Abbildung 7.15: Der *Online*-Aspekt im Zusammenhang mit der WBH-Blockbildung

Aus statischer Sicht mag diese Blockbildung sinnvoll erscheinen. Im Zeitverlauf birgt sie jedoch einige Tücken. Angenommen, die Behandlung der Anlasser beginnt mit den Waben 4 und 6. Sobald diese allerdings fertiggestellt sind, würde eine darauffolgende WBH-Blockbildung den VW-Block vor den A-Block einplanen. Den gleichen Effekt hätte die Ankunft eines Vorwärmers, der entsprechend eiliger wäre als die Waben 4 und 6. Dabei stellt sich die Frage, ob es wirklich sinnvoll ist, die Vorwärmer immer direkt vorzuziehen, insbesondere wenn man bedenkt, dass wenige Zeit später vielleicht wieder eine noch eiligere Anlasswabe eintreffen könnte. Eine solche Planungsweise hätte womöglich häufige Wechsel zwischen den WBH-Verfahren zur Folge, welche ja ursprünglich durch die WBH-Blockbildung eigentlich verhindert werden sollten.

Eine Strategie, um häufige WBH-Block-Wechsel zu vermeiden, besteht darin, den ersten WBH-Block einfach solange zu fixieren bis die letzte Wabe dieses Blocks in den Ofen eingefahren ist. Die Gefahr hierbei ist, dass daraus nur sehr selten oder sogar nie WBH-Block-Wechsel resultieren würden. Letzteres würde dann eintreten, wenn der Lagerbestand des aktuellen WBH-Blocks trotz dessen Bearbeitung nie auf null sinkt. Um dies wiederum zu verhindern, wird die WBH-Blockbildung um folgende Regel ergänzt:

- Entferne aus dem ersten WBH-Block alle Waben, deren Sollproduktionswochen über dem auf die nächste Ganzzahl gerundeten Durchschnitt des darauffolgenden WBH-Blocks liegen.

Diese Strategie führt bei dem in Abbildung 7.15 dargestellten Beispiel dazu, dass die Waben 10 und 3 nachträglich aus dem A-Block entfernt und zunächst zurückgestellt werden. Dies gilt analog für im Zeitverlauf neu hinzukommende Waben, die prinzipiell dem ersten WBH-Block zugehörig wären. Wenn also, bezogen auf das Beispiel, eine neue Anlasswabe eintrifft, so würde diese zurückgestellt werden, sofern ihre Sollproduktionswoche (bzw. die Differenz zur aktuellen Produktionswoche) größer -1 beträgt. Andernfalls würde die Wabe dem aktuellen A-Block zugeordnet werden.

Abschließend sei noch darauf hingewiesen, dass dem Betrieb, insbesondere den Ofensteuermännern, aber auch den Kranfahrern oder Bedienern der Brennanlagen, ein wenig Zeit eingeräumt werden sollte, um sich auf einen WBH-Block-Wechsel vorzubereiten. Daher ist es sinnvoll, auch den zweiten WBH-Block

frühzeitig zu fixieren. Als geeigneter Zeitpunkt wurde hierfür die Ofeneinfahrt der vorletzten Wabe des aktuellen WBH-Blocks festgelegt. Ab diesem Zeitpunkt dürfen diesem WBH-Block dann auch keine neu hinzukommenden Waben mehr zugeordnet werden.

7.4 Temperaturlückenoptimierung

Bislang konzentrierten sich die Optimierungsgedanken hauptsächlich auf die Gestaltung der Reihenfolge, in der die Waben den Ofen passieren. Hinsichtlich der Temperaturlücken wurde sich lediglich auf Aussagen und Verhalten der Ofensteuermänner gestützt. Konkret wurden jeweils zwei Szenarien betrachtet ($f_{templ} = 1000$ und $f_{templ} = 600$), wobei die bisherigen Rechenergebnisse belegen, dass die Temperaturlückenvorgabe den Ofendurchsatz zum Teil erheblich beeinflusst. Aus diesem Grund lohnt es sich, diese Vorgaben einer genaueren Untersuchung zu unterziehen.

Zunächst fällt auf, dass bezüglich der Temperaturlückenvorgabe wohl eine gewisse Unkenntnis bzw. Unsicherheit vorliegt. Dies belegt bereits die Diskrepanz zwischen den beiden angewandten Regeln. Nun könnte man einwenden, dass die Vorgabe $f_{templ} = 1000$ grundsätzlich verworfen werden kann, da die mit $f_{templ} = 600$ gebildeten Lücken scheinbar auch ausreichen. Dies ist allerdings nicht immer der Fall, wie auch die nachfolgenden Untersuchungen zeigen. In diesem Zusammenhang muss allerdings zunächst geklärt werden, wann eine Temperaturlücke überhaupt zu klein ist, bzw. welche Bedingungen erfüllt sein müssen, damit sie als ausreichend groß gilt.

Der grundsätzliche Sinn und Zweck einer Temperaturlücke besteht darin, den gegenseitigen Einfluss, den zwei im Ofen benachbarte Waben mit unterschiedlicher Wärmebehandlungstemperatur aufeinander ausüben, zu minimieren. Ist dies der Fall, so ist jede Wabe genau der Temperatur ausgesetzt, die für ihre Wärmebehandlung vorgesehen ist. Aus Sicht des Qualitätsmanagements wäre es somit am besten, überhaupt keine Waben mit unterschiedlichen Temperaturen gleichzeitig im Ofen zu behandeln. Darunter würden allerdings logistische Zielsetzungen, wie Durchsatz und Liefertreue, stark leiden. Eine Regelung zur

Temperaturlückenvorgabe sollte somit immer einen Kompromiss aus beiden Aspekten darstellen. Grundsätzlich gilt eine Temperaturlücke zwischen zwei Waben unterschiedlicher Wärmebehandlungstemperatur als zu klein, wenn eine Wabe zu viel oder zu wenig Wärme erhält. In gewissen Grenzen stellt beides separat gesehen eigentlich gar kein Problem dar. Eine Wabe, die mehr oder weniger Wärme als vorgesehen erhält, ist einfach früher oder später fertig als ursprünglich geplant. Die folgende Abbildung soll die eigentliche Problematik dahinter verdeutlichen.

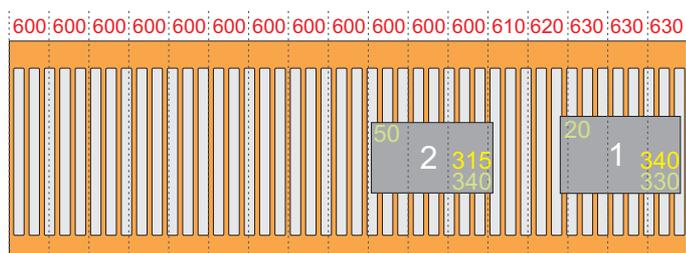


Abbildung 7.16: Auswirkung einer zu kleinen Temperaturlücke

Abbildung 7.16 zeigt eine Ofensituation in der zwei Waben unterschiedlicher Wärmebehandlungstemperatur aufeinander folgen. Wabe 1 benötigt eine Ofentemperatur von $630\text{ }^{\circ}\text{C}$, wohingegen Wabe 2 nur $600\text{ }^{\circ}\text{C}$ erfordert. Zwischen den Waben wird eine ca. 8 m lange Lücke gelassen. In dem dargestellten Beispiel entspricht dies einer zu kleinen Lücke, wie man anhand der geplanten (grün gekennzeichneten) und tatsächlichen (gelb gekennzeichneten) Ofenzeiten beobachten kann. Ursprünglich geplant war, dass Wabe 1 zum Zeitpunkt 330 und Wabe 2 zum Zeitpunkt 340 aus dem Ofen ausfährt. Aufgrund der zu kleinen Lücke wird Wabe 1 nun jedoch erst 10 min später fertig. Dagegen müsste Wabe 2 nun eigentlich bereits zum Zeitpunkt 315 aus dem Ofen ausfahren. Da Überholvorgänge im Ofen nicht möglich sind, muss Wabe 2 zwangsläufig mindestens 25 min warten. Die Folge könnte im schlimmsten Fall ein verfehlter Wärmebehandlungseffekt sein.

Im Rahmen dieser Arbeit wurden zwei Ansätze zur Verbesserung der Temperaturlückenvorgabe entwickelt. Sie werden in den folgenden Unterkapiteln vorgestellt.

7.4.1 Eine neue Ofenzonenregelung

Der erste Ansatz betrifft die reine Regelung der Ofenzonen. Das Ziel besteht darin, den gegenseitigen Einfluss zwischen zwei Waben unterschiedlicher Wärmebehandlungstemperatur, zunächst unabhängig von der eigentlichen Länge der Temperaturlücke, durch eine neue Ofenzonenregelung zu verringern.

Bislang wurden die Ofenzonen bei einem Temperatursprung, wie bereits in Kapitel 5.1.2 anhand von Abbildung 5.5 erläutert, geregelt. Demnach wurde entsprechend der Lückenlänge ein linear ansteigender bzw. absteigender Temperaturverlauf angestrebt. Die Idee dahinter bestand darin, den Temperaturübergang möglichst gleichmäßig zu gestalten. Im Rahmen von im Zusammenhang mit der vorliegenden Arbeit durchgeführten Beobachtungen haben sich jedoch zwei Schwachpunkte dieser Regelungsart herauskristallisiert.

Es hat sich gezeigt, dass vergleichbare Temperatursprünge (mit entsprechend gleich großen Lücken) bei manchen Ofenbelegungskonstellationen zu sehr unterschiedlichen Wärmeübertragungen führen. Eine Ursache hierfür kann die Position der Waben im Ofen sein. Eine ungünstige Positionierung lässt sich auch in Abbildung 5.5 erkennen. Hier ragt der Fuß von Wabe 1 (der Randbereich zum Ofeneingang hin) noch um ca. einen Meter in die 14. Ofenzone. Bei dieser Zone wird nun bereits begonnen herunterzuregeln, da Wabe 2 eine geringere Ofentemperatur erfordert und Wabe 1 weniger als die Hälfte der Zone abdeckt. Genau im letzten Punkt liegt ein Schwachpunkt der alten Regelungsweise. Eine mögliche Folge dieser Situation könnte sein, dass Wabe 1 langsamer erwärmt und somit länger als geplant im Ofen bleiben muss. Dies könnte sich wiederum auf Wabe 2 auswirken und zu einer Situation, wie im Zusammenhang mit Abbildung 7.16 geschildert, führen. Selbst wenn dieser Fall nicht eintritt, so wird die dargestellte Ofenregelung dennoch dem Ziel einer möglichst homogenen Wabenerwärmung bei Wabe 1 nicht gerecht.

Aufgrund der erläuterten Problematik besteht die erste Änderung der Ofenzonenregelung bei Temperatursprüngen darin, alle Zonen, die auch nur zum Teil durch eine Wabe ausgefüllt werden, auf die der Wabe entsprechende Temperatur zu regeln. Bezüglich der in Abbildung 5.5 dargestellten Situation würde dies bedeuten, dass auch die 14. Ofenzone auf 620 °C geregelt wird. Dies hät-

te wiederum zur Folge, dass sich die Regelung der Ofenzonen 11, 12 und 13 entsprechend auf 605 °C, 610 °C und 615 °C anpasst.

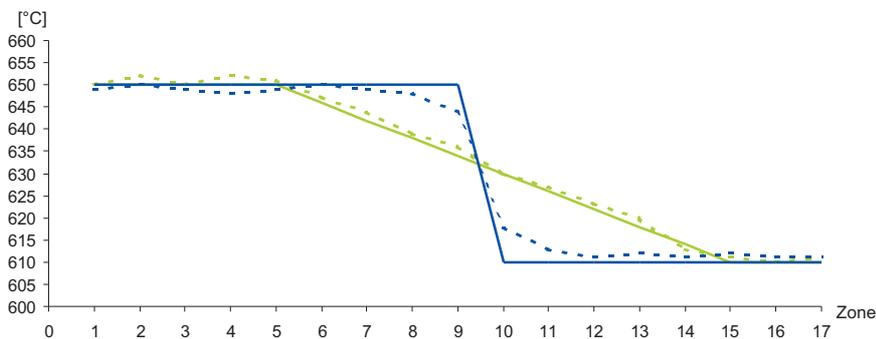
Der zweite Schwachpunkt der ursprünglichen Regelungsart betrifft die bei Temperatursprüngen gezielt linear eingestellten Temperaturregelungsverläufe. Hier stellt sich die Frage, ob es grundsätzlich sinnvoll ist, den Temperaturübergang innerhalb einer Temperaturlücke möglichst gleichmäßig zu gestalten. Umgekehrt könnte man den Übergang auch mit einem großen Sprung in der Mitte der Temperaturlücke einstellen. Diese Idee stellt die zweite Änderung der ursprünglichen Ofenzonenregelung dar. Bezogen auf Abbildung 5.5 würden die Zonen 11 und 12 nun auf 600 °C und die Zonen 13 und 14 auf 620 °C geregelt werden. Es ist utopisch anzunehmen, dass die realen Ofentemperaturen, insbesondere in den Zonen 12 und 13, dann genau den Regelungstemperaturen entsprechen werden. Der Sinn dieses Ansatzes liegt viel eher darin, sich den Wärmeübergang auf natürliche Weise einstellen zu lassen, anstatt ihm „künstlich“ einen linearen Verlauf aufzuzwingen.

Beobachtungen der realen Ofenzonentemperaturen belegen, dass der gegenseitige Einfluss zweier Waben unterschiedlicher Wärmebehandlungstemperatur bei dem sprunghaften Temperaturwechsel in der Mitte der Temperaturlücke geringer ausfällt als bei der linearen Zonenregelung. Abbildung 7.17 stellt zwei beispielhafte Situationen gegenüber.

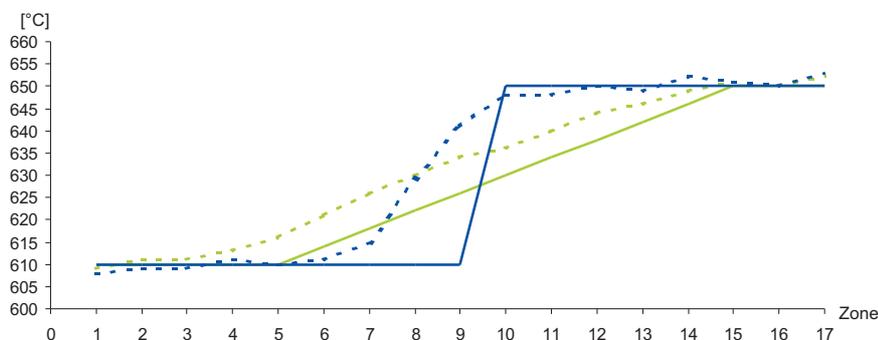
In beiden Graphiken repräsentieren die durchgezogenen Linien die jeweiligen Zonenregelungstemperaturen (grün für die alte und blau für die neue Regelungsmethode). Zudem sind typische, bei der jeweiligen Regelung zu beobachtende Realtemperaturwerte durch die in entsprechender Farbe gekennzeichneten, gestrichelten Linien abgebildet. Die beiden Teilabbildungen 7.17(a) und 7.17(b) unterscheiden sich lediglich darin, ob der Temperatursprung von kalt (am Ofenende) auf heiß (am Ofenanfang) oder von heiß auf kalt erfolgt. Bei den Beispielen liegt jeweils ein Temperatursprung um 40 °C (von 610 °C auf 650 °C und umgekehrt) bei einer Temperaturlücke von ca. 9 Zonen (ca. 40 m) zugrunde.

Abbildung 7.17(a) zeigt, dass Real- und Regelungswerte bei der linearen Regelung nahezu identisch sind, was an der ausreichend großen Temperaturlücke (kleine Steigung) liegt. Bei der sprunghaften Regelung ist das insbesondere

Abbildung 7.17: Vergleich des neuen mit dem ursprünglichen Temperaturregelungsansatzes für zwei beispielhafte Situationen



(a) Temperatursprung kalt (am Ofenende) auf heiß (am Ofenanfang)



(b) Temperatursprung heiß (am Ofenende) auf kalt (am Ofenanfang)

in den Zonenbereichen um den Mittelpunkt der Temperaturlücke weniger der Fall. Dies ist für den gegenseitigen Wärmeeinfluss allerdings nur indirekt von Bedeutung. Dieser ist vielmehr anhand der realen Temperaturen zu messen und damit bei der sprunghaften Regelung deutlich geringer. Hier vollzieht sich der Wärmeübergang lediglich in den Zonen 8 - 11 und damit schneller als bei der linearen Regelung, welche die komplette Temperaturlücke (Zonen 6 - 14) in Anspruch nimmt. Mit der sprunghaften Regelung hätte somit auch eine 3 - 4 Zonen lange Lücke ausgereicht. Eine derart kleinere Lücke hätte für die lineare Regelung eine größere Steigung zur Folge. In diesem Fall wäre mit wesentlich größeren Abweichungen zwischen den Real- und Regelungswerten und damit einem starken gegenseitigen Wärmeeinfluss zu rechnen.

Abbildung 7.17(b) schildert den umgekehrten Fall, bei dem der Temperatursprung also von heiß nach kalt erfolgt. Bei diesem Szenario kommt ein zusätzlicher Aspekt zum Tragen, der darin besteht, dass die Abwärme des Ofens

immer in Richtung Ofeneingang geleitet wird. Für einen Temperatursprung von heiß nach kalt resultiert daraus ein deutlich verzögertes Abkühlverhalten im Ofen. Dieser Effekt ist in Abbildung 7.17(b) direkt zu beobachten. Gerade bei diesen Temperatursprüngen wird der Nachteil der linearen Regelung besonders deutlich, da hier die Real- und Regelungstemperaturen auch bei großen Temperaturlücken divergieren. Die sprunghafte Regelung ist von diesem Effekt zwar auch betroffen, jedoch ist auch hier ein schnellerer Wärmeübergang und ein geringerer gegenseitiger Wärmeeinfluss zu verzeichnen.

7.4.2 Ein Ansatz zur Temperaturlückenvorgabe aus der Wärmelehre

Während der erste Ansatz zur Temperaturlückenoptimierung eher auf rein regelungstechnischen Aspekten beruht und nur indirekt abhängig von den tatsächlichen Temperaturlückenzuständen ist, folgt nun ein Ansatz zur effektiveren Bestimmung bzw. Vorgabe der Lückenzustände. Hierzu ist ein kurzer Ausflug in die Wärmelehre erforderlich. Die folgenden Ausführungen sind größtenteils aus dem VDI-Wärmeatlas [93] entnommen.

Bei Temperaturen über dem Gefrierpunkt absorbieren und emittieren materielle Körper Energie durch Wärmestrahlung. Als direkte Konsequenz findet eine Energieübertragung zwischen verschiedenen temperierten Oberflächen durch einen Wärmestrom statt. Dieser stellt eine richtungsabhängige Größe dar. Folglich ist neben der reinen Temperaturdifferenz auch die Lage der beteiligten Oberflächen zueinander von Bedeutung. Dabei findet ein Wärmestrom immer von Oberflächen höherer Temperaturen zu Oberflächen mit niedrigeren Temperaturen statt. Außerdem können Flächen nur Wärme durch Strahlung auf Flächen übertragen, die sie „sehen“. Dabei ist „sehen“ als ungehinderte Erreichbarkeit jedes Punktes der einen Fläche durch jeden Punkt der anderen Fläche mit Hilfe einer Geraden zu verstehen.

Als Bezugsgröße für die meisten wärmestrahlungstheoretischen Ansätze dient die Annahme des schwarzen Körpers bzw. Strahlers. Dabei handelt es sich um einen idealisierten, diffusen Strahler, der auftreffende Strahlung vollständig absorbiert und maximal in alle Richtungen abstrahlt. Stehen sich zwei gleich

große, schwarze Oberflächen A_1 und A_2 ($A_1 = A_2$) mit unterschiedlichen Temperaturen TA_1 und TA_2 ($TA_1 < TA_2$) parallel gegenüber, so berechnet sich der Wärmestrom \dot{Q} nach Gleichung 7.5:

$$\dot{Q} = \sigma A_1 (TA_1^4 - TA_2^4). \quad (7.5)$$

Es ist zu beachten, dass die Temperaturen (auch in den folgenden Abschnitten) stets in Kelvin anzugeben sind. Der Proportionalitätsfaktor σ stellt die sogenannte Stefan-Boltzmann-Konstante dar, welche sich wie folgt berechnet:

$$\sigma = 5,67 \cdot 10^{-8} \frac{W}{(m^2 \cdot K)}. \quad (7.6)$$

Für die meisten praktischen Anwendungen ist Gleichung 7.5 eher ungeeignet, da die beteiligten Flächen i. d. R. unterschiedliche Größen aufweisen und sich häufig auch nicht parallel gegenüberstehen. In diesen Fällen trifft nur ein bestimmter Anteil der insgesamt emittierten Wärmestrahlung einer Fläche auf eine ihr benachbarte Fläche. Entscheidend sind in diesem Zusammenhang insbesondere die Abmessungen der betrachteten Flächen sowie deren Lage zueinander. Zur Berechnung des Anteils der absorbierten bzw. emittierten Wärmestrahlung zwischen zwei Oberflächen dient die dimensionslose Größe φ , auch Einstrahlzahl oder Sichtfaktor genannt. Grundsätzlich kann diese wie folgt berechnet werden (mit β_1 und β_2 als Winkel zwischen den Flächennormalen und der Verbindungsgerade zwischen den Flächenmittelpunkten s):

$$\varphi_{12} = \frac{1}{\pi A_1} \int_{A_1} \int_{A_2} \frac{\cos \beta_1 \cos \beta_2}{s^2} dA_1 dA_2. \quad (7.7)$$

Für verschiedene geometrische Konfigurationen existieren bereits Lösungen dieser Differentialgleichung (vgl. VDI [93]). So z. B. auch für den Fall, zweier paralleler Rechteckflächen, bei dem die eine Fläche mit einer Ecke in der Mittenkrechten der anderen Fläche liegt. Abbildung 7.18 stellt eine derartige Konstellation schematisch dar.

Für den abgebildeten Fall lassen sich die Einstrahlzahlen mit Hilfe von Gleichung 7.8 ermitteln:

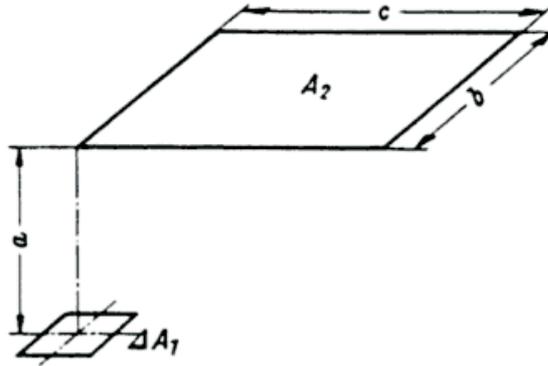


Abbildung 7.18: Zwei parallele Rechteckflächen, wobei eine mit einer Ecke in der Mittelsenkrechten der anderen Fläche liegt (VDI [93])

$$\varphi_{12} = \frac{1}{2\pi} \left(\frac{\left(\frac{b}{a}\right)}{\sqrt{1 + \left(\frac{b}{a}\right)^2}} \arctan \frac{\left(\frac{c}{a}\right)}{\sqrt{1 + \left(\frac{b}{a}\right)^2}} + \frac{\left(\frac{c}{a}\right)}{\sqrt{1 + \left(\frac{c}{a}\right)^2}} \arctan \frac{\left(\frac{b}{a}\right)}{\sqrt{1 + \left(\frac{c}{a}\right)^2}} \right). \quad (7.8)$$

Durch die explizite Betrachtung der Einstrahlzahlen erweitert sich Gleichung 7.5 schließlich zu Gleichung 7.9:

$$\dot{Q} = \sigma A_1 \varphi_{12} (TA_1^4 - TA_2^4). \quad (7.9)$$

Der in dieser Arbeit entwickelte Ansatz zur Temperaturlückenvorgabe basiert auf der Idee, den Wärmestrom, den eine Wabe j aufgrund eines Temperatursprungs zu ihrer Vorgängerwabe j' zusätzlich zu dem für sie prinzipiell erforderlichen Wärmestrom erhält, mit diesem in Verhältnis zu setzen. Eine Temperaturlücke ist dann so zu wählen, dass die beiden folgenden Bedingungen erfüllt sind:

$$\frac{\dot{Q}_{j,WZ_j}^K}{\dot{Q}_{j,WZ_{j'}}^K} \leq \frac{\dot{Q}_{krit}^K}{d_j}, \quad (7.10)$$

$$\frac{\dot{Q}_{j,WZ_j}^K}{\dot{Q}_{j,WZ_{j'}}^K} - \frac{\dot{Q}_{j,WZ_j}^F}{\dot{Q}_{j,WZ_{j'}}^F} \leq \frac{\dot{Q}_{krit}^{KF}}{d_j}, \quad (7.11)$$

wobei $\dot{Q}_{j,WZ_{j'}}^K$ den Wärmestrom bezeichnet, der von der Wärmezone von Wabe j' auf den Kopf von Wabe j strömt (mit j' als direkten Nachbar von Wabe

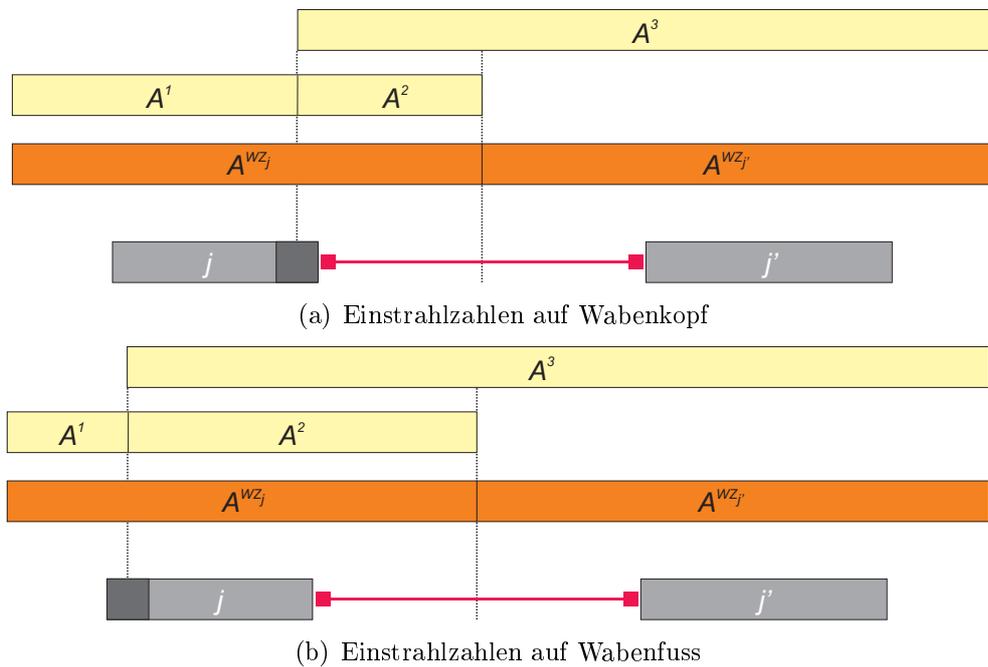
j). Analog kennzeichnet $\dot{Q}_{j,WZ_{j'}}^F$ den entsprechenden Wärmestrom, der auf den Fuß von Wabe j wirkt. An dieser Stelle sei darauf hingewiesen, dass eine Wärmezone (WZ) als Verbund von Ofenzonen zu verstehen ist. Zudem stellt Wabe j in den folgenden Ausführungen immer die Wabe dar, deren Wärmebehandlungstemperatur unter der von Wabe j' liegt. Des Weiteren ist zu beachten, dass die Wärmeströme hier lediglich für einen stationären Zustand berechnet werden. In diesem Zusammenhang wird TA_j stets mit $wbht_j^{Ziel}$ initialisiert.

Die Toleranzparameter \dot{Q}_{krit}^K und \dot{Q}_{krit}^{KF} bilden jeweils dividiert durch die Dicke von Wabe j (in mm) die entsprechend für den Wabekopf sowie für die Kopf-Fußabweichung einzuhaltenden Wärmestromverhältnisse. Sie wurden mit $\dot{Q}_{krit}^K = 3,3$ und $\dot{Q}_{krit}^{KF} = 2,5$ implementiert. Die Toleranzwerte werden bewusst mit der Wabendicke in Abhängigkeit gebracht, um unterschiedliche Vorgaben für verschieden dicke Waben zu generieren. Dies ist sinnvoll, da z. B. dünnere Waben schneller aufheizen als dickere.

Entscheidend für die Berechnung der Wärmeströme ist die Ermittlung der Einstrahlzahlen. Diesbezüglich kann auf die in Abbildung 7.18 dargestellte Konstellation und demnach auf Gleichung 7.8 zurückgegriffen werden. Eine ähnliche Konstellation lässt sich nämlich auch im Ofen feststellen, wenn man Fläche A_1 als eine bestimmte Wabe und Fläche A_2 als eine bestimmte Wärmezone betrachtet. Die Strecke a stellt in diesem Fall den Abstand zwischen der Oberfläche der betrachteten Wabe und der Ofendecke dar. Dieser wird für eine Wabe j im Folgenden mit d_j^{Ofen} bezeichnet. Die Strecken b und c entsprechen folglich der Breite bzw. Länge der betrachteten Wärmezone. Die Längen und Breiten einer Fläche A werden im weiteren Verlauf dieser Arbeit allgemein mit l^A und b^A bezeichnet.

Die Vorgehensweise zur Berechnung der Einstrahlzahlen im Zusammenhang mit der in dieser Arbeit vorliegenden Problemstellung lässt sich mit Hilfe von Abbildung 7.19 beschreiben. Teilabbildung 7.19(a) zeigt zwei Waben j und j' , die um die Länge einer bestimmten Temperaturlücke auseinander liegen. Über ihnen sind mit A^{WZ_j} und $A^{WZ_{j'}}$ ihre zugehörigen Wärmezonen gekennzeichnet. Wie man sieht, ist hier bereits die in Kapitel 7.4.1 erläuterte, sprunghafte Regelung vorausgesetzt. Zur Vereinfachung werden die Ofenzonen im Folgenden allerdings vernachlässigt, d. h. es wird angenommen, dass die Tempera-

Abbildung 7.19: Berechnung der auf Wabenkopf und -fuß bezogenen Einstrahlzahlen bei einem Temperatursprung



turregelung genau in der Mitte der Temperaturlücke umspringt. Zudem sei dies auch gleichzeitig der Mittelpunkt des Ofens, sodass $A^{WZ_j} = A^{WZ_{j'}}$ und $l^{WZ_j} + l^{WZ_{j'}} = l^{Ofen}$ gelte.

Um Gleichung 7.8 anwenden zu können, müssen die Wärmezonen in Teilzonen zerlegt bzw. zusammengefasst werden. So zerlegt die Mittelsenkrechte des Kopfes von Wabe j (die Länge dieses Flächenelements sei auf 1 m gesetzt) Fläche A^{WZ_j} in A^1 und A^2 . Letztere wird mit der Fläche $A^{WZ_{j'}}$ zusammengefasst und bildet somit die Fläche A^3 . Diese Vorgehensweise ist erforderlich, da Gleichung 7.8 nur für in Abbildung 7.18 dargestellte, geometrische Konstellationen gültig ist. Eine derartige Konstellation liegt nun für die Flächen A^1 , A^2 und A^3 in Bezug auf den Kopf von Wabe j vor. Demnach lässt sich die Einstrahlzahl $\varphi_{j,WZ_{j'}}^K$ zwischen dem Kopf von Wabe j und Wärmezone j' mit Hilfe des für Einstrahlzahlen geltenden Zerlegungsgesetzes (siehe VDI [93], Abschnitt Ka 7) wie folgt ermitteln:

$$\varphi_{j,WZ_{j'}}^K = 2(\varphi_{j,A^3}^K - \varphi_{j,A^2}^K). \quad (7.12)$$

Für die Berechnung des Wärmestroms, den Wabe j von ihrer eigenen Wärme-

zone erhält, gilt dementsprechend:

$$\varphi_{j,WZ_j}^K = 2(\varphi_{j,A^1}^K + \varphi_{j,A^2}^K). \quad (7.13)$$

Bei beiden Berechnungen ist zu beachten, dass die Breiten der fiktiven Flächen A^1 , A^2 und A^3 aufgrund der in Abbildung 7.18 dargestellten, geometrischen Konstellation lediglich $\frac{b^{Ofen}}{2}$, also der Hälfte der Ofenbreite, betragen dürfen. Aus diesem Grund ist hier die nachträgliche Multiplikation mit 2 erforderlich, um die komplette Ofenbreite zu berücksichtigen.

Die soeben erläuterte Vorgehensweise zur Berechnung der Einstrahlzahl bezüglich des Kopfes von Wabe j lässt sich analog auf die für den Fuß von Wabe j geltende Einstrahlzahl $\varphi_{j,WZ_{j'}}^F$ übertragen. Es gelten die folgenden Gleichungen:

$$\varphi_{j,WZ_{j'}}^F = 2(\varphi_{j,A^3}^F - \varphi_{j,A^2}^F). \quad (7.14)$$

$$\varphi_{j,WZ_j}^F = 2(\varphi_{j,A^1}^F + \varphi_{j,A^2}^F). \quad (7.15)$$

Wie Teilabbildung 7.19(b) zeigt, ergeben sich bei der Betrachtung des Wabenfußes lediglich andere Flächenlängenverhältnisse.

Die Wärmestrahlung ist jedoch nicht alleiniger Verursacher für den Transport innerer Energie im Ofen. Wie bereits erläutert und in Abbildung 7.17(b) gut zu erkennen, spielt auch der Abtransport der Abwärme eine Rolle. Diese als (erzwungene) Konvektion bezeichnete Form der Wärmeübertragung ist jedoch nur schwer analytisch berechenbar. Im Folgenden wird durch die Einführung von Konvektionsparametern versucht, den konvektiven Effekt im Rahmen des vorgestellten Wärmestrahlungsansatzes abzubilden bzw. zu berücksichtigen.

Die Idee besteht darin, den Temperaturwechsel von der Mitte einer Temperaturlücke je nach Situation einfach ein Stück in Richtung des Ofeneingangs zu verschieben. Je nach Situation bedeutet, dass hierbei Richtung und Höhe des Temperatursprungs ausschlaggebend sind. Bezüglich der Temperaturdifferenz wird eine lineare Beziehung angenommen. Die Verschiebung $templ^v$ wird wie folgt berechnet:

$$templ^v = \begin{cases} kvp^{hk}(wbht_{j'}^{Ofen} - wbht_j^{Ofen}), & \text{wenn } fpos_{j'} = fpos_j - 1, \\ kvp^{kh}(wbht_{j'}^{Ofen} - wbht_j^{Ofen}), & \text{wenn } fpos_{j'} = fpos_j + 1. \end{cases} \quad (7.16)$$

Dabei stellt kvp^{hk} den Konvektionsparameter für einen Temperatursprung von heiß nach kalt (anzuwenden, wenn Wabe j' direkter Vorgänger von Wabe j ist) und kvp^{kh} den entsprechenden Parameter für einen Wechsel von kalt nach heiß (anzuwenden, wenn Wabe j' direkter Nachfolger von Wabe j ist) dar. Die Parameter wurden experimentell ermittelt und schließlich mit $kvp^{hk} = 150$ und $kvp^{kh} = 5 \frac{\text{mm}}{^\circ\text{C}}$ implementiert. Bei der in Abbildung 7.17(b) dargestellten Situation würde der Temperatursprung somit künstlich um sechs Meter ($150 \cdot (650 - 610)$ mm) in Richtung des Ofeneingangs verschoben werden.

Algorithmus 11 fasst den entwickelten Ansatz zur Temperaturlückenvorgabe in Form eines Pseudocodes zusammen.

Algorithmus 11 Temperaturlückenberechnung

Require: $wbht_j^{Ofen}$, $wbht_j^{Ziel}$, $wbht_{j'}^{Ofen}$, $fpos_j$, $fpos_{j'}$, l_j , b_j , d_j , d_j^{Ofen}
mit $j, j' \in J$, l^{Ofen} , b^{Ofen} , kvp^{hk} , kvp^{kh} , \dot{Q}_{krit}^K , \dot{Q}_{krit}^{KF}

Ensure: $templ_j$

- 1: berechne $templ^v$ gemäß Gleichung 7.16
- 2: $A_j = l_j \cdot b_j$
- 3: $a = d_j^{Ofen}$
- 4: $b = \frac{b^{Ofen}}{2}$
- 5: $l^{WZ_{j'}} = \frac{l^{Ofen}}{2} + templ^v$
- 6: $l^{WZ_j} = l^{Ofen} - l^{WZ_{j'}}$
- 7: **for** $templ_j = 500, 1000, 1500, \dots, 50000$ **do**
- 8: $l^{A^2} = 500 + \frac{templ_j}{2} - templ^v$
- 9: $l^{A^3} = l^{WZ_{j'}} + l^{A^2}$
- 10: $c = l^{A^3}$
- 11: berechne φ_{j,A^3}^K gemäß Gleichung 7.8
- 12: $c = l^{A^2}$
- 13: berechne φ_{j,A^2}^K gemäß Gleichung 7.8
- 14: $\varphi_{j,WZ_{j'}}^K = 2(\varphi_{j,A^3}^K - \varphi_{j,A^2}^K)$
- 15: $\dot{Q}_{j,WZ_{j'}}^K = \sigma A_j \varphi_{j,WZ_{j'}}^K (wbht_{j'}^{Ofen^4} - wbht_j^{Ziel^4})$
- 16: $l^{A^1} = l^{WZ_j} - l^{A^2}$
- 17: $c = l^{A^1}$
- 18: berechne φ_{j,A^1}^K gemäß Gleichung 7.8
- 19: $\varphi_{j,WZ_j}^K = 2(\varphi_{j,A^1}^K + \varphi_{j,A^2}^K)$
- 20: $\dot{Q}_{j,WZ_j}^K = \sigma A_j \varphi_{j,WZ_j}^K (wbht_j^{Ofen^4} - wbht_j^{Ziel^4})$
- 21: $l^{A^2} = l_j - 500 + \frac{templ_j}{2} - templ^v$
- 22: $l^{A^3} = l^{WZ_{j'}} + l^{A^2}$
- 23: $c = l^{A^3}$

```

24:   berechne  $\varphi_{j,A^3}^F$  gemäß Gleichung 7.8
25:    $c = l^{A^2}$ 
26:   berechne  $\varphi_{j,A^2}^F$  gemäß Gleichung 7.8
27:    $\varphi_{j,WZ_{j'}}^F = 2(\varphi_{j,A^3}^F - \varphi_{j,A^2}^F)$ 
28:    $\dot{Q}_{j,WZ_{j'}}^F = \sigma A_j \varphi_{j,WZ_{j'}}^F (wbht_{j'}^{Ofen^4} - wbht_j^{Ziel^4})$ 
29:    $l^{A^1} = l^{WZ_j} - l^{A^2}$ 
30:    $c = l^{A^1}$ 
31:   berechne  $\varphi_{j,A^1}^F$  gemäß Gleichung 7.8
32:    $\varphi_{j,WZ_j}^F = 2(\varphi_{j,A^1}^F + \varphi_{j,A^2}^F)$ 
33:    $\dot{Q}_{j,WZ_j}^F = \sigma A_j \varphi_{j,WZ_j}^F (wbht_j^{Ofen^4} - wbht_j^{Ziel^4})$ 
34:   if  $\frac{\dot{Q}_{j,WZ_j}^K}{\dot{Q}_{j,WZ_{j'}}^K} \leq \frac{\dot{Q}_{krit}^K}{d_j}$  and  $\frac{\dot{Q}_{j,WZ_j}^K}{\dot{Q}_{j,WZ_{j'}}^K} - \frac{\dot{Q}_{j,WZ_j}^F}{\dot{Q}_{j,WZ_{j'}}^F} \leq \frac{\dot{Q}_{krit}^{KF}}{d_j}$  then
35:       break
36:   end if
37: end for

```

Im ersten Schritt wird geschaut, in welcher Reihenfolgebeziehung die Waben j und j' zueinander stehen und dementsprechend $templ^v$ ermittelt (Zeile 1). In den Zeilen 2 - 6 werden dann die Berechnungen durchgeführt, die von der Länge der Temperaturlücke unabhängig sind. In der anschließenden *for*-Schleife wird die Temperaturlücke $templ_j$ sukzessive in 500 mm Schritten bis zu einer maximalen Lücke von 50000 mm erhöht. In jedem Durchlauf werden zunächst die Längen der Flächen A^2 und A^3 angepasst und deren Einstrahlzahlen bezüglich des Kopfes von Wabe j berechnet (Zeilen 8 - 13). Die Länge des Wabenkopfes wird übrigens (ebenso wie die des Wabenfußes) grundsätzlich mit 1000 mm veranschlagt. Aus φ_{j,A^2}^K und φ_{j,A^3}^K lässt sich nun $\varphi_{j,WZ_{j'}}^K$ ableiten (Zeile 14), womit sich schließlich der Wärmestrom $\dot{Q}_{j,WZ_{j'}}^K$ ermitteln lässt (Zeile 15). Es folgt in den Zeilen 16 - 20 eine analoge Vorgehensweise zur Berechnung des von Wärmzone j ausgehenden Wärmestroms \dot{Q}_{j,WZ_j}^K . Im darauffolgenden Abschnitt (bis Zeile 33) werden die gleichen, bis hierhin auf den Wabenkopf bezogenen Berechnungen nun für den Wabenfuß wiederholt. Zum Schluss wird auf die Einhaltung der vorgegebenen Bedingungen geprüft (Zeile 34). Sobald beide Bedingungen erfüllt sind, bricht die *for*-Schleife ab und die aktuelle Länge von $templ_j$ stellt die vorgeschlagene Temperaturlücke dar.

Es ist zu beachten, dass die mit Algorithmus 11 ermittelten Temperaturlücken zum Teil stark von denen mit der ursprünglichen Berechnungsmethode (mit $f_{templ} = 600$ bzw. $f_{templ} = 1000$) ermittelten Lücken abweichen. Allgemein

sind die neuen Lücken in den meisten Fällen wesentlich kürzer. Es gibt jedoch auch Situationen, in denen größere Lücken als zuvor vorgeschlagen werden. Diese Unterschiede können zum einen dadurch begründet werden, dass der neue Ansatz mehr Faktoren berücksichtigt als der ursprüngliche, bei dem prinzipiell nur die Höhe des Temperatursprungs betrachtet wird. In den neuen Ansatz gehen z.B. zusätzlich die Dicke und die Länge der jeweils relevanten Wabe sowie die Richtung des Temperaturwechsels mit ein. Zum anderen geht die Berücksichtigung der Höhe des Temperatursprungs hier mit den auf Basis der Einstrahlzahlen berechneten Wärmeströmen wesentlich tiefer.

Anhand der Abbildungen 7.20(a) - 7.20(d) lassen sich die Auswirkungen der unterschiedlichen Einflussfaktoren ablesen. Die Auswertungen basieren grundsätzlich auf folgenden Daten bzw. Annahmen:

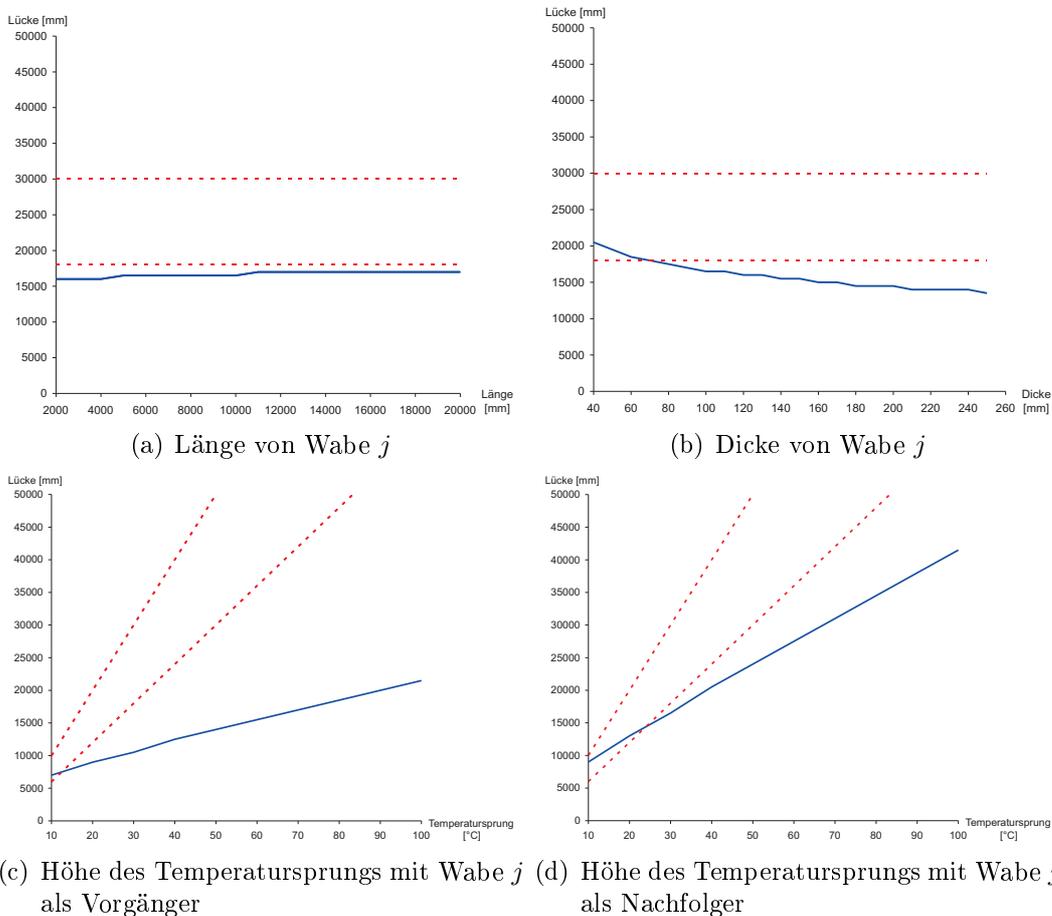
- $l_j = 10000$,
- $d_j = 100$,
- $wbht_j^{Ofen} = 630$,
- $wbht_j^{Ziel} = 610$,
- $wbht_j^{Ofen} = 660$.

Auf Basis dieser Daten wurde nun eine Ceteris-Paribus-Analyse durchgeführt. In Teilabbildung 7.20(a) wurde die Wabengänge von 2000 - 20000 mm variiert. Die blaue Linie repräsentiert die resultierenden Temperaturlückenvorgaben bei Anwendung des neuen Ansatzes. Die alten Vorgaben mit $f_{templ} = 600$ und $f_{templ} = 1000$ sind mit den rot gestrichelten Linien gekennzeichnet. Wie man sieht, liegen die neuen leicht unter den mit $f_{templ} = 600$ berechneten Lücken. Allgemein ist mit steigender Wabengänge ein leichter Anstieg der Lücken zu beobachten.

Auf die gleiche Weise wie mit der Wabengänge, wurde in Teilabbildung 7.20(b) nun die Wabendicke von 40 - 240 mm variiert. Hier sieht man, dass die neuen Lücken bei sehr dünnen Waben (bis ca. 80 mm) noch leicht über den ursprünglich mit $f_{templ} = 600$ berechneten Lücken liegen. Mit steigender Dicke sinken die neuen Lücken jedoch weiter ab.

Den größten Einfluss hat jedoch nach wie vor die Höhe des Temperatursprungs,

Abbildung 7.20: Auswirkungen unterschiedlicher Einflussfaktoren auf die Temperaturlückenvorgabe



wobei nun die Richtung des Temperaturwechsels zusätzlichen Einfluss ausübt. Teilabbildung 7.20(c) zeigt die unterschiedlichen Verläufe der Temperaturlücken, wenn die Ofentemperaturdifferenzen sukzessive von 10 - 100 °C erhöht werden (d. h. $wbht_{j'}^{Ofen}$ von 640 - 730 °C variiert) unter der Annahme, dass Wabe j nun den Vorgänger bildet (d. h. ein Wechsel von kalt nach heiß erfolgt). Aufgrund des geringeren Konvektionsfaktors sind diese Lücken wesentlich kleiner als die nach der alten Methode berechneten Lücken. Führt man nun die gleiche Analyse mit Wabe j als Nachfolger durch, so ergeben sich die in Teilabbildung 7.20(d) dargestellten Lücken. Diese liegen für Temperatursprünge unter 20 °C noch über den ursprünglich mit $f_{templ} = 600$ berechneten Lücken, steigen bei höheren Temperaturdifferenzen allerdings wesentlich geringer an.

Es ist festzuhalten, dass die Vorteile der neuen Lückenvorgabe darin bestehen,

dass nun wesentlich mehr Einflussfaktoren berücksichtigt werden können und die Lücken zudem größtenteils wesentlich kürzer sind als zuvor, was sich positiv auf den Ofendurchsatz auswirkt. In diesem Zusammenhang gilt es hervorzuheben, dass die neue Temperaturlückenberechnung bereits im Realbetrieb des RHO3 eingesetzt wird und bislang keine Qualitätsbeeinträchtigungen oder negative Rückmeldungen von Seiten des Betriebs zu verzeichnen sind.

Zum Abschluss sei darauf hingewiesen, dass in Verbindung mit der neuen Temperaturlückenberechnung zusätzlich Anpassungen an den Ofenverweilzeiten vorgenommen werden. Dies ist sinnvoll, da bei einem Temperatursprung, trotz Lücke, immer noch eine (wenn auch geringe) Wärmeübertragung stattfindet. Folglich wird bei einem Temperatursprung immer die Ofenverweilzeit der Wabe angepasst, welche die geringere Wärmebehandlungstemperatur benötigt (Wabe j). Dabei erfolgt die Anpassung, wie schon die Ermittlung der Temperaturlücke (vgl. Ungleichungen 7.10 und 7.11), auf Basis der Wärmestromverhältnisse. Hierzu werden die in Algorithmus 11 berechneten Werte für Kopf und Fuß gemittelt, sodass im ersten Schritt eine angepasste Ofentemperatur für Wabe j entsprechend Gleichung 7.17 berechnet wird:

$$wbht_j^{Ofen*} = \left(1 + 0,5 \frac{(\dot{Q}_{j,WZ_{j'}}^K - \dot{Q}_{j,WZ_{j'}}^F)}{d_j} \right) wbht_j^{Ofen}. \quad (7.17)$$

Im zweiten Schritt wird mit dieser angepassten Ofentemperatur die neue Erwärmdauer ewd_j^* berechnet. Die neue Ofenverweilzeit ovz_j^* ergibt sich schließlich aus der Summe aus ewd_j^* und hd_j (vgl. Gleichung 5.1).

7.5 *Simulation-3*

Sowohl der in Unterkapitel 7.3 vorgestellte *Online*-Aspekt als auch der in Unterkapitel 7.4 eingeführte Ansatz zur Ermittlung geeigneter Temperaturlückenvorschläge lassen sich durch Erweiterungen an *Simulation-2* realisieren. Die resultierende und letztendlich derzeit im Realbetrieb des RHO3 verwendete Materialflusssimulation wird als *Simulation-3* bezeichnet und in Algorithmus 12 zusammengefasst.

Algorithmus 12 *Simulation-3*

Require: Wabenmenge J , $wbht_j^{Ofen}$, ovz_j , l_j , akd_j^K , akd_j^O , brd_j mit $j \in J$, Pseudowabenmenge J' , $wbht_{j'}^{Ofen}$, $ovz_{j'}$, $l_{j'}$ mit $j' \in J'$, Stapelmenge S , $folge$, f_{templ} , $templ^{min}$, t^{min} , $AnzBR$

Ensure: $FAZO_j$, $FEZO_j$, $FEZO_j^{ARG}$, $FEZB_j$ mit $j \in J$, C_{max}

```

1:  $folge \leftarrow folge \cup \{J'\}$ 
2: for  $p = 1, \dots, n^*$  do
3:   if  $p \leq n$  then
4:      $t^{trans} = t^u$ 
5:     while  $folge_p \neq \bar{s}^{EZiel}$  do
6:       call Umstapelalgorithmus
7:       aktualisiere Stapelmenge  $S$ 
8:        $t^{trans} \pm t^u$ 
9:     end while
10:  end if
11:  if  $p = 1$  then
12:     $templ_{folge_p} = 0$ 
13:     $FAZO_{folge_p} = t^{trans}$ 
14:  else
15:    if  $wbh_{folge_p} = VW$  and  $wbh_{folge_{p-1}} = VW$  then
16:       $tmpd = 10$ 
17:    else if  $wbh_{folge_p} = VW$  and  $wbh_{folge_{p-1}} \neq VW$  then
18:       $tmpd = \max\{|630 - wbht_{folge_{p-1}}^{Ofen}|, |580 - wbht_{folge_{p-1}}^{Ofen}|\}$ 
19:    else if  $wbh_{folge_p} \neq VW$  and  $wbh_{folge_{p-1}} = VW$  then
20:       $tmpd = \max\{|wbht_{folge_p}^{Ofen} - 630|, |wbht_{folge_p}^{Ofen} - 580|\}$ 
21:    else
22:       $tmpd = |wbht_{folge_p}^{Ofen} - wbht_{folge_{p-1}}^{Ofen}|$ 
23:    end if
24:    call Temperaturlückenberechnung
25:    ermittle  $wbht_{folge_p}^{Ofen*}$  bzw.  $wbht_{folge_{p-1}}^{Ofen*}$  (Gleichung 7.17)
26:    passe  $ovz_{folge_p}^*$  bzw.  $ovz_{folge_{p-1}}^*$  an
27:     $FAZO_{folge_p} = FAZO_{folge_{p-1}} + \max\{t^{min}, t^{trans}\}$ 
28:  end if
29:   $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
30:  if  $|W^{Ofen}| > 0$  then
31:    if  $FEZO_{W_1^{Ofen}} + t^{min} \geq FEZO_{folge_p}$  then
32:       $FAZO_{folge_p} \pm FEZO_{W_1^{Ofen}} + t^{min} - FEZO_{folge_p}$ 
33:       $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
34:    end if
35:    while  $FEZO_{W_{|W^{Ofen}|}^{Ofen}} \leq FAZO_{folge_p}$  do

```

```

36:         if  $|W^{Ofen}| \leq 1$  then
37:              $l_{rest}^{Ofen} = l^{Ofen}$ 
38:         else
39:              $l_{rest}^{Ofen} \pm l_{W_{|W^{Ofen}|}}^{Ofen} + templ_{W_{|W^{Ofen}|-1}}^{Ofen}$ 
40:         end if
41:          $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}^{Ofen}\}$ 
42:     end while
43:     while  $l_{rest}^{Ofen} \leq l_{folge_p} + templ_{folge_p}$  do
44:         if  $|W^{Ofen}| = 0$  then
45:              $l_{rest}^{Ofen} = l^{Ofen}$ 
46:         end if
47:         if  $FEZO_{W_{|W^{Ofen}|}}^{Ofen} > FAZO_{folge_p}$  then
48:              $wl_{folge_p} = FEZO_{W_{|W^{Ofen}|}}^{Ofen} - FAZO_{folge_p}$ 
49:         end if
50:         if  $|W^{Ofen}| = 1$  then
51:              $l_{rest}^{Ofen} = l^{Ofen}$ 
52:         else
53:              $l_{rest}^{Ofen} \pm l_{W_{|W^{Ofen}|}}^{Ofen} + templ_{W_{|W^{Ofen}|-1}}^{Ofen}$ 
54:         end if
55:          $W^{Ofen} \leftarrow W^{Ofen} \setminus \{W_{|W^{Ofen}|}^{Ofen}\}$ 
56:     end while
57:      $FAZO_{folge_p} \pm wl_{folge_p}$ 
58:      $FEZO_{folge_p} = FAZO_{folge_p} + t^{min} + ovz_{folge_p}$ 
59: end if
60: if  $p \leq n$  then
61:      $l_{rest}^{ARG} = l^{ARG}$ 
62:      $W^{ARG} = ()$ 
63:     for  $q = 1, \dots, p - 1$  do
64:         if  $FEZO_{folge_q}^{ARG} > FEZO_{folge_p} + t^{min}$  then
65:              $W^{ARG} \leftarrow W^{ARG} \cup \{folge_q\}$ 
66:              $l_{rest}^{ARG} \pm l_{folge_q} + templ^{min}$ 
67:         end if
68:     end for
69:     while  $l_{rest}^{ARG} < l_{folge_p}$  do
70:          $FEZO_{folge_p} = FEZO_{W_{|W^{ARG}|}}^{ARG}$ 
71:          $FAZO_{folge_p} = FEZO_{folge_p} - t^{min} - ovz_{folge_p}$ 
72:          $l_{rest}^{ARG} \pm l_{W_{|W^{ARG}|}}^{ARG} + templ^{min}$ 
73:          $W^{ARG} \leftarrow W^{ARG} \setminus \{W_{|W^{ARG}|}^{ARG}\}$ 
74:     end while
75:      $FEZO_{folge_p}^{ARG} = FEZO_{folge_p} + t^{min} + akd_{folge_p}^K(wbht_{folge_p}^{Ziel}, 500)$ 
76:     if  $brd_{folge_p} > 0$  then

```

```

77:       $W^{BR} = ()$ 
78:      for  $q = 1, \dots, p - 1$  do
79:          if  $FEZB_{folge_q} > FEZO_{folge_p}^{ARG} + t^{min}$  then
80:               $W^{BR} \leftarrow W^{BR} \cup \{folge_q\}$ 
81:          end if
82:      end for
83:      if  $|W^{BR}| = AnzBR$  then
84:           $j^* = \operatorname{argmin}_{j \in W^{BR}} \{FEZB_j\}$ 
85:           $FEZO_{folge_p}^{ARG} = FEZB_{j^*}$ 
86:           $FEZO_{folge_p} = FEZO_{folge_p}^{ARG} - t^{min} - akd_{folge_p}^K (wbht_{folge_p}^{Ziel}, 500)$ 
87:           $FAZO_{folge_p} = FEZO_{folge_p} - t^{min} - ovz_{folge_p}$ 
88:      end if
89:       $FEZB_{folge_p} = FEZO_{folge_p}^{ARG} + t^{min} + brd_{folge_p}$ 
90:  end if
91:  end if
92:  if  $|W^{Ofen}| = 0$  then
93:       $l_{rest}^{Ofen} \equiv l_{folge_p} - templ_{folge_p}$ 
94:  else
95:       $l_{rest}^{Ofen} \equiv l_{folge_p}$ 
96:  end if
97:   $W^{Ofen} \leftarrow W^{Ofen} \cup \{folge_p\}$ 
98: end for
99:  $C_{max} = FEZO_{folge_{n^*}}$ 

```

Die Erweiterungen gegenüber *Simulation-2* betreffen im Prinzip nur wenige Zeilen. Zunächst wird die Pseudowabenmenge J' eingeführt, welche die fiktiven Waben beinhaltet, die in Verbindung mit der Berücksichtigung des *Online*-Aspekts benötigt werden. In Zeile 1 werden diese der Reihenfolge *folge* hinzugefügt bzw. hinten angehängt. Die *for*-Schleife aus Zeile 2 läuft demzufolge nun nicht mehr nur bis n , sondern bis n^* , wobei $n^* = n + |J'|$ gilt. Die letzten den *Online*-Aspekt betreffenden Erweiterungen sind in den Zeilen 3 und 60 zu finden, wobei beide *if*-Bedingungen lediglich bewirken, dass die entsprechend folgenden Schritte nur bei den Waben $folge_1, \dots, folge_n$ durchgeführt werden. Schließlich verbergen sich die Erweiterungen bezüglich des neu entwickelten Ansatzes zur Temperaturlückenvorgabe in den Zeilen 24 - 26. Sie beinhalten den Aufruf der in Algorithmus 11 beschriebenen Lückenberechnung, sowie die darauffolgende Ofenverweilzeit Anpassung.

Mit Abschluss dieses Kapitels ist nun das in dieser Arbeit entwickelte, gesamtheitliche Lösungskonzept vorgestellt (vgl. nochmals Abbildung 7.1). Das nächste Kapitel geht im Folgenden im Detail auf die konkrete Implementierung ein und liefert eine Gesamtbewertung.

Kapitel 8

Implementierung und Bewertung

In diesem Kapitel wird zunächst die Implementierung und Umsetzung des in dieser Arbeit entwickelten Lösungskonzepts in den Realbetrieb der Dillinger Hüttenwerke vorgestellt. Anfangs widmet sich Unterkapitel 8.1 noch der Beschreibung des Alt-Systems, ehe anschließend auf die Neuentwicklungen und den damit verbundenen neuen Planungsablauf eingegangen wird. Mit Unterkapitel 8.2 folgt zum Abschluss eine quantitative und qualitative Gesamtbewertung.

8.1 Realisierung des entwickelten Konzepts und Einbindung in den Realbetrieb

Es sei darauf hingewiesen, dass sich die Einbindung des in dieser Arbeit entwickelten Konzepts in den Realbetrieb der Dillinger Hüttenwerke nahezu parallel mit dessen Entwicklung vollzog. So wurden bereits die ersten Entwicklungen, wie beispielsweise die Bewertung durch *Simulation-1* (vgl. Algorithmus 2), direkt implementiert und folglich frühzeitig einer Praxistauglichkeitsprüfung unterzogen. Auf die Beschreibung der einzelnen Implementierungsphasen wird im Folgenden allerdings verzichtet, um den konzeptionellen Vergleich zwischen Alt- und Neu-System deutlicher hervorzuheben.

8.1.1 Das Alt-System

Das Alt-System ist in einer in VB6 programmierten Anwendung integriert. Nach dem Start des Programms erscheint die in Abbildung 8.1 abgebildete Oberfläche.



Abbildung 8.1: Startoberfläche des Alt-Systems

Von hier aus gelangt der Anwender über entsprechende Buttons zur Ansicht bzw. Erstellung des Produktionsprogramms für einen der drei existierenden Rollenherdöfen. Die folgenden Ausführungen beschränken sich auf den Erstellungsmodus des Produktionsprogramms für den RHO3. Nach dessen Aufruf öffnet sich das in Abbildung 8.2 (bzw. im Anhang Abbildung A.2) beispielhaft dargestellte Fenster.

Dem Bediener und damit im Normalfall dem Steuermann des RHO3 stehen nun vier verschiedene tabellarisch aufgebaute Ansichten zur Verfügung. Das Feld links oben verweist auf den Bestand an Waben, für die eine Wärmebehandlung am RHO3 vorgesehen ist. Dieser Gesamtbestand ist prinzipiell spaltenweise nach der Sollproduktionswoche und zeilenweise nach dem aktuellen Fertigungsabschnitt aufgeschlüsselt, wobei hier prioritäre Waben noch einmal gesondert gelistet sind. Für den Ofensteuermann sind hauptsächlich die in den grünen Zeilen dargestellten Bestände relevant, da dort alle Waben aufgeführt sind, deren nächster offener Fertigungsabschnitt eine Wärmebehandlung am RHO3 darstellt. Des Weiteren ist es dem Bediener möglich, sich nur die Bestände eines bestimmten Wärmebehandlungsverfahrens auszugeben.

Das Feld rechts oben stellt eine weitere Bestandsansicht dar. Hier werden die Waben entsprechend ihrer Zugehörigkeit zu bestimmten Lagerbereichen gruppiert. Aufgrund der geringen Relevanz dieser Ansichtsform wird auf diese ansonsten nicht weiter eingegangen.

| Alle | <= 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | >= 22 | Gesamt |
|-----------|-------|----|----|----|----|----|----|----|-------|--------|
| sonst. Or | 11 | 3 | 6 | 27 | 24 | 8 | 18 | 3 | 1 | 101 |
| Prio Pos | 1 | | | | | | | | | 1 |
| am RH03 | 2 | 1 | | 16 | 17 | 3 | 2 | 1 | | 42 |
| Prio Pos | | | | | | | | | | |
| Gesamt | 14 | 4 | 6 | 43 | 41 | 11 | 20 | 4 | 1 | 144 |

| Lage | Stoss | D min | D max | T min | T max | Produvo | Anz | Ton |
|--------------|---------|-------|-------|-------|-------|---------|-----|-----|
| BR 12.9.4 | 3039617 | 52 | 120 | 305 | 540 | 213-218 | 10 | 193 |
| 525 | 3032047 | 70 | 132 | 540 | 540 | 213-221 | 5 | 47 |
| UTS 12.4 (1) | 3038680 | 100 | 100 | 540 | 540 | 219-219 | 1 | 10 |
| BR 12.9.3 | 3034224 | 100 | 120 | 540 | 540 | 218-219 | 4 | 60 |
| BR 12.10.3 | 3034224 | 120 | 120 | 540 | 540 | 218-220 | 3 | 33 |
| BR 12.9.4 | 3027090 | 140 | 160 | 540 | 540 | 217-218 | 7 | 153 |

| ProdWo | WerksNr | Po | Ref | Bles | Qua | Di | Br | Lae | Gew | FAB | IFAB-nFAB | Temp | HD | OVZ | Lage | Stoss | LVU |
|--------|---------|----|--------|---------|-----|-------|------|-------|-------|------|-----------|------|----|-----|-----------|---------|-----|
| 201217 | 370876 | 31 | 712404 | 111 | E94 | 160,0 | 2062 | 5944 | 15394 | A1R3 | HS2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 10 |
| 201217 | 370876 | 31 | 847150 | 111 | E94 | 160,0 | 2065 | 5960 | 15458 | A1R3 | HS2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 9 |
| 201217 | 370876 | 31 | 872148 | 111 | E94 | 160,0 | 2065 | 5960 | 15458 | A1R3 | HS2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 8 |
| 201217 | 370876 | 31 | 647151 | 111 | E94 | 160,0 | 2063 | 5950 | 15417 | A1R3 | HS2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 7 |
| 201217 | 370876 | 31 | 012407 | 111 | E94 | 160,0 | 2064 | 5800 | 15036 | A1R3 | HS2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 6 |
| 201215 | 373056 | 8 | 507581 | 111 | H71 | 80,0 | 2525 | 9050 | 14351 | KER3 | KAFU-LMR3 | 540 | 0 | 154 | BR 12.9.4 | 3039617 | 11 |
| 201216 | 371493 | 1 | 027658 | 1415161 | H62 | 26,0 | 1027 | 11565 | 2424 | KER3 | BLH2-LMR3 | 540 | 0 | 55 | BR 12.9.4 | 3039617 | 5 |
| 201216 | 371493 | 1 | 027658 | 1112131 | H62 | 26,0 | 1030 | 11565 | 2431 | KER3 | BLH2-LMR3 | 540 | 0 | 55 | BR 12.9.4 | 3039617 | 4 |

Anz: To: aus fallender Hitze Bienen

| ProdWo | WerksNr | Po | Ref | Bles | Qua | Di | Br | Lae | Gew | FAB | IFAB-nFAB | Temp | HD | OVZ | Lage | Stoss | LVU |
|--------|---------|----|--------|-------|-----|-------|------|-------|-------|------|-----------|------|----|-----|------------|---------|-----|
| 201219 | 374186 | 4 | 633682 | 11121 | Q77 | 200,0 | 2150 | 11845 | 39983 | VWR3 | OER3-VWR3 | 410 | 0 | 79 | RH03 | 0 | 0 |
| 201219 | 374186 | 1 | 116072 | 11121 | Q77 | 180,0 | 2207 | 11137 | 34731 | VWR3 | OER3-VWR3 | 410 | 0 | 71 | RH03 | 0 | 0 |
| 201219 | 372296 | 3 | 342187 | 111 | E94 | 94,0 | 2238 | 9030 | 14912 | VWR3 | VWR3-OER3 | 540 | 0 | 156 | RH03-OE | 0 | 9 |
| 201216 | 371493 | 4 | 528687 | 11121 | H62 | 112,0 | 2238 | 5764 | 11381 | VWR3 | OPCL-LMR3 | 395 | 0 | 43 | RH03-OE | 0 | 8 |
| 201215 | 372222 | 3 | 186271 | 111 | E85 | 82,0 | 1727 | 3626 | 4039 | VWR3 | OPCL-LMR3 | 540 | 0 | 63 | BR 12.9.3 | 3034224 | 11 |
| 201214 | 371650 | 5 | 785892 | 111 | E94 | 180,0 | 2037 | 6322 | 15164 | VWR3 | USKA-LMR3 | 540 | 0 | 119 | BR 12.10.4 | 3039617 | 10 |
| 201218 | 370876 | 25 | 810814 | 121 | E94 | 102,2 | 2051 | 5955 | 9799 | VWR3 | USRA-LMR3 | 540 | 0 | 98 | BR 12.9.3 | 3034224 | 10 |
| 201218 | 373911 | 13 | 358790 | 111 | E94 | 200,0 | 2211 | 7533 | 26149 | VWR3 | HS2-LMR3 | 540 | 0 | 162 | BR 12.9.4 | 3025759 | 13 |
| 201218 | 373911 | 13 | 558789 | 111 | E94 | 200,0 | 2219 | 7507 | 26153 | VWR3 | HS2-LMR3 | 540 | 0 | 162 | BR 12.9.4 | 3025759 | 12 |

Anz: To:

Abbildung 8.2: Programmiererstellung des Alt-Systems

Die mittig dargestellte Listenansicht dient dem Bediener dazu, sich nähere Informationen über bestimmte Waben einzuholen. Hierzu kann er aus den oberen Beständen beliebig Waben auswählen und diese der Liste hinzufügen. Für jede hinzugefügte Wabe erhält er dann die folgenden Informationen:

- Sollproduktionswoche (ProdWo)
- Werksnummer (WerksNr)
- Auftragspositionsnummer (Po)
- Referenznummer (Ref)
- Blechnummern (Bles)
- Qualitätskennzeichnung (Qua)
- Dicke (Di)
- Breite (Br)
- Länge (Lae)
- Gewicht (Gew)
- Nächste Wärmebehandlung (FAB)
- Zuletzt durchgeführter Fertigungsabschnitt (IFAB)

- Nächster offener Fertigungsabschnitt (nFAB)
- Wärmebehandlungstemperatur (Temp)
- Haltedauer (HD)
- Ofenverweilzeit (OVZ)
- Lagerbereich (Lage)
- Stoßnummer (Stoss)
- Lage von unten im Stapel (LVU)

Über den rechts neben der Liste angeordneten Button *Liste sortieren* können die Waben nach bestimmten, vorher zu wählenden Kriterien automatisch sortiert werden.

Nun kann der Ofensteuerer entscheiden, welche Waben er in das aktuelle Produktionsprogramm aufnehmen möchte. Hierzu zieht er die ausgewählten Waben von der mittigen in die untere Liste, welche das Produktionsprogramm repräsentiert. Die Spaltenstruktur ist die gleiche wie zuvor. Die Anordnung der Zeilen kennzeichnet nun allerdings die geplante Produktionsfolge. Der Ofensteuerer kann die Positionierung der Waben entweder manuell vornehmen, indem er die Waben entsprechend nach oben oder unten schiebt oder er betätigt den zweiten Button *Liste sortieren*. Auch hierdurch lässt sich die Liste wieder nach bestimmten Kriterien automatisch sortieren. Wie bereits in Kapitel 5.3 erläutert, wird hier i. d. R. die Wärmebehandlungstemperatur als erstes und die Ofenverweilzeit als zweites Sortierkriterium verwendet. Die Vorgehensweise entspricht somit der *Rule-based Sorting*-Heuristik (vgl. Algorithmus 1).

8.1.2 Das neue System

Das neue System ist mehrschichtig aufgebaut. Die einzelnen Bausteine des in Kapitel 7 vorgestellten Lösungskonzepts (vgl. Abbildung 7.1) wurden in C++ programmiert und zu zwei Anwendungen zusammengefasst. Diese unterscheiden sich jedoch ausschließlich in der Parametrierung des HTFSPD-HGA (insbesondere hinsichtlich der Abbruchkriterien) voneinander.

Mit der Implementierung zweier verschiedener Optimierungsstrategien wurde das Ziel verfolgt, auf der einen Seite eine ausführliche und auf der anderen Seite

eine Kurzoptimierung bereitzustellen. Dabei soll die ausführliche Optimierung mit entsprechend höherer Laufzeit speziell für die Verbesserung der Güte des aktuellen Produktionsprogramms sorgen, während die Kurzoptimierung eher dazu gedacht ist, um kurzfristig auf Änderungen reagieren zu können und möglichst schnell zulässige und zumindest halbwegs gute Lösungen zu generieren (*Online-Aspekt*).

Beide Anwendungen laufen von dem ursprünglichen Planungsprogramm getrennt auf einem separaten Server und werden in regelmäßigen Abständen automatisch gestartet. Die ausführliche Variante wird derzeit mit einem Zeitlimit von 600 s in einem Rhythmus von 675 s ausgeführt. Die Laufzeit der kurzfristigen Variante ist dagegen auf 15 s begrenzt und wird alle 45 s angestoßen. Die Ergebnisse werden jeweils in der Datenbank hinterlegt.

Um den Ofensteuermännern die Umstellung auf das neue System etwas zu erleichtern, wurde die neue Programmierstellungsansicht in das ursprüngliche VB6-Programm integriert. Wie in Abbildung 8.3 erkennbar, steht dem Bediener nun bei Programmstart eine zusätzliche Option (*Automatische Optimierung*) zur Verfügung, mit der er die Möglichkeit hat, zwischen Neu- und Alt-System auszuwählen. Diese Auswahloption wurde bewusst gestaltet, um während der Entwicklungsphase des neuen Programms zur Not immer auf das altbewährte, stabile System zurückgreifen zu können. Zukünftig (und auch derzeit schon der Fall) sollen die Ofensteuermänner ausschließlich mit dem neuen System arbeiten.



Abbildung 8.3: Startoberfläche des neuen Systems

Die Oberfläche des neuen Systems ist in Abbildung 8.4 (bzw. im Anhang Abbildung A.3) dargestellt. Das aktuelle Produktionsprogramm wird, wie zuvor auch, in der unteren Listenansicht abgebildet. In dieser werden neben den bereits bekannten Spalten nun allerdings noch zusätzlich die folgenden Infor-

mationen angezeigt:

- Folge­nummer (FNr)
- Wa­ben­num­mer (WB)
- La­ge von oben im Stapel (LVO)
- Brennschritte (BRFAB)
- Brennendtemperaturen (BRTEMP)
- Brennzeit (BRZ)
- Frü­he­ster Ab­la­ge­zeit­punkt auf dem Ofeneinlaufrollgang (FAZ Einlauf)
- Frü­he­ster An­fangs­zeit­punkt der Wär­me­be­hand­lung im Ofen (FAZ Ofen)
- Frü­he­ster End­zeit­punkt der Wär­me­be­hand­lung im Ofen (FEZ Ofen)
- Frü­he­ster Ent­nah­me­zeit­punkt vom Ofenauslaufrollgang (FEZ Auslauf)
- Frü­he­ster An­fangs­zeit­punkt des Brennschneidens auf einem Brennportal (FAZ BP)
- Frü­he­ster End­zeit­punkt des Brennschneidens auf einem Brennportal (FEZ BP)
- Wa­rte­zeit aus­ge­hend von der Ofeneinfahrt der Vorgängerwabe (Warten)
- Vor­ge­sch­la­gene mi­ni­male Tem­pe­ra­tur­lücke zur Vorgängerwabe (Luecke)

Vorgaben für die Optimierung:

Anzahl zur Verfügung stehender Brennanlagen (für Brennen aus fallender Hitze):

Wechsel des WBH-Verfahrens:

Anlassen
 Vorwärmen
 Kantenentspannen
 Effusionsglühen

Umstapler für den Kran bestrafen: schwach
 mittel
 hart

Zurückstellungen:

| Ref | WB | Bles | ProdMo | Qua | Di | Br | Lae | Ge | ▲ |
|--------|----|------|--------|-----|-------|------|------|-----|---|
| 594837 | 3 | 111 | 201220 | E94 | 130,0 | 2054 | 5900 | 123 | |
| 456660 | 5 | 111 | 201213 | E94 | 132,0 | 2017 | 3900 | 81 | |
| 739377 | 3 | 111 | 201221 | E94 | 90,0 | 2031 | 5950 | 85 | |

Nächste Einfahrt: **81954** um: **2012-05-03 15:58:40** mit Lücke: **1500** mm Einzel-Fahrweise Aktualisierung in: **53** sek

| Fkz | Ref | WB | Bles | ProdMo | Qua | Di | Br | Lae | Gev | FAB | nFAB | Temp | HD | OVZ | Lage | Stoss | LVO | BRFAB | BRTEMP | BRZ | ▲ |
|-----|--------|----|-------|--------|-----|-------|------|-------|-------|------|------|------|----|-----|-----------|---------|-----|-------|--------|-----|----|
| 1 | 093870 | 1 | 111 | 201217 | 077 | 150,0 | 2423 | 5466 | 16350 | AIR3 | AKR3 | 620 | 5 | 299 | RH03-0A | 0 | 0 | BCH2 | 80 | 78 | |
| 2 | 894079 | 1 | 111 | 201217 | 077 | 150,0 | 2420 | 5626 | 16280 | AIR3 | AIR3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 79 | |
| 3 | 893991 | 1 | 111 | 201217 | 077 | 150,0 | 2437 | 5611 | 16030 | AIR3 | AIR3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 79 | |
| 4 | 194073 | 1 | 111 | 201217 | 077 | 150,0 | 2418 | 5714 | 16520 | AIR3 | AIR3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 80 | |
| 5 | 594105 | 1 | 111 | 201217 | 077 | 150,0 | 2470 | 5757 | 16760 | AIR3 | AIR3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 81 | |
| 6 | 389106 | 1 | 111 | 201217 | 077 | 150,0 | 2415 | 5684 | 16380 | AIR3 | AIR3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 80 | |
| 7 | 294096 | 1 | 111 | 201217 | 077 | 150,0 | 2418 | 5611 | 16190 | AIR3 | AIR3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 79 | |
| 8 | 781954 | 1 | 111 | 201217 | 077 | 150,0 | 2407 | 5669 | 16140 | AIR3 | OER3 | 620 | 5 | 299 | RH03-OE | 0 | 0 | BCH2 | 80 | 79 | 20 |
| 9 | 193262 | 1 | 111 | 201217 | 077 | 150,0 | 2425 | 6000 | 16230 | AIR3 | LMR3 | 620 | 5 | 299 | BR 12.9.5 | 3060906 | 1 | BCH2 | 80 | 82 | 20 |
| 10 | 010454 | 3 | 121 | 201218 | E94 | 100,0 | 2065 | 5970 | 9678 | VMR3 | LMR3 | 475 | 0 | 102 | BR 12.9.3 | 3056365 | 3 | BNH2 | 300 | 66 | 20 |
| 11 | 111327 | 4 | 111 | 201218 | E94 | 120,0 | 2065 | 5567 | 10829 | VMR3 | LMR3 | 425 | 0 | 97 | BR 12.9.3 | 3056364 | 1 | BNH2 | 300 | 68 | 20 |
| 12 | 111327 | 5 | 121 | 201218 | E94 | 120,0 | 2065 | 5602 | 10897 | VMR3 | LMR3 | 425 | 0 | 97 | BR 12.9.3 | 3056364 | 2 | BNH2 | 300 | 68 | 20 |
| 13 | 972143 | 2 | 111 | 201218 | E94 | 120,0 | 2063 | 5950 | 11563 | VMR3 | LMR3 | 425 | 0 | 97 | BR 12.9.3 | 3056365 | 1 | BNH2 | 300 | 70 | 20 |
| 14 | 478161 | 4 | 131 | 201215 | E94 | 70,0 | 2050 | 5950 | 6703 | VMR3 | LMR3 | 550 | 0 | 119 | BR 12.5.2 | 3057600 | 1 | BFH2 | 300 | 58 | 20 |
| 15 | 832870 | 1 | 11121 | 201218 | E95 | 90,0 | 2237 | 16702 | 26500 | VMR3 | LMR3 | 540 | 0 | 142 | BR 12.9.4 | 3060289 | 1 | BFH2 | 300 | 127 | 20 |
| 16 | 579484 | 1 | 11121 | 201218 | E95 | 100,0 | 2228 | 15963 | 28500 | VMR3 | LMR3 | 540 | 0 | 158 | BR 12.9.4 | 3060289 | 2 | BFH2 | 300 | 128 | 20 |

Abbildung 8.4: Die neue Programmerstellungsansicht (1)

Die Folgenummer stellt die aktuelle Reihenfolgeposition einer Wabe dar und gibt gleichzeitig Auskunft über deren Ofenfahrweise, da Waben innerhalb eines Verbands über identische Folgenummern verfügen. Die Angabe der Wabenummer und der Lage von oben (anstatt der Lage von unten) im Stapel entstammt eher den Wünschen der Ofensteuermänner. Die restlichen Spalten enthalten die für die Ofensteuermänner interessantesten Informationen, welche von der Simulation der aktuellen Produktionsfolge geliefert werden. Dabei werden die Spalten, in der die acht letztgenannten Daten enthalten sind, erst durch Scrollen nach rechts ersichtlich. Abbildung 8.5 (bzw. im Anhang Abbildung A.4) zeigt die zu Abbildung 8.4 entsprechende Ansicht.

Vorgaben für die Optimierung:

Anzahl zur Verfügung stehender Brennanlagen (für Brennen aus fallender Hitze):

Wechsel des WBH-Verfahrens:

Anlassen
 Vorwärmen
 Kantenentspannen
 Effusionsglühen

Umstapler für den Kran bestrafen: schwach mittel hart

Zurückstellungen:

| Ref | WB | Bles | ProdMo | Qua | D1 | Br | Lae | Ge |
|--------|----|------|--------|-----|-------|------|------|-----|
| 594837 | 3 | 111 | 201220 | E94 | 130,0 | 2054 | 5900 | 123 |
| 456660 | 5 | 111 | 201213 | E94 | 132,0 | 2017 | 3900 | 81 |
| 739377 | 3 | 111 | 201221 | E94 | 90,0 | 2031 | 5950 | 85 |

Nächste Einfahrt: **81954** um: **2012-05-03 15:58:40** mit Lücke: **1500** mm Einzel-Fahrweise Aktualisierung in: **41** sek

| z | FA2 Einlauf | FA2 Ofen | FEZ Ofen | FEZ Auslauf | FA2 BP | FEZ BP | Warten | Luecke |
|---|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|--------|--------|
| | | 2012-05-03 11:42:40 | 2012-05-03 16:51:40 | 2012-05-03 17:34:40 | 2012-05-03 17:49:40 | 2012-05-03 19:08:40 | -256 | 0 |
| | | 2012-05-03 12:19:40 | 2012-05-03 17:28:40 | 2012-05-03 18:11:40 | 2012-05-03 18:26:40 | 2012-05-03 19:45:40 | 37 | 1500 |
| | | 2012-05-03 13:06:40 | 2012-05-03 18:15:40 | 2012-05-03 18:58:40 | 2012-05-03 19:13:40 | 2012-05-03 20:33:40 | 47 | 1500 |
| | | 2012-05-03 13:47:40 | 2012-05-03 18:56:40 | 2012-05-03 19:39:40 | 2012-05-03 19:54:40 | 2012-05-03 21:15:40 | 41 | 1500 |
| | | 2012-05-03 14:37:40 | 2012-05-03 19:46:40 | 2012-05-03 20:29:40 | 2012-05-03 20:44:40 | 2012-05-03 22:04:40 | 50 | 1500 |
| | | 2012-05-03 15:20:40 | 2012-05-03 20:29:40 | 2012-05-03 21:12:40 | 2012-05-03 21:27:40 | 2012-05-03 22:46:40 | 43 | 1500 |
| | 2012-05-03 14:48:40 | 2012-05-03 15:58:40 | 2012-05-03 21:07:40 | 2012-05-03 21:50:40 | 2012-05-03 22:05:40 | 2012-05-03 23:24:40 | 38 | 1500 |
| | 2012-05-03 16:01:40 | 2012-05-03 16:39:40 | 2012-05-03 21:48:40 | 2012-05-03 22:31:40 | 2012-05-03 22:46:40 | 2012-05-04 00:08:40 | 41 | 1500 |
| | 2012-05-03 16:48:40 | 2012-05-03 21:51:40 | 2012-05-03 23:43:40 | 2012-05-03 23:53:40 | 2012-05-04 00:08:40 | 2012-05-04 01:14:40 | 312 | 13000 |
| | 2012-05-03 21:54:40 | 2012-05-03 22:05:40 | 2012-05-03 23:53:40 | 2012-05-04 00:03:40 | 2012-05-04 00:18:40 | 2012-05-04 01:26:40 | 14 | 7500 |
| | 2012-05-03 22:08:40 | 2012-05-03 23:01:40 | 2012-05-04 00:49:40 | 2012-05-04 00:59:40 | 2012-05-04 01:14:40 | 2012-05-04 02:22:40 | 56 | 7000 |
| | 2012-05-03 23:04:40 | 2012-05-03 23:13:40 | 2012-05-04 01:01:40 | 2012-05-04 01:11:40 | 2012-05-04 01:26:40 | 2012-05-04 02:36:40 | 12 | 7000 |
| | 2012-05-03 23:16:40 | 2012-05-03 23:35:40 | 2012-05-04 01:46:40 | 2012-05-04 02:07:40 | 2012-05-04 02:22:40 | 2012-05-04 03:20:40 | 22 | 7000 |
| 7 | 2012-05-03 23:38:40 | 2012-05-03 23:45:40 | 2012-05-04 02:19:40 | 2012-05-04 02:42:40 | 2012-05-04 02:57:40 | 2012-05-04 05:04:40 | 10 | 8000 |
| 8 | 2012-05-03 23:48:40 | 2012-05-03 23:55:40 | 2012-05-04 02:46:40 | 2012-05-04 03:09:40 | 2012-05-04 03:24:40 | 2012-05-04 05:32:40 | 10 | 7500 |

Abbildung 8.5: Die neue Programmerstellungsansicht (2)

Ein großer Unterschied gegenüber dem Alt-System besteht darin, dass der Ofensteuermann sich die Waben, die er als nächstes in den Ofen fahren möchte, grundsätzlich nicht mehr selbst zusammensuchen muss. Das neue Programm entnimmt sich minütlich das jeweils durch die beiden Optimierungsanwendungen erstellte Produktionsprogramm aus der Datenbank und aktualisiert die Listenansicht dementsprechend. Somit ist nun prinzipiell eine vollautomatisierte Programmerstellung möglich. Aufgrund von im betrieblichen Alltag immer wieder vorkommenden Störungen, Verzögerungen oder erforderlichen Umla-

nungen, ist es dennoch wichtig, dem Ofensteuerer verschiedene Mechanismen bereitzustellen, mit denen er manuell in die automatische Programmierung eingreifen kann. Diese Mechanismen werden im Folgenden vorgestellt.

Zunächst besteht weiterhin grundsätzlich die Möglichkeit, die Reihenfolgepositionen der Waben in der Listenansicht zu ändern. Eine solche Änderung wird in den beiden Optimierungsanwendungen derart berücksichtigt, dass die in der aktuellen Listenansicht dargestellte Produktionsfolge immer in Form eines Individuums im Rahmen des HTFSPD-HGA mit initialisiert wird. Vom Ofensteuerer durchgeführte Positionsänderungen wirken somit für die Optimierung wie ein Startvorschlag. Nur wenn im Laufe des Verfahrens eine besser bewertete Lösung gefunden wird, bildet diese das neue Produktionsprogramm.

Es gibt jedoch auch begründete Situationen, in denen Änderungen durch die Steuerer beibehalten werden sollten. Daher wurde ihnen die Möglichkeit eingeräumt, bestimmte Positionen zu fixieren. Diese dürfen von den beiden Optimierungsanwendungen dann nicht mehr geändert werden. In der Listenansicht sind die Zeilen fixierter Positionen durch eine gelbe Hintergrundmarkierung zu erkennen (vgl. Abbildung 8.4). Hier ist allerdings im Prinzip nur die neunte Position fixiert, da Waben, die im Ofen oder auf dem Einlauf- bzw. Auslaufrollgang liegen, grundsätzlich immer fixiert sind. Selbstverständlich besteht für den Ofensteuerer jederzeit auch die Möglichkeit vorgenommene Fixierungen wieder aufzuheben.

Als weitere Option kann der Steuerer bestimmte Waben auch zurückstellen lassen. Diese werden dann aus der unteren Listenansicht entfernt und in der Listenansicht rechts oben eingetragen. Eine Zurückstellung ist gleichbedeutend mit einer zeitweisen Löschung bzw. Sperrung der Waben aus dem Produktionsprogramm. Die Gründe hierfür können z. B. Fehlbuchungen oder betriebsbedingte Umplanungen sein, die datentechnisch nicht erfasst, dafür aber dem Ofensteuerer bekannt sind. Zurückgestellte Waben dürfen von den beiden Optimierungsanwendungen nicht mehr berücksichtigt bzw. wieder eingeplant werden. In der in Abbildung 8.4 dargestellten Situation wurden beispielsweise drei Waben zurückgestellt. Sie werden erst wieder berücksichtigt, wenn der Ofensteuerer sie freigibt.

Neben den bisher erläuterten Eingriffsoptionen besteht für den Ofensteuer-

mann zusätzlich die Möglichkeit, gewisse Vorgaben an die Optimierung zu übergeben (siehe Bereich links oben in Abbildung 8.4). Zum einen kann die Anzahl der aktuell für das Brennen aus fallender Hitze zur Verfügung stehenden Brennanlagen editiert werden. Zum anderen besteht die Möglichkeit, in die WBH-Blockbildung einzugreifen. Hierzu muss ein Haken bei *Wechsel des WBH-Verfahrens* gesetzt und anschließend das aktuell gewünschte WBH-Verfahren ausgewählt werden. Beim nächsten Optimierungsdurchlauf wird dieser WBH-Block dann unabhängig von den jeweiligen Sollproduktionswochen an der frühestmöglichen Position eingeplant. Ist bei *Wechsel des WBH-Verfahrens* kein Haken gesetzt, so verfährt die WBH-Blockbildung wie in den Kapiteln 7.1 und 7.3 beschrieben. Zuletzt kann der Ofensteuermann noch beeinflussen, wie stark eventuell für den Kran erforderliche Umstapelaktionen in der Bewertung des HTFSPD-HGA bestraft werden sollen. Dies geschieht bei *Umstapler für den Kran bestrafen* über die Ausprägungen *schwach, mittel, hart*, welche gleichbedeutend mit Änderungen des Parameters t_u auf 1, 3 oder 30 min sind. Die Änderung betrifft dann jedoch nur die Bewertung der Individuen innerhalb des HTFSPD-HGA. Die beste gefundene Lösung wird abschließend immer noch einmal mit dem Standardwert $t_u = 3$ simuliert. Insbesondere eine stärkere Bestrafung von Umstapelaktionen ist in den Fällen sinnvoll, in denen der Kran eher den Engpass bildet als der Ofen, d. h. die Beschickung anderer Anlagen gerade eiliger ist als die des Ofens.

Während der Implementierungsphase des neuen Systems hat sich die zeitgleiche Programmierung einer Visualisierung der von der Simulation ausgegebenen Daten zwecks Verifizierung als durchaus hilfreich erwiesen. Ursprünglich nur für Testzwecke gedacht, wurde daraus schließlich eine zusätzliche Anwendung erstellt, die letztendlich auch aus dem neuen VB6-Programm aufgerufen werden kann. Die wiederum in C++ programmierte Visualisierung öffnet sich durch Anklicken des Buttons *Visualisieren*. Abbildung 8.6 (bzw. im Anhang Abbildung A.5) zeigt eine beispielhafte Ansicht des Programms.

Direkt links oben wird der aktuelle Wert eines im Programm integrierten Timers im Zeitformat [hh:mm] dargestellt. Unmittelbar darunter werden die zu diesem Zeitpunkt von der Simulation prognostizierten, verfügbaren Restkapazitäten des Ofens und der entsprechenden Rollgänge (in Millimetern) angezeigt. Rechts daneben sind fünf Buttons angeordnet, mit denen die Simulation

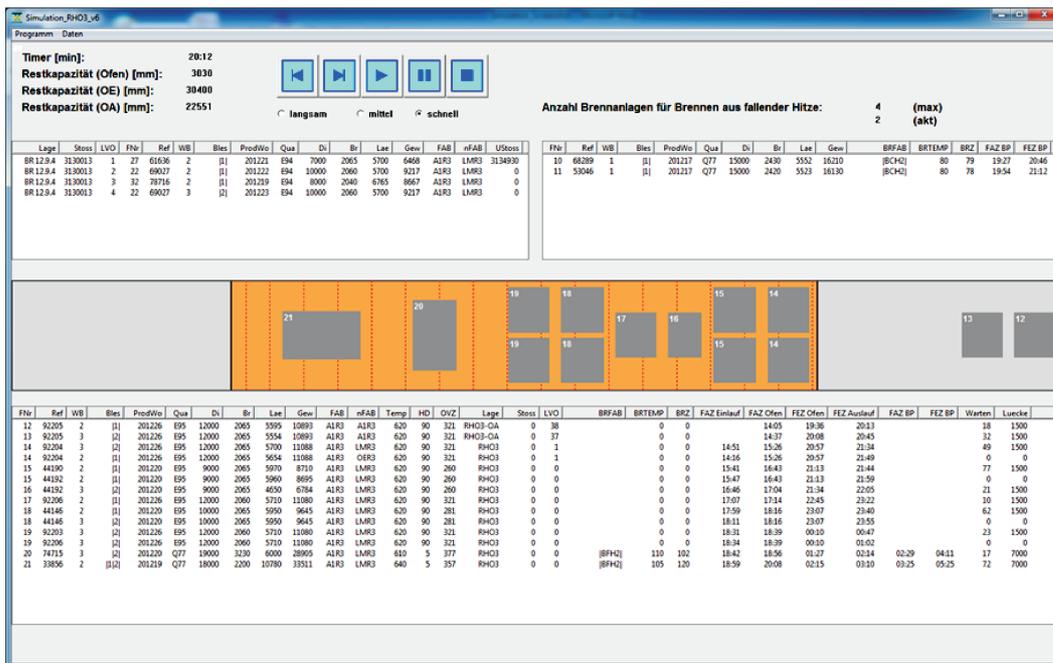


Abbildung 8.6: Screenshot der Visualisierung

bzw. die Visualisierung gesteuert werden kann. Der Reihe nach haben die Buttons folgende Bedeutungen:

- eine Minute zurück
- eine Minute vor
- Start
- Pause
- Stopp

Beim Drücken des *Start*-Buttons wird der Timer automatisch mit bestimmter Geschwindigkeit jeweils minutenweise erhöht. Hierbei kann zwischen drei Geschwindigkeiten (*langsam, mittel, schnell*) gewählt werden.

Die eigentliche Visualisierung erfolgt im mittleren Bereich. Hier werden die von der Simulation zum aktuellen Zeitpunkt des Timers berechneten Belegungen des Ofens bzw. des Ofenein- und Ofenauslaufrollgangs abgebildet. In der in Abbildung 8.6 dargestellten Situation befinden sich demnach 12 Waben im Ofen (davon acht in Verband-Fahrweise) und zwei Waben auf dem Ofenauslaufrollgang. Die Waben sind links oben durch ihre Folge Nummern gekennzeichnet.

Neben dieser Visualisierung werden die zum jeweiligen Zeitpunkt des Timers produktionsprogrammrelevanten Daten in Form von drei Listenansichten angezeigt. Die Daten der Waben, die sich im Ofen bzw. auf dem Ofenein- oder Ofenauslaufrollgang befinden, sind der unteren Listenansicht zu entnehmen. Die Spalten sind analog zu der Listenansicht des neuen VB6-Programms gestaltet. Die Liste links oben enthält zusätzlich bestimmte Wabendaten des Stapels, in dem sich die nächste zum Ofen zu transportierende Wabe befindet. In dem in Abbildung 8.6 dargestellten Beispiel ist dies die Wabe mit der Referenznummer 69027 (Folgenummer 22). Bevor diese Wabe vom Kran entnommen werden kann, muss zunächst allerdings die darüber liegende Wabe umgestapelt werden. Der hierfür von dem in Kapitel 6.3.1 beschriebenen *Umstapelalgorithmus* vorgeschlagene Zielstoß wird in der letzten Spalte angezeigt (hier: $UStoss = 3134930$). Die dritte Listenansicht (rechts oben) enthält schließlich die Waben, die sich zum aktuellen Zeitpunkt des Timers in oder unmittelbar vor einer Brennanlage befinden. Direkt darüber steht die vom Ofensteuerer vorgegebene maximal verfügbare Anzahl der Brennanlagen zum Brennen aus fallender Hitze (max) sowie die Anzahl der in der aktuell dargestellten Situation belegten Anlagen (akt).

Die Implementierung des neuen Systems zusammenfassend, wurde das alte Programm zur Erstellung des Produktionsprogramms um die automatische Optimierung erweitert, zudem jeweils eine ausführliche und eine kurzzeitige Optimierung hinterlegt sowie letztendlich eine Visualisierung der Simulationsdaten bereitgestellt. Abschließend sei hervorgehoben, dass die mit diesen Anwendungen verbundene Planungsweise zur Produktionsprogrammerstellung des RHO3 bei den Dillinger Hüttenwerken derzeit wie in diesem Kapitel beschrieben vollzogen wird.

8.2 Quantitative und qualitative Bewertung

Nach der Beschreibung der Implementierung des neuen Systems folgt nun eine abschließende Gesamtbewertung. Die Verbesserungen gegenüber dem Alt-System sind jedoch nicht alle vollständig quantifizierbar. Aus diesem Grund wird im Anschluss an die Vorstellung der numerischen Ergebnisse noch eine

qualitative Bewertung vorgenommen.

8.2.1 Numerische Ergebnisse

Im Rahmen dieser Arbeit wurden bereits mehrere Rechenergebnisse vorgestellt. Meist dienten diese dazu, die Vorteile bzw. das Potential einzelner im Gesamtkonzept integrierter Bausteine zu belegen. Nun erfolgt ein konkreter Vergleich zwischen Alt- und Neu-System.

Für diesen Vergleich werden reale, in der Vergangenheit am RHO3 produzierte Produktionsprogramme herangezogen. Da verschiedene Daten und Informationen im Nachhinein jedoch nicht mehr abrufbar sind, müssen für einen fairen Vergleich zwischen Alt- und Neu-System einige Annahmen getroffen werden. Zum einen dürfen nicht die in der Datenbank hinterlegten Ofenein- und ausfahrzeiten als Grundlage für das alte System herangezogen werden, da diese eventuell von betrieblichen Störungen und Verzögerungen betroffen gewesen sein könnten. Auch ein reiner Vergleich mit der real produzierten Reihenfolge wäre nicht fair, da heute keine Informationen mehr über frühere Lagerpositionen oder konkrete Bereitstellungszeitpunkte der Waben vorliegen. Folglich wurde sowohl die Umstapelproblematik als auch der *Online*-Aspekt bei den abschließenden Rechentests vernachlässigt. Die Tests basieren somit lediglich auf einer Menge von Waben, die den RHO3 innerhalb eines bestimmten Zeitraums durchlaufen haben. Aufgrund ihrer Relevanz für den Ofen bzw. die Optimierung werden nur Anlasswaben betrachtet. Zudem wird beim Start von einem leeren Ofen ausgegangen.

Die bis hierhin erläuterten Testannahmen betreffen jeweils beide betrachteten Systeme. Des Weiteren sind noch jeweils für Alt- und Neu-System separate Annahmen erforderlich. Das Alt-System wird wieder durch die (RBS)-Heuristik repräsentiert. Die Lückenberechnung erfolgt hier gemäß Gleichung 5.4, wobei die Vorgabe $f_{templ} = 600$ gesetzt wird. Da sie in der Vergangenheit von den Ofensteuerern so gut wie nie vorgenommen wurde, wird die Verbandfahrweise im Alt-System nicht in Betracht gezogen. Das neue System wird durch das in Abbildung 7.1 dargestellte gesamtheitliche Lösungskonzept repräsentiert. Folglich sind hier sowohl Verband- als auch *Hollomon*-Fahrweise

berücksichtigt. Lediglich die WBH-Blockbildung entfällt, da den Tests nur Anlasswaben zugrunde liegen. Die Temperaturlücken werden gemäß Algorithmus 11 berechnet.

Es sei nochmals hervorgehoben, dass die getroffenen Annahmen größtenteils aufgrund fehlenden Datenmaterials erforderlich sind. Bei der konkreten Gestaltung der Annahmen wurde stets das Ziel eines fairen und realistischen Vergleichs zwischen Alt- und Neu-Systems verfolgt.

Die Ergebnisse der abschließenden Rechentests sind Tabelle 8.1 zu entnehmen. Unter den insgesamt 25 getesteten Instanzen sind auch die Datensätze, die bereits aus den vorherigen Auswertungen bekannt sind. Bei den Ergebnissen

Tabelle 8.1: Vergleich der Rechenergebnisse zwischen Alt- und Neu-System

| Daten- satz | Alt-System (mit RBS) | | Neu-System (mit HTFSPD-HGA) | |
|----------------|----------------------|------|-----------------------------|--------------------|
| | C_{max} | Zeit | $\varnothing C_{max}$ | \varnothing Zeit |
| <i>dat3</i> | 426 | < 1 | 426.0 | 119.2 |
| <i>dat5</i> | 861 | < 1 | 504.0 | 269.4 |
| <i>dat8</i> | 661 | < 1 | 476.0 | 371.6 |
| <i>dat10</i> | 711 | < 1 | 616.0 | 600.0 |
| <i>dat12</i> | 964 | < 1 | 847.2 | 600.0 |
| <i>dat15</i> | 883 | < 1 | 777.0 | 600.0 |
| <i>dat20</i> | 909 | < 1 | 677.4 | 600.0 |
| <i>dat23</i> | 1093 | < 1 | 1043.0 | 600.0 |
| <i>dat30</i> | 1368 | < 1 | 999.6 | 600.0 |
| <i>dat34</i> | 1404 | < 1 | 999.8 | 600.0 |
| <i>dat35</i> | 1183 | < 1 | 977.6 | 600.0 |
| <i>dat36</i> | 1445 | < 1 | 1020.6 | 600.0 |
| <i>dat40</i> | 2194 | < 1 | 1918.8 | 600.0 |
| <i>dat46</i> | 1606 | < 1 | 1304.2 | 600.0 |
| <i>dat48</i> | 1986 | < 1 | 1532.6 | 600.0 |
| <i>dat49</i> | 1969 | < 1 | 1402.0 | 600.0 |
| <i>dat51</i> | 1560 | < 1 | 1242.4 | 600.0 |
| <i>dat53</i> | 2512 | < 1 | 2124.4 | 600.0 |
| <i>dat54</i> | 2522 | < 1 | 2058.6 | 600.0 |
| <i>dat60</i> | 3299 | < 1 | 2476.4 | 600.0 |
| <i>dat64</i> | 2618 | < 1 | 2054.2 | 600.0 |
| <i>dat67</i> | 2887 | < 1 | 2210.6 | 600.0 |
| <i>dat78</i> | 2271 | < 1 | 1640.2 | 600.0 |
| <i>dat80</i> | 3613 | < 1 | 2770.6 | 600.0 |
| <i>dat93</i> | 3184 | < 1 | 2303.4 | 600.0 |

des neuen Systems handelt es sich wieder um die aus jeweils fünf Durchläufen gemittelten Lösungswerte. Der Vergleich zeigt, dass die Gesamtfertigungszeiten des Neu-Systems im Durchschnitt ca. 21 % unter denen des Alt-Systems liegen. Die größte Verbesserung konnte mit 41,46 % bei Datensatz *dat5* erzielt werden. Bezüglich der Rechenzeiten wurde außer bei den ersten drei Datensätzen immer das auf 10 min gesetzte Zeitlimit erreicht. Die höheren Rechenzeiten werden angesichts der mit dem neuen System verbundenen Lösungsverbesserungen allerdings in Kauf genommen.

Die Rechentests belegen eindrucksvoll, dass sich die Investitionen in die Entwicklung eines neuen Konzepts zur Produktionsprogrammierstellung gelohnt haben. Das neue System bringt größtenteils erhebliche Durchsatzsteigerungen mit sich und ist dem Alt-System daher vorzuziehen.

8.2.2 Allgemeine Verbesserungen im Planungs- und Produktionsablauf

Wie bereits angedeutet, sind nicht alle Verbesserungen des in dieser Arbeit entwickelten Lösungskonzepts vollständig quantifizierbar. Die numerischen Ergebnisse aus dem vorangegangenen Unterkapitel sind insbesondere auf die Reihenfolgeoptimierung, die Verband- und *Hollomon*-Fahrweise, sowie auf die Temperaturlückenoptimierung zurückzuführen. Weitere Entwicklungen aus dieser Arbeit, wie beispielsweise der *Umstapelalgorithmus* oder die Berücksichtigung des *Online*-Aspekts, sind hierbei aus bereits genannten Gründen nicht enthalten. Auf diese sowie weitere allgemeine Verbesserungen im neuen Planungs- und Produktionsablauf wird im Folgenden eingegangen.

Die Vorteile des *Umstapelalgorithmus* sind vielseitig. Der grundsätzliche Vorteil liegt darin, unnötige Umstapelaktionen zu vermeiden und somit sowohl die Belastung des Krans als auch die Entnahme- bzw. Bereitstellungsdauern der Waben für den Ofen zu senken. Des Weiteren werden nun die Auswirkungen eventuell erforderlicher Umstapelaktionen bei der Reihenfolgebildung implizit in der Materialflusssimulation des Ofens berücksichtigt. Mit der Option, Umstapelaktionen unterschiedlich stark zu bestrafen, kann der Ofensteuerer den Grad der Berücksichtigung sogar variieren. Dies ist sinnvoll, um auf wech-

selnde Engpasssituationen flexibel reagieren zu können.

Auch die Berücksichtigung des *Online*-Aspekts bringt vielseitige Verbesserungen mit sich. Insbesondere aus der regelmäßigen Aktualisierung von Lagersituation und Reihenfolgebildung resultiert eine gesteigerte Flexibilität im Vergleich zum ursprünglichen, manuellen Planungs- und Produktionsablauf. Neu hinzukommende bzw. freiwerdende Waben können nun schneller berücksichtigt und dadurch gezielter eingeplant werden. Neben dieser gesteigerten Flexibilität profitiert die Reihenfolgebildung aber auch grundsätzlich von den regelmäßigen Aktualisierungen, da mit jeder Aktualisierung auch die Ausführung des HTFSPD-HGA einhergeht, welcher wiederum immer auf die aktuelle Lösung aufbaut. Aus diesem Grund ist anzunehmen, dass sich die Lösungen mit fortwährender Zeit mehr und mehr verbessern, vorausgesetzt die optimale Lösung ist noch nicht gefunden bzw. das Verfahren steckt nicht in einem lokalen Optimum fest.

Weitere Vorteile des neuen Planungs- und Produktionsablaufs resultieren im Wesentlichen aus der Simulation der Produktionsfolge. Ein solches Werkzeug stand dem Ofensteuerer zuvor nicht zur Verfügung. Vielmehr stand er früher vor der schwierigen Aufgabe, die potentiellen Ofenzeiten der Waben selbst errechnen bzw. abschätzen zu müssen. Dabei ist es selbst für einen erfahrenen Steuerer nahezu unmöglich alle diesbezüglich relevanten Einflussgrößen in Betracht zu ziehen. Diese Aufgabe übernimmt nun die Simulation. So werden für jede Wabe nicht nur die Ofenverweilzeiten, sondern beispielsweise auch Zeiten für deren Bereitstellung, die erforderliche Abkühlung am Ofenauslauf oder auch für nachträgliche Brennschritte errechnet. Zudem fließen die Abhängigkeiten zwischen Kran, Ofen und Brennanlagen einschließlich der jeweiligen Kapazitätsbeschränkungen mit ein. Als Ergebnis erhält der Ofensteuerer Planzeiten für die komplette Produktionsfolge. Somit werden zum einen nicht nur verlässliche Ofenein- und ausfahrzeiten für die nächsten wärmezubehandelnden Waben vorgegeben, sondern auch längerfristige Prognosen ermöglicht.

Hervorzuheben ist zudem der verbesserte Umgang mit Vorwärmern. Früher konnte der Ofensteuerer nur grob abschätzen, auf welche Temperatur eine Wabe aufgeheizt werden muss, damit sie während des Brennens aus fallender Hitze nicht unter die vorgegebene Brennendtemperatur fällt. Tritt dieser

Fall ein, weil er sich eventuell verschätzt hat, konnte es sein, dass die Wabe erneut vorgewärmt werden musste. Um dies zu vermeiden wurden die Waben i. d. R. eher mehr als nötig erwärmt, was andererseits jedoch zu längeren Ofenverweilzeiten führte. Die Simulation verwendet hingegen formatabhängige Abkühlkurven und berechnet sowohl format- als auch brennschrittabhängige Brenndauern. Zum einen können somit zuverlässige Zieltemperaturen für Vorwärmer vorgegeben werden. Zum anderen ermöglicht dies auch eine bessere Planungsgrundlage für die Belegung der für das Brennen aus fallender Hitze zur Verfügung stehenden Brennanlagen.

Abschließend gilt es festzuhalten, dass das am RHO3 implementierte System nicht nur grundsätzlich zu verbesserten Produktionsfolgen führt, sondern auch zahlreiche Vereinfachungen für den Ofensteuerer bietet. Diesem steht nun ein vollautomatisches System zur Verfügung, welches ihm sowohl die Erstellung des Produktionsprogramms vollständig abnimmt als auch die jeweiligen Ofenein- und ausfahrzeiten sowie die Längen eventuell erforderlicher Temperaturlücken vorgibt. Dies hat zudem auch den Effekt, dass die Ofenfahrweise nun nicht mehr vom jeweiligen Steuerer abhängig ist. Insbesondere dieser Aspekt ist nicht zu vernachlässigen, da diesbezüglich in der Vergangenheit gravierende Unterschiede zu beobachten waren. Dennoch erwies es sich für die betriebliche Umsetzung aus bereits erläuterten Gründen als sinnvoll bzw. zum Teil erforderlich, den Steuerern verschiedene Optionen bereitzustellen, um manuell in die Planungen eingreifen zu können.

Kapitel 9

Zusammenfassung

In diesem letzten Kapitel werden die zentralen Punkte der vorliegenden Arbeit zunächst noch einmal rekapituliert. Anschließend folgt ein Ausblick auf mögliche Weiterentwicklungen für die Zukunft, ehe die Arbeit mit dem Ziehen eines Fazits abgeschlossen wird.

Die Einleitung der vorliegenden Arbeit diente dazu, dem Leser die grundsätzliche Thematik näher zu bringen und ihn für die Problemstellung zu motivieren. In Kapitel 2 folgte schließlich eine Beschreibung des Materialflusses und der Logistik in integrierten Hüttenwerken. Neben den logistischen, wurden hier auch technische Aspekte einbezogen. Das Hauptaugenmerk lag bei diesem Kapitel allerdings auf der abschließenden Ausarbeitung der speziellen, produktionslogistischen Herausforderungen, mit denen sich ein integriertes Hüttenwerk im Allgemeinen konfrontiert sieht. An dieser Stelle sei der komplexe, sowie anlagen- und transportintensive Produktionsprozess nochmals ausdrücklich hervorgehoben.

Da sich die in der vorliegenden Arbeit behandelte Problemstellung größtenteils mit der Wärmebehandlung von Stahl beschäftigt, wurden hierzu in Kapitel 3 die erforderlichen Grundlagen vermittelt. Hierbei wurde sich allerdings mehr auf den ablauf- und verfahrenstechnischen als auf den metallurgischen Aspekt konzentriert. Zum Ende des Kapitels wurden die verschiedenen Bauarten der bei den Dillinger Hüttenwerken vorherrschenden Wärmebehandlungsöfen mit Hinblick auf logistische Vor- und Nachteile, welche sich aus den unterschiedli-

chen Transport- und Beschickungssystemen ergeben, untersucht.

Mit Kapitel 4 folgte eine Einführung in die Theorie der Maschinenbelegungsprobleme. Nach deren Einordnung in den betrieblichen Entscheidungs- und Planungsprozess wurden grundsätzliche Notationen sowie eine in der Literatur geläufige Klassifizierung vorgestellt. Im Anschluss wurde zunächst die Komplexität derartiger Problemstellungen hervorgehoben, ehe verschiedene exakte und heuristische Lösungsverfahren präsentiert und erläutert wurden. Da es sich bei der in der vorliegenden Arbeit behandelten Problemstellung um eine spezielle Form eines kapazitätsbeschränkten Maschinenbelegungsproblems handelt, wurde diese Klasse zum Abschluss nochmals explizit betrachtet. Hierbei wurden zum einen allgemeine *Batching*-Probleme und zum anderen das *Resource-Constrained Project Scheduling Problem* vorgestellt. Es wurde für beide Varianten jeweils eine grundlegende mathematische Formulierung aufgestellt und schließlich beide miteinander verglichen.

Kapitel 5 widmete sich daraufhin der Beschreibung der in dieser Arbeit behandelten Problemstellung. Zunächst wurde mit dem RHO3 der betreffende Ofen vorgestellt. Hierbei standen insbesondere der Fertigungsablauf sowie die Ofensteuerung im Vordergrund. Dies betraf vor allem die Erläuterung der unterschiedlichen Gründe für Wartezeiten und die Art und Weise der Temperaturregelung. Anschließend wurden mit dem (ZDF)- und dem (EBF)-Modell zwei mathematische Problemformulierungen erarbeitet. Im darauffolgenden Unterkapitel wurde dann ein intuitives Lösungsverfahren (die (RBS)-Heuristik) vorgestellt, welches auch bei den Dillinger Hüttenwerken im alltäglichen Betrieb verwendet wurde und damit die Ist-Situation widerspiegelte. Um überhaupt unterschiedliche Reihenfolgen gegeneinander bewerten zu können, wurde mit *Simulation-1* anschließend die erste Materialflusssimulation des Ofens entwickelt. Mit dem Ziel das Optimierungspotential der (RBS)-Heuristik zu ermitteln, wurde diese zunächst auf einige (kleinere) Datensätze angewandt und mit der Simulation bewertet. Die Ergebnisse wurden dann mit denen verglichen, die sich für diese Datensätze aus der Anwendung bzw. Lösung des (ZDF)- und des (EBF)-Modells ergaben. Der Vergleich konnte jedoch nicht ausreichend zur Analyse des Optimierungspotentials beitragen, da das Lösen der beiden gemischt-ganzzahligen Modelle sich als zu rechenintensiv herausstellte (Abbruch nach einer maximalen Rechenzeit von fünf Stunden). Mit der

anschließenden Entwicklung des *HTFSPD-Branch&Bound*-Verfahrens stand dann jedoch ein exaktes Lösungsverfahren zur Verfügung, mit dem sich zumindest kleine Instanzen in annehmbarer Zeit lösen ließen. Mit Hilfe dieser Ergebnisse konnte das bereits vermutete Optimierungspotential bezüglich der (RBS)-Heuristik letztendlich belegt werden.

Das nächste Kapitel beschäftigte sich zunächst mit der durch die Option der Verband-Fahrweise entstehenden Problemerweiterung. Mit dem (VF)-Modell und der (VF)-Heuristik wurde sowohl ein exakter als auch ein heuristischer Lösungsansatz entwickelt, um aus einer vorgegebenen Wabenmenge möglichst optimale Verbände zu bilden. Das Zielkriterium hierfür war die Minimierung der Gesamtlänge aus den betrachteten Waben bzw. Verbänden. Da die (VF)-Heuristik bei allen getesteten Datensätzen die diesbezüglich optimale Lösung schneller gefunden hat als das (VF)-Modell, wurde im Folgenden nur noch das heuristische Verfahren betrachtet. Allgemein ließ sich anhand verschiedener Rechentests feststellen, dass die Durchlaufzeiten bei Berücksichtigung der Verband-Fahrweise zum Teil stark unter denen bei reiner Anwendung der Einzel-Fahrweise resultierenden Zeiten lagen.

Im anschließenden Unterkapitel wurde mit der *Hollomon*-Fahrweise eine zusätzliche Problemerweiterung vorgestellt, welche ebenfalls zu Durchlaufzeitenreduzierungen beitragen konnte. Nun galten Wärmebehandlungsvorgaben nicht mehr per se als fix, sondern konnten in einem gewissen Rahmen variiert werden. Voraussetzung war, dass der durch den *Hollomon*-Parameter charakteristische Wärmebehandlungseffekt sich nicht ändern durfte. Schließlich ist das bereits bestehende *HTFSPD-Branch&Bound*-Verfahren um den Aspekt der *Hollomon*-Fahrweise erweitert worden, woraus das *HTFSPD-HF-Branch&Bound*-Verfahren resultierte. Mit diesem Verfahren ließ sich das mit der *Hollomon*-Fahrweise verbundene Verbesserungspotential wiederum anhand einiger Rechentests belegen.

Den Abschluss von Kapitel 6 bildete mit Unterkapitel 6.3 die Einbeziehung bestimmter dem Ofen vor- und nachgelagerter Prozesse. Zum einen betraf dies die Stapel- bzw. Umstapelproblematik. Mit der Zielsetzung unnötige Umstapelaktionen zu vermeiden wurde mit dem *Umstapelalgorithmus* ein Verfahren entwickelt, welches dem Kran für jedes Umstapeln jeweils einen geeigneten

Zielstapel vorschlägt. Das Verfahren wurde zudem in die erweiterte Materialflusssimulation des Ofens (*Simulation-2*) aufgenommen. Von da an konnten in der Reihenfolgebewertung auch unterschiedliche Bereitstellungszeiten der Waben berücksichtigt werden. Zudem wurde die Simulation um eventuell durch nachgelagerte Prozesse entstehende Wartezeiten erweitert. Dies betraf zum einen die Einbindung des Abkühlprozesses auf dem Ofenauslaufrollgang und zum anderen die Berücksichtigung der für das Brennen aus fallender Hitze zur Verfügung stehenden Brennanlagenkapazitäten. Eine Wiederholung der Rechentests unter Verwendung von *Simulation-2* hatte schließlich ergeben, dass das Optimierungspotential der (RBS)-Heuristik noch deutlich größer war, als es bereits aus den Tests mit *Simulation-1* hervorging. Allerdings wurde auch deutlich, dass die neue Bewertungsmethode bereits etwas rechenintensiver war.

Da die vorgestellten exakten Lösungsverfahren aufgrund ihrer Rechenzeiten für den betrieblichen Einsatz nicht infrage kamen, wurde in Kapitel 7 schließlich die Entwicklung eines heuristischen Lösungskonzepts beschrieben. Dabei galt es insbesondere, die zuvor meist separat betrachteten Teilprobleme zu einem gesamtheitlichen Verfahren zusammenzuführen. Zunächst wurde mit der WBH-Blockbildung eine Methode zur Problemzerlegung entwickelt. Hierbei wurden die Waben je nach Wärmebehandlungsverfahren in verschiedene Blöcke eingeteilt, um später dann für jeden Block separat (bzw. nur für den eiligsten) eine Reihenfolgeoptimierung durchzuführen. Zur Ermittlung des eiligsten WBH-Blocks wurden die Sollproduktionswochen der Waben hinzugezogen. Den zweiten Schritt bildete dann die Ausführung der (VF)-Heuristik.

Es folgte mit der Beschreibung des HTFSPD-HGA der Kern des in der vorliegenden Arbeit entwickelten Lösungskonzepts. Zunächst wurden die beiden Bestandteile des hybriden Ansatzes einzeln erläutert. Dies betraf zum einen den reinen genetischen Algorithmus HTFSPD-GA und zum anderen den *Tabu Search* HTFSPD-TS. Der HTFSPD-HGA entstand schließlich durch Integration des HTFSPD-TS in den HTFSPD-GA. Die Idee dahinter war, gleichzeitig die Vorteile beider Verfahren auszunutzen. Beim genetischen Algorithmus lagen diese eher in der weiträumigen Durchsuchung des Lösungsraums, während der *Tabu Search* mehr zur Suchintensivierung in bestimmten Bereichen beitragen sollte. Letzteres betraf u. a. das in der *Hollomon*-Fahrweise verborgene Optimierungspotential. Die darauffolgenden Rechentests zeigten, dass der

HTFSPD-HGA für die kleineren Datensätze immer die optimale Lösung gefunden hat. Dies ließ sich anhand der Ergebnisse des *HTFSPD-Branch&Bound*-Verfahrens nachweisen. Für größere Datensätze konnte dieser Nachweise allerdings nicht erbracht werden, da das *HTFSPD-Branch&Bound*-Verfahren für diese die Optima nicht innerhalb des vorgegebenen Zeitraums finden konnte. Es ließ sich allerdings festhalten, dass die nach 10 min mit dem HTFSPD-HGA erhaltenen Lösungen zum Teil wesentlich besser waren, als die des *HTFSPD-Branch&Bound*-Verfahrens bei fünfstündiger Laufzeit.

Neben dem HTFSPD-HGA bestand die Entwicklung des gesamtheitlichen Lösungskonzepts jedoch auch darin die Simulation (erneut) um zwei wichtige Inhaltspunkte zu erweitern. Dies betraf zunächst die Einbindung bzw. Berücksichtigung des *Online*-Aspekts. An dieser Stelle konnte insbesondere aufgezeigt werden, wie negativ sich im Zeitablauf das Fehlen oder die Nichtberücksichtigung bestimmter Informationen auf die Qualität der Lösung bzw. des Produktionsprogramms auswirken kann. Dieser Punkt sprach letztendlich auch dafür, die Optimierung mehrmals in regelmäßigen Abständen zu aktualisieren. Hierfür bedurfte es dann schließlich noch einigen Anpassungen an der Reihenfolgebewertung sowie der WBH-Blockbildung. Die zweite Erweiterung betraf die Optimierung der Temperaturlücken. Hierzu wurde zum einen eine Verbesserung der ursprünglichen Ofenzonenregelung vorgenommen. Zum anderen wurde mit Algorithmus 11 ein analytischer Ansatz aus der Wärmelehre zur Vorgabe neuer Lückenlängen entwickelt. Dieser hat nicht nur zu einer viel detaillierteren Berechnung gegenüber zuvor beigetragen, sondern auch aufgezeigt, dass die ursprünglichen Lückenvorgaben teilweise wesentlich zu lang waren. Mit den beiden Erweiterungen durch den *Online*-Aspekt und die Temperaturlückenoptimierung entstand schließlich mit *Simulation-3* die letztendlich auch im Realbetrieb implementierte Version der Materialflusssimulation des Ofens.

Das vorletzte Kapitel dieser Arbeit beschäftigte sich daraufhin mit der Beschreibung der Implementierung und einer abschließenden Gesamtbewertung. Bezüglich der Implementierung sei nochmals hervorgehoben, dass das in dieser Arbeit entwickelte Programmerstellungskonzept bei den Dillinger Hüttenwerken vollständig umgesetzt wurde und sich im Laufe der Zeit auch bewährt hat. Sowohl die abschließend durchgeführte quantitative sowie die erläuterte qualitative Bewertung unterstreichen die mit den Entwicklungen der vorliegenden

Arbeit einhergehenden Verbesserungen gegenüber dem Alt-System.

Im Folgenden Abschnitt seien nun einige Ideen vorgestellt, wie und an welchen Punkten zukünftig noch weitere Entwicklungen angegangen werden könnten. Hinsichtlich des HTFSPD-HGA würde es sich sicher lohnen, genauer zu überprüfen, ob die Parametrierung weiteres Optimierungspotential bietet. Hierzu erscheint es sinnvoll, weitere Parameterkonstellationen zu testen. Auch eine je nach vorliegender Situation flexible Anpassung bestimmter Parameter wäre an dieser Stelle denkbar. Ein weiterer Punkt betrifft den *Online*-Aspekt. Hierin besteht das wohl noch größte Optimierungspotential, welches sich jedoch erst ausschöpfen lässt, wenn genauere Vorhersagen über Lagereintrittszeitpunkte zukünftig wärmezubehandelnder Waben vorliegen. An diesem Punkt sollten die Dillinger Hüttenwerke verstärkt ansetzen. Der letzte hier vorgestellte Ansatzpunkt für mögliche Weiterentwicklungen betrifft die anderen Wärmebehandlungsöfen der Dillinger Hüttenwerke. Insbesondere auf den bauähnlichen RHO2 ließe sich das in dieser Arbeit entwickelte Programmerstellungskonzept mit wenigen Anpassungen übertragen. Mit der Entwicklung entsprechender Materialflusssimulationen ließe sich der grundsätzliche Ansatz des HTFSPD-HGA aber auch in Verbindung mit den anderen Wärmebehandlungsöfen einsetzen. Da an einigen Öfen zum Teil die gleichen Wärmebehandlungen durchführbar sind, könnte man zudem auch schon eine Planungsebene höher ansetzen und die Vorabverteilung der Waben zu den Öfen optimieren.

Als Fazit lässt sich für die vorliegende Arbeit festhalten, dass mit dem hybriden Optimierungsansatz unter Verwendung einer speziellen Materialflusssimulation ein modernes und gesamtheitliches Lösungsverfahren für ein reales Maschinenbelegungsproblem entwickelt wurde. Es stellte sich heraus, dass weder diese noch eine vergleichbare Problemstellung in der Literatur bislang untersucht wurde. Die Effektivität des erarbeiteten Lösungsverfahrens im Vergleich zur ursprünglich im Betrieb vorherrschenden Verfahrensweise konnte nicht nur anhand numerischer Testergebnisse belegt, sondern aufgrund der Begleitung der kompletten Implementierung auch von betrieblicher Seite rückblickend untermauert werden.

Anhang

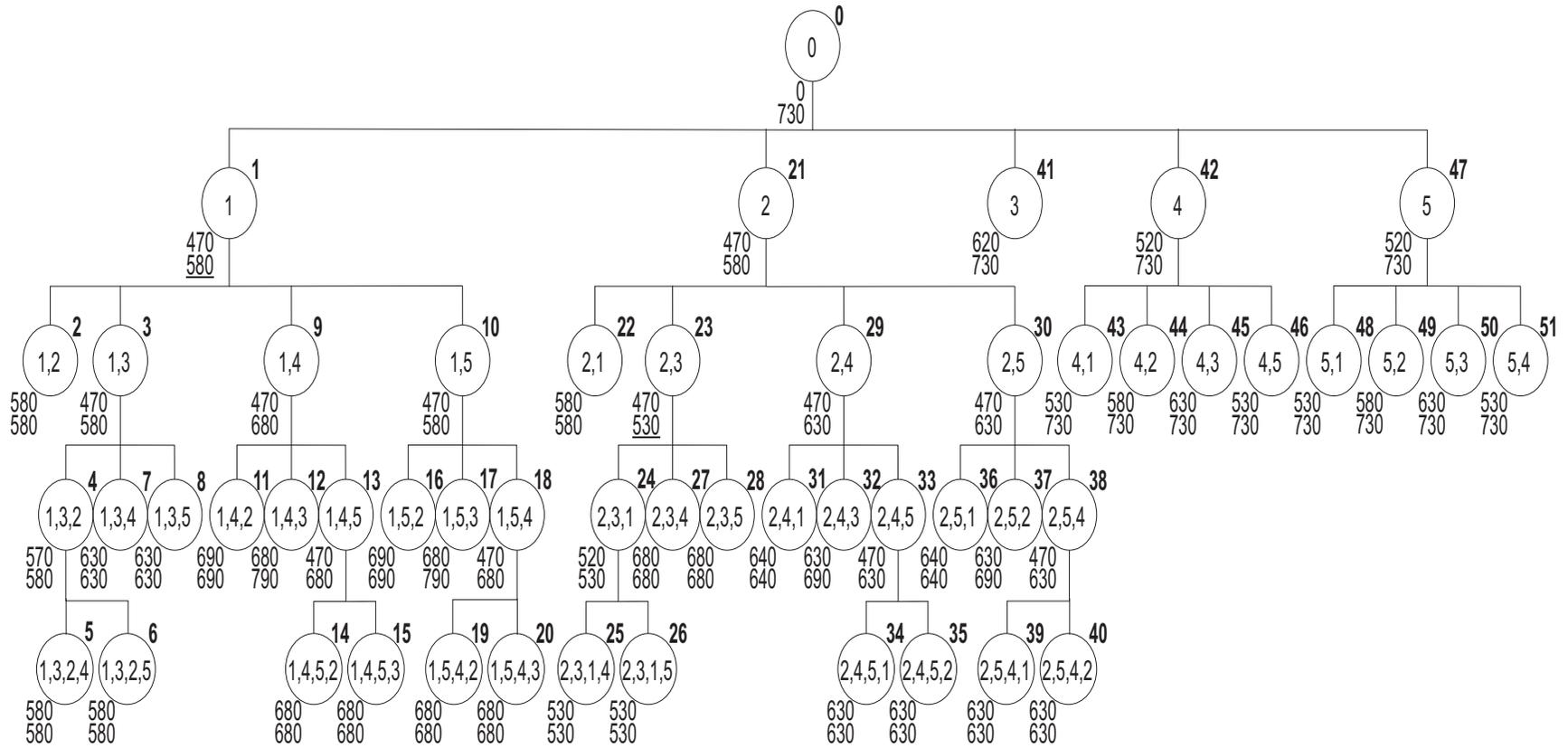


Abbildung A.1: Beispiel des *HTFSPD-Branch&Bound*-Verfahrens

| Alle | <= 14 | 15 | 16 | 17 | > 18 | 19 | 20 | 21 | >= 22 | Gesamt |
|-----------|-------|----|----|----|------|----|----|----|-------|--------|
| sonst. Or | 11 | 3 | 6 | 27 | 24 | 8 | 18 | 3 | 1 | 101 |
| Prio Pos | 1 | | | | | | | | | 1 |
| am RH03 | 2 | 1 | | 16 | 17 | 3 | 2 | 1 | | 42 |
| Prio Pos | | | | | | | | | | |
| Gesamt | 14 | 4 | 6 | 43 | 41 | 11 | 20 | 4 | 1 | 144 |

| Lage | Stoss | D min | D max | T min | T max | Prodwo | Anz | Ton |
|--------------|---------|-------|-------|-------|-------|---------|-----|-----|
| BR 12.9.4 | 3039617 | 52 | 120 | 305 | 540 | 213-218 | 10 | 193 |
| 525 | 3032047 | 70 | 132 | 540 | 540 | 213-221 | 5 | 47 |
| QTS 12.4 (1) | 3038680 | 100 | 100 | 540 | 540 | 219-219 | 1 | 10 |
| BR 12.9.3 | 3034224 | 100 | 120 | 540 | 540 | 218-219 | 4 | 60 |
| BR 12.10.3 | 3034224 | 120 | 120 | 540 | 540 | 218-220 | 3 | 33 |
| BR 12.9.4 | 3027090 | 140 | 160 | 540 | 540 | 217-218 | 7 | 153 |

| ProdWo | WerksNr | Po | Ref | Bles | Qua | Di | Br | Lae | Gew | FAB | 1FAB-nFAB | Temp | HD | OVZ | Lage | Stoss | LVU |
|--------|---------|----|--------|--------|-----|-------|------|-------|-------|------|-----------|------|----|-----|-----------|---------|-----|
| 201217 | 370876 | 31 | 712404 | 11 | E94 | 160,0 | 2062 | 5944 | 15394 | A1R3 | H5H2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 10 |
| 201217 | 370876 | 31 | 847150 | 11 | E94 | 160,0 | 2065 | 5960 | 15458 | A1R3 | H5H2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 9 |
| 201217 | 370876 | 31 | 872148 | 11 | E94 | 160,0 | 2065 | 5960 | 15458 | A1R3 | H5H2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 8 |
| 201217 | 370876 | 31 | 647151 | 11 | E94 | 160,0 | 2063 | 5950 | 15417 | A1R3 | H5H2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 7 |
| 201217 | 370876 | 31 | 012407 | 11 | E94 | 160,0 | 2064 | 5800 | 15036 | A1R3 | H5H2-LMR3 | 610 | 90 | 412 | BR 12.9.3 | 2896130 | 6 |
| 201215 | 373056 | 8 | 507581 | 11 | H71 | 80,0 | 2525 | 9050 | 14351 | KER3 | KAPU-LMR3 | 540 | 0 | 154 | BR 12.9.4 | 3039617 | 11 |
| 201216 | 371493 | 1 | 027658 | 11 2 3 | H62 | 26,0 | 1027 | 11565 | 2424 | KER3 | BLH2-LMR3 | 540 | 0 | 55 | BR 12.9.4 | 3039617 | 5 |
| 201216 | 371493 | 1 | 027658 | 11 2 3 | H62 | 26,0 | 1030 | 11565 | 2431 | KER3 | BLH2-LMR3 | 540 | 0 | 55 | BR 12.9.4 | 3039617 | 4 |

Anz: To: aus fallender Hitze Brennen Liste sortieren

| ProdWo | WerksNr | Po | Ref | Bles | Qua | Di | Br | Lae | Gew | FAB | 1FAB-nFAB | Temp | HD | OVZ | Lage | Stoss | LVU |
|--------|---------|----|--------|------|-----|-------|------|-------|-------|------|-----------|------|----|-----|------------|---------|-----|
| 201219 | 374186 | 4 | 633682 | 11 2 | Q77 | 200,0 | 2150 | 11845 | 39983 | VWR3 | OER3-VWR3 | 410 | 0 | 79 | RH03 | 0 | 0 |
| 201219 | 374186 | 1 | 116072 | 11 2 | Q77 | 180,0 | 2207 | 11137 | 34731 | VWR3 | OER3-VWR3 | 410 | 0 | 71 | RH03 | 0 | 0 |
| 201219 | 372296 | 3 | 342187 | 11 | E94 | 94,0 | 2238 | 9030 | 14912 | VWR3 | TWR3-OER3 | 540 | 0 | 156 | RH03-OE | 0 | 9 |
| 201216 | 371493 | 4 | 528687 | 11 2 | H62 | 112,0 | 2238 | 5784 | 11381 | VWR3 | OPC1-LMR3 | 395 | 0 | 43 | RH03-OE | 0 | 8 |
| 201215 | 372222 | 3 | 186271 | 11 | E85 | 82,0 | 1727 | 3626 | 4039 | VWR3 | OPC1-LMR3 | 540 | 0 | 63 | BR 12.9.3 | 3034224 | 11 |
| 201214 | 371650 | 5 | 785892 | 11 | E94 | 150,0 | 2037 | 6322 | 15164 | VWR3 | USKA-LMR3 | 540 | 0 | 119 | BR 12.10.4 | 3039617 | 10 |
| 201218 | 370876 | 25 | 810814 | 2 | E94 | 102,2 | 2051 | 5955 | 9799 | VWR3 | USRA-LMR3 | 540 | 0 | 98 | BR 12.9.3 | 3034224 | 10 |
| 201218 | 373911 | 13 | 358790 | 11 | E94 | 200,0 | 2211 | 7533 | 26149 | VWR3 | H5H2-LMR3 | 540 | 0 | 162 | BR 12.9.4 | 3025759 | 13 |
| 201218 | 373911 | 13 | 558789 | 11 | E94 | 200,0 | 2219 | 7507 | 26153 | VWR3 | H5H2-LMR3 | 540 | 0 | 162 | BR 12.9.4 | 3025759 | 12 |

Anz: To: Speichern Drucken Liste sortieren

Abbildung A.2: Programmierstellung des Alt-Systems

Vorgaben für die Optimierung:

Anzahl zur Verfügung stehender Brennanlagen
(für Brennen aus fallender Hitze):

Wechsel des WBH-Verfahrens:

- Anlassen
- Vorwärmen
- Kantenentspannen
- Effusionsglühen

Umstapler für den Kran bestrafen:

- schwach
- mittel
- hart

Simulieren

Speichern

Zurückstellungen:

| Ref | WB | Bles | ProdWo | Qua | Di | Br | Lae | Ge |
|--------|----|------|--------|-----|-------|------|------|-----|
| 594837 | 3 | 1 | 201220 | E94 | 130,0 | 2054 | 5900 | 123 |
| 456660 | 5 | 1 | 201213 | E94 | 132,0 | 2017 | 3900 | 81 |
| 739377 | 3 | 1 | 201221 | E94 | 90,0 | 2031 | 5950 | 85 |

Nächste Einfahrt: **81954** um: **2012-05-03 15:58:40** mit Lücke: **1500** mm Einzel-Fahrweise Aktualisierung in: **53** sek

| FNr | Ref | WB | Bles | ProdWo | Qua | Di | Br | Lae | Gew | FAB | nFAB | Temp | HD | OVZ | Lage | Stoss | LVO | BRFAB | BRTEMP | BRZ |
|-----|--------|----|------|--------|-----|-------|------|-------|-------|------|------|------|----|-----|-----------|---------|-----|-------|--------|-----|
| 1 | 093870 | 1 | 1 | 201217 | Q77 | 150,0 | 2423 | 5466 | 16350 | A1R3 | AKR3 | 620 | 5 | 299 | RH03-OA | 0 | 0 | BCH2 | 80 | 78 |
| 2 | 894079 | 1 | 1 | 201217 | Q77 | 150,0 | 2420 | 5626 | 16280 | A1R3 | A1R3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 79 |
| 3 | 893991 | 1 | 1 | 201217 | Q77 | 150,0 | 2437 | 5611 | 16030 | A1R3 | A1R3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 79 |
| 4 | 194073 | 1 | 1 | 201217 | Q77 | 150,0 | 2418 | 5714 | 16520 | A1R3 | A1R3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 80 |
| 5 | 594105 | 1 | 1 | 201217 | Q77 | 150,0 | 2470 | 5757 | 16760 | A1R3 | A1R3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 81 |
| 6 | 389106 | 1 | 1 | 201217 | Q77 | 150,0 | 2415 | 5684 | 16380 | A1R3 | A1R3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 80 |
| 7 | 294096 | 1 | 1 | 201217 | Q77 | 150,0 | 2418 | 5611 | 16190 | A1R3 | A1R3 | 620 | 5 | 299 | RH03 | 0 | 0 | BCH2 | 80 | 79 |
| 8 | 781954 | 1 | 1 | 201217 | Q77 | 150,0 | 2407 | 5669 | 16140 | A1R3 | OER3 | 620 | 5 | 299 | RH03-OE | 0 | 0 | BCH2 | 80 | 79 |
| 9 | 193262 | 1 | 1 | 201217 | Q77 | 150,0 | 2425 | 6000 | 16230 | A1R3 | LMR3 | 620 | 5 | 299 | BR 12.9.5 | 3060906 | 1 | BCH2 | 80 | 82 |
| 10 | 010454 | 3 | 2 | 201218 | E94 | 100,0 | 2065 | 5970 | 9678 | VWR3 | LMR3 | 475 | 0 | 102 | BR 12.9.3 | 3056365 | 3 | BNH2 | 300 | 66 |
| 11 | 111327 | 4 | 1 | 201218 | E94 | 120,0 | 2065 | 5567 | 10829 | VWR3 | LMR3 | 425 | 0 | 97 | BR 12.9.3 | 3056364 | 1 | BNH2 | 300 | 68 |
| 12 | 111327 | 5 | 2 | 201218 | E94 | 120,0 | 2065 | 5602 | 10897 | VWR3 | LMR3 | 425 | 0 | 97 | BR 12.9.3 | 3056364 | 2 | BNH2 | 300 | 68 |
| 13 | 972143 | 2 | 1 | 201218 | E94 | 120,0 | 2063 | 5950 | 11563 | VWR3 | LMR3 | 425 | 0 | 97 | BR 12.9.3 | 3056365 | 1 | BNH2 | 300 | 70 |
| 14 | 478161 | 4 | 3 | 201215 | E94 | 70,0 | 2050 | 5950 | 6703 | VWR3 | LMR3 | 550 | 0 | 119 | BR 12.5.2 | 3057600 | 1 | BFH2 | 300 | 58 |
| 15 | 832870 | 1 | 1 2 | 201218 | E95 | 90,0 | 2237 | 16702 | 26500 | VWR3 | LMR3 | 540 | 0 | 142 | BR 12.9.4 | 3060289 | 1 | BFH2 | 300 | 127 |
| 16 | 579484 | 1 | 1 2 | 201218 | E95 | 100,0 | 2228 | 15963 | 28500 | VWR3 | LMR3 | 540 | 0 | 158 | BR 12.9.4 | 3060289 | 2 | BFH2 | 300 | 128 |

Abbildung A.3: Die neue Programmierstellungsansicht (1)

Vorgaben für die Optimierung:

Anzahl zur Verfügung stehender Brennanlagen
(für Brennen aus fallender Hitze):

Wechsel des WBH-Verfahrens:

- Anlassen
- Vorwärmen
- Kantenentspannen
- Effusionsglühen

Umstapler für den Kran bestrafen:

- schwach
- mittel
- hart

Simulieren

Speichern

Zurückstellungen:

| Ref | WB | Bles | ProdWo | Qua | Di | Br | Lae | Ge |
|--------|----|------|--------|-----|-------|------|------|-----|
| 594837 | 3 | 111 | 201220 | E94 | 130,0 | 2054 | 5900 | 123 |
| 456660 | 5 | 111 | 201213 | E94 | 132,0 | 2017 | 3900 | 81 |
| 739377 | 3 | 111 | 201221 | E94 | 90,0 | 2031 | 5950 | 85 |

Nächste Einfahrt: **81954** um: **2012-05-03 15:58:40** mit Lücke: **1500** mm Einzel-Fahrweise Aktualisierung in: **41** sek

| Z | FAZ Einlauf | FAZ Ofen | FEZ Ofen | FEZ Auslauf | FAZ BP | FEZ BP | Warten | Luecke |
|---|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|--------|--------|
| | | | 2012-05-03 15:39:40 | 2012-05-03 16:22:40 | 2012-05-03 16:37:40 | 2012-05-03 17:55:40 | 0 | 0 |
| | | 2012-05-03 11:42:40 | 2012-05-03 16:51:40 | 2012-05-03 17:34:40 | 2012-05-03 17:49:40 | 2012-05-03 19:08:40 | -256 | 0 |
| | | 2012-05-03 12:19:40 | 2012-05-03 17:28:40 | 2012-05-03 18:11:40 | 2012-05-03 18:26:40 | 2012-05-03 19:45:40 | 37 | 1500 |
| | | 2012-05-03 13:06:40 | 2012-05-03 18:15:40 | 2012-05-03 18:58:40 | 2012-05-03 19:13:40 | 2012-05-03 20:33:40 | 47 | 1500 |
| | | 2012-05-03 13:47:40 | 2012-05-03 18:56:40 | 2012-05-03 19:39:40 | 2012-05-03 19:54:40 | 2012-05-03 21:15:40 | 41 | 1500 |
| | | 2012-05-03 14:37:40 | 2012-05-03 19:46:40 | 2012-05-03 20:29:40 | 2012-05-03 20:44:40 | 2012-05-03 22:04:40 | 50 | 1500 |
| | | 2012-05-03 15:20:40 | 2012-05-03 20:29:40 | 2012-05-03 21:12:40 | 2012-05-03 21:27:40 | 2012-05-03 22:46:40 | 43 | 1500 |
| | 2012-05-03 14:48:40 | 2012-05-03 15:58:40 | 2012-05-03 21:07:40 | 2012-05-03 21:50:40 | 2012-05-03 22:05:40 | 2012-05-03 23:24:40 | 38 | 1500 |
| | 2012-05-03 16:01:40 | 2012-05-03 16:39:40 | 2012-05-03 21:48:40 | 2012-05-03 22:31:40 | 2012-05-03 22:46:40 | 2012-05-04 00:08:40 | 41 | 1500 |
| | 2012-05-03 16:48:40 | 2012-05-03 21:51:40 | 2012-05-03 23:43:40 | 2012-05-03 23:53:40 | 2012-05-04 00:08:40 | 2012-05-04 01:14:40 | 312 | 13000 |
| | 2012-05-03 21:54:40 | 2012-05-03 22:05:40 | 2012-05-03 23:53:40 | 2012-05-04 00:03:40 | 2012-05-04 00:18:40 | 2012-05-04 01:26:40 | 14 | 7500 |
| | 2012-05-03 22:08:40 | 2012-05-03 23:01:40 | 2012-05-04 00:49:40 | 2012-05-04 00:59:40 | 2012-05-04 01:14:40 | 2012-05-04 02:22:40 | 56 | 7000 |
| | 2012-05-03 23:04:40 | 2012-05-03 23:13:40 | 2012-05-04 01:01:40 | 2012-05-04 01:11:40 | 2012-05-04 01:26:40 | 2012-05-04 02:36:40 | 12 | 7000 |
| | 2012-05-03 23:16:40 | 2012-05-03 23:35:40 | 2012-05-04 01:46:40 | 2012-05-04 02:07:40 | 2012-05-04 02:22:40 | 2012-05-04 03:20:40 | 22 | 7000 |
| 7 | 2012-05-03 23:38:40 | 2012-05-03 23:45:40 | 2012-05-04 02:19:40 | 2012-05-04 02:42:40 | 2012-05-04 02:57:40 | 2012-05-04 05:04:40 | 10 | 8000 |
| 8 | 2012-05-03 23:48:40 | 2012-05-03 23:55:40 | 2012-05-04 02:46:40 | 2012-05-04 03:09:40 | 2012-05-04 03:24:40 | 2012-05-04 05:32:40 | 10 | 7500 |

Abbildung A.4: Die neue Programmerstellungsansicht (2)

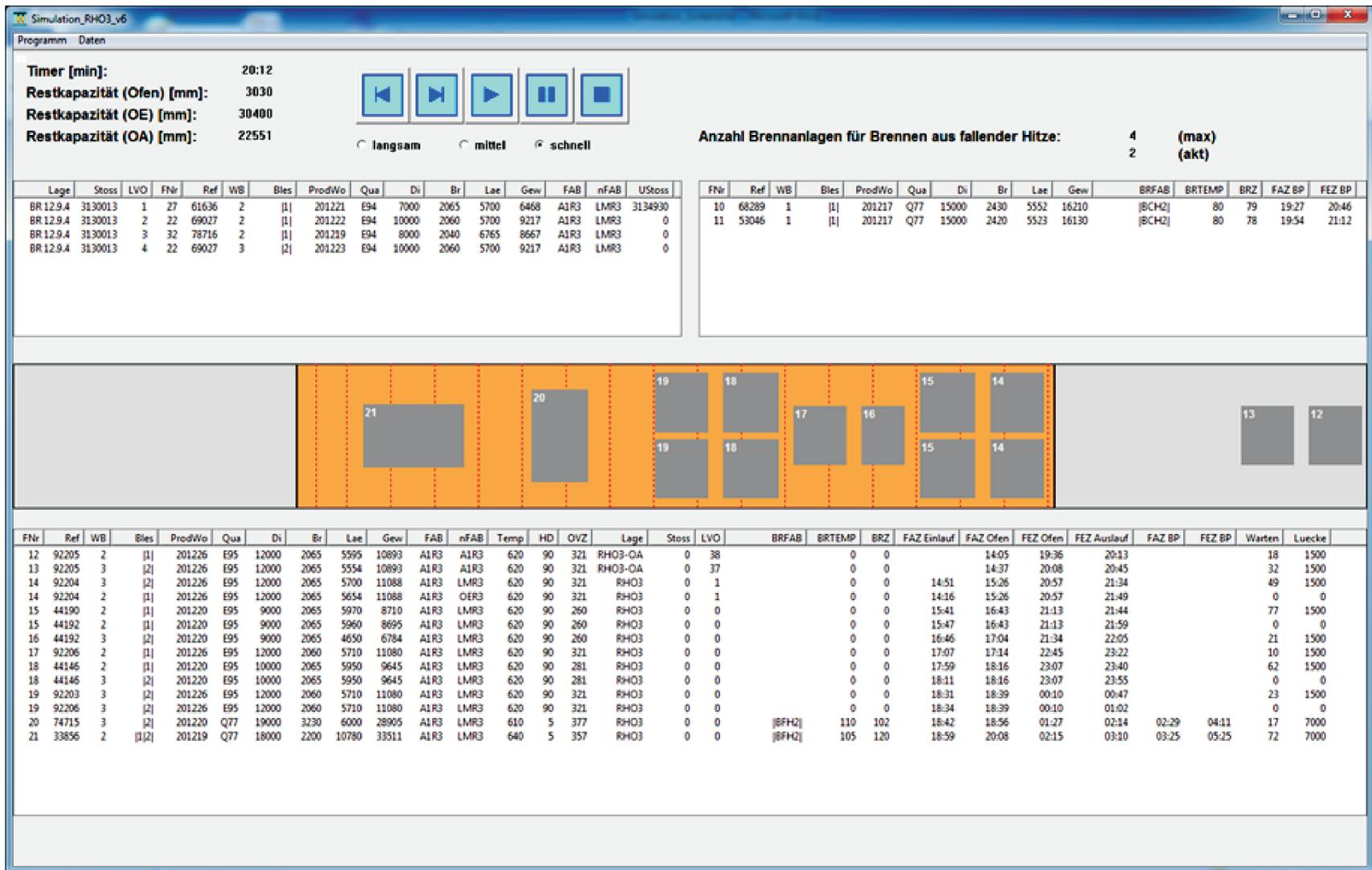


Abbildung A.5: Screenshot der Visualisierung

Abkürzungsverzeichnis

| | |
|-------------|--|
| A | Anlassen |
| ACC | <i>Accelerated Controlled Cooling</i> |
| BB | <i>Branch&Bound</i> |
| Bles | Blechnummern |
| Br | Breite |
| BRFAB | Brennschritte |
| BRP | <i>Blocks Relocation Problem</i> |
| BRTEMP | Brennendtemperaturen |
| BRZ | Brennzeit |
| DFPN | <i>Depth First Promise Next</i> |
| Di | Dicke |
| EBF | Ereignisbasierte Formulierung |
| EF | Effusionsglühen |
| EO | Einlegofen |
| FAB | Nächste Wärmebehandlung (Fertigungsabschnitt) |
| FAZ BP | Frühester Anfangszeitpunkt des Brennschneidens auf einem Brennportal |
| FAZ Einlauf | Frühester Ablagezeitpunkt auf dem Ofeneinlaufrollgang |
| FAZ Ofen | Frühester Anfangszeitpunkt der Wärmebehandlung im Ofen |
| FEZ Auslauf | Frühester Entnahmezeitpunkt vom Ofenauslaufrollgang |
| FEZ BP | Frühester Endzeitpunkt des Brennschneidens auf einem Brennportal |

| | |
|----------|--|
| FEZ Ofen | Frühester Endzeitpunkt der Wärmebehandlung im Ofen |
| FIFO | <i>First In First Out</i> |
| FNr | Folgenummer |
| GA | <i>Genetic Algorithm</i> |
| Gew | Gewicht |
| HD | Haltedauer |
| HF | <i>Hollomon-Fahrweise</i> |
| HGA | <i>Hybrid Genetic Algorithm</i> |
| HOM | <i>Hollomon Mutation Operator</i> |
| HTFSPD | <i>Heat Treatment Furnace Scheduling Problem with Distance Constraints</i> |
| HWO | Herdwagenofen |
| IDA | <i>Iterative Deepening Algorithm</i> |
| INSM | <i>Insert Mutation Operator</i> |
| INVM | <i>Inversion Mutation Operator</i> |
| KE | Kantenentspannen |
| Lae | Länge |
| Lage | Lagerbereich |
| LBE | <i>Lance-Bubbling-Equilibrium</i> |
| LD | Linz-Donawitz |
| lFAB | Zuletzt durchgeführter Fertigungsabschnitt |
| Luecke | Vorgeschlagene minimale Temperaturlücke zur Vorgängerwabe |
| LVO | Lage von oben im Stapel |
| LVU | Lage von unten im Stapel |
| MIP | <i>Mixed Integer Program</i> |
| N | Normalisieren |
| nFAB | Nächster offener Fertigungsabschnitt |
| NO | Normalisierofen |

| | |
|--------|--|
| OBM | Oxygen-Bottom-Maxhütte |
| OBX | <i>Order-based Crossover Operator</i> |
| OVZ | Ofenverweilzeit |
| PBX | <i>Position-based Crossover Operator</i> |
| PDR | <i>Promise Depth Relation</i> |
| PFDN | <i>Promise First Depth Next</i> |
| Po | Auftragspositionsnummer |
| PPS | Produktionsplanung und -steuerung |
| ProdWo | Sollproduktionswoche |
| Q | Quetten |
| Qua | Qualitätskennzeichnung |
| RBS | <i>Rule-based Sorting</i> |
| RCPSP | <i>Resource-Constrained Project Scheduling Problem</i> |
| Ref | Referenznummer |
| RHO | Rollenherdofen |
| ROGESA | Roheisengesellschaft Saar |
| SCRM | <i>Scramble Mutation Operator</i> |
| SSSP | <i>Slab Stack Shuffling Problem</i> |
| Stoss | Stoßnummer |
| SWM | <i>Swap Mutation Operator</i> |
| Temp | Wärmebehandlungstemperatur |
| TS | <i>Tabu Search</i> |
| UStoss | Umstapelstoßnummer |
| VB | <i>Visual Basic</i> |
| VF | Verband-Fahrweise |
| VW | Vorwärmen |
| Wabe | Wärmebehandlungseinheit |

| | |
|---------|--|
| Warten | Wartezeit ausgehend von der Ofeneinfahrt der Vorgängerwabe |
| WB | Wabenummer |
| WBH | Wärmebehandlung |
| WerksNr | Werksnummer |
| WZ | Wärmezone |
| ZDF | Zeitdiskrete Formulierung |
| ZKS | Zentralkokerei Saar |

Symbolverzeichnis

| | |
|---------------------------------|---|
| A | Fläche |
| $akd_j^K(tmp^{beg}, tmp^{end})$ | Abkühldauer des Kerns von Wabe j von tmp^{beg} auf tmp^{end} |
| $akd_j^O(tmp^{beg}, tmp^{end})$ | Abkühldauer der Oberfläche von Wabe j von tmp^{beg} auf tmp^{end} |
| α | Maschinencharakteristik |
| $AnzBR$ | Anzahl für das Brennen aus fallender Hitze zur Verfügung stehender Brennanlagen |
| | |
| B | <i>Batch</i> -Menge |
| b | Laufindex für <i>Batches</i> |
| b^A | Breite von Fläche A |
| b^{min} | Mindestabstand zwischen Waben innerhalb eines Verbands |
| b^{Ofen} | Ofenbreite |
| b_v^{vb} | Breite von Verband v |
| \bar{b} | Maximale Anzahl in einem <i>batch</i> enthaltener Aufträge |
| b_j | Breite von Wabe j |
| $bestlist$ | Qualitätsspeicher |
| $bestlist^{max}$ | Speicherkapazität des Qualitätsspeichers |
| β | Auftragscharakteristik |
| brd_j | Brenndauer von Wabe j |
| $brtemp_j$ | Brennendtemperatur von Wabe j |
| | |
| C_{max}^{akt} | Aktuell bester Wert für C^{max} |
| C_{max}^{glob} | Optimaler Wert für C^{max} |
| $C_{max,k}^{lb}$ | Untere Schranke für C^{max} bezüglich Knoten k |

| | |
|-------------------|---|
| C_{max}^{lb} | Untere Schranke für C^{max} |
| $C_{max,k}^{ub}$ | Obere Schranke für C^{max} bezüglich Knoten k |
| C_{max}^{ub} | Obere Schranke für C^{max} |
| $C_{max,k}^{lb*}$ | Untere Schranke für C^{max} bezüglich Knoten k (bei vorgeschalteter Temperaturanpassung) |
| c_e | Freie Ofenkapazität zum Ereignis e |
| C_j | Fertigstellungszeitpunkt von Auftrag j |
| c_t | Freie Ofenkapazität zum Zeitpunkt t |
| C_{max} | Gesamtfertigstellungszeitpunkt |
| C_{max}^V | Wert für C^{max} bei Anwendung der Verband- Fahrweise |
| d_j^{Ofen} | Abstand zwischen der Oberfläche von Wabe j und der Ofendecke |
| d_j | Dicke von Wabe j |
| dd_j | Fälligkeitstermin von Auftrag j |
| δ^b | Breitentoleranzparameter bei der Stapelbildung |
| δ^d | Dicken- bzw. Höhentoleranzparameter bei der Stapelbildung |
| δ^l | Längentoleranzparameter bei der Stapelbildung |
| δ^{b+} | Toleranzparametererhöhung für δ^b |
| δ^{d+} | Toleranzparametererhöhung für δ^d |
| δ^{l+} | Toleranzparametererhöhung für δ^l |
| E | Ereignisanzahl |
| E_j | Verfrühung von Auftrag j |
| ε | Parameter für den Sicherheitspuffer bei der Berechnung der Zieltemperaturen für Vorwärmer |
| ewd_j | Erwärmdauer von Wabe j |
| F | Anzahl Auftrag-Familien |
| f_{templ} | Temperaturlückenfaktor |
| $FAZB_j$ | Frühester Zeitpunkt, zu dem das Brennen von Wabe j beginnt |
| $FAZO_j$ | Frühestmöglicher Anfangszeitpunkt (Ofeneinfahrt) von Wabe j |
| $FEZB_j$ | Frühester Zeitpunkt, zu dem das Brennen von Wabe j endet |

| | |
|-------------------|--|
| $FEZO_j^{ARG}$ | Frühester Zeitpunkt, zu dem Wabe j den Auslaufrollgang verlässt |
| $FEZO_j$ | Frühestmöglicher Endzeitpunkt (Ofenausfahrt) von Wabe j |
| $folge$ | Bearbeitungsreihenfolge |
| $folge'$ | Fixierte Bearbeitungsreihenfolge |
| $folge^{akt}$ | Aktuell beste Bearbeitungsreihenfolge |
| $folge^{glob}$ | Optimale Bearbeitungsreihenfolge |
| $folge^{ind}$ | Bearbeitungsreihenfolge von Individuum ind |
| $fpos_j$ | Position von Wabe j in der Bearbeitungsreihenfolge |
| $frei_{s'}$ | Indikatorvariable für die Existenz eines freien Stapelplatzes s' |
| G | Hinreichend große Zahl |
| g | Laufindex für Auftrags-Familien |
| g' | Laufindex für Auftrags-Familien |
| $g_{jj'}$ | Indikatorvariable für die prinzipielle Zuordbarkeit der Waben j und j' im gleichen Verband |
| γ | Zielcharakteristik |
| H | Planungshorizont |
| hd^{start} | Startvorgabe bezüglich der Haltedauer |
| hd_j | Haltedauer von Wabe j |
| HJP_j | Hollomon-Jaffe-Parameter für Wabe j |
| i | Laufindex für Maschinen |
| $impit^{max,GA}$ | Maximale Anzahl an Iterationsschritten ohne Verbesserung im <i>HTFSPD-GA</i> |
| $impit^{max,HGA}$ | Maximale Anzahl an Iterationsschritten ohne Verbesserung im <i>HTFSPD-HGA</i> |
| $impit^{max,TS}$ | Maximale Anzahl an Iterationsschritten ohne Verbesserung im <i>HTFSPD-TS</i> |
| ind | Individuum |
| it | Laufindex für Iterationen |
| J | Auftragsmenge |
| j | Laufindex für Aufträge |

| | |
|-------------------|--|
| J' | Pseudowabenmenge |
| j' | Laufindex für Aufträge |
| j^* | Wabe, deren Brennvorgang als nächstes endet |
| \tilde{j} | Fiktive, beliebig teilbare Wabe |
| K | Knoten- bzw. Kandidatenliste |
| k | Laufindex für Knoten bzw. Teilprobleme |
| kb_j | Kapazitätsbedarf von Auftrag j |
| kvp^{hk} | Konvektionsparameter für einen Temperatursprung von heiß nach kalt |
| kvp^{kh} | Konvektionsparameter für einen Temperatursprung von kalt nach heiß |
| l^A | Länge von Fläche A |
| l^V | Verbandlängensumme |
| l^{ARG} | Länge des Ofenauslaufrollgangs |
| l_{rest}^{ARG} | Aktuelle Restkapazität auf dem Ofenauslaufrollgang |
| l^{Ofen} | Ofenlänge |
| l_{rest}^{Ofen} | Aktuelle Restkapazität des Ofens |
| l_v^{vb} | Länge von Verband v |
| L_j | Terminabweichung von Auftrag j |
| l_j | Länge von Wabe j |
| L_{max} | Maximale Terminabweichung |
| $lager^{x_1}$ | x_1 -Koordinate des Lagerbereichs |
| $lager^{x_2}$ | x_2 -Koordinate des Lagerbereichs |
| $lager^{y_1}$ | y_1 -Koordinate des Lagerbereichs |
| $lager^{y_2}$ | y_2 -Koordinate des Lagerbereichs |
| lwo_j | Lage, welche Wabe j von oben im Stapel einnimmt |
| M | Maschinenmenge |
| m | Maschinenanzahl |
| MK | Maschinenkapazität |
| n | Auftragsanzahl bzw. Wabenanzahl |
| n' | Anzahl fixierter Waben |
| n^* | Wabenanzahl (inklusive Pseudowaben) |

| | |
|-------------------------|--|
| n^V | Verbandanzahl |
| n^{V*} | Anzahl Verbände mit mindestens zwei enthaltenen Waben |
| nb | Nachbarschaftsmenge |
| $nbsuch^*$ | Parameter für die Bestimmung von $tsize^{nbsuch}$ |
| $nbwahl^*$ | Parameter für die Bestimmung von $tsize^{nbwahl}$ |
| NP | Komplexitätsklasse |
| $open_{k,j}$ | Indikatorvariable für die Verzweigung von Knoten k in Richtung Wabe j |
| ovz_{max}^{unvp} | Maximale Ofenverweilzeit der unverplanten Waben |
| ovz_j | Ofenverweilzeit von Wabe j |
| ovz_{jz} | Ofenverweilzeit von Wabe j bezüglich Temperatur-Zeit-Kombination z |
| P | Komplexitätsklasse |
| p | Laufindex für Reihenfolgepositionen |
| p^b | Bearbeitungsdauer von <i>Batch</i> b |
| p_j | Bearbeitungsdauer von Auftrag j |
| pdr | Wert für die Knotenauswahl in Rahmen der (BB-PDBR)-Strategie |
| φ | Einstrahlzahl |
| $\varphi_{j,WZ_{j'}}^F$ | Einstrahlzahl zwischen dem Fuß von Wabe j und der Wärmezone von Wabe j' |
| $\varphi_{j,WZ_{j'}}^K$ | Einstrahlzahl zwischen dem Kopf von Wabe j und der Wärmezone von Wabe j' |
| Pop | Population |
| Pop^{it} | Population bei Iteration it (it -te Generation) |
| q | Laufindex für Reihenfolgepositionen |
| \dot{Q} | Wärmestrom |
| $\dot{Q}_{j,WZ_{j'}}^F$ | Wärmestrom, der von der Wärmezone von Wabe j' auf den Fuß von Wabe j strömt |
| $\dot{Q}_{j,WZ_{j'}}^K$ | Wärmestrom, der von der Wärmezone von Wabe j' auf den Kopf von Wabe j strömt |
| \dot{Q}_{krit}^K | Toleranzparameter für die bezüglich des Wabenkopfs einzuhaltenden Wärmestromverhältnisse |

| | |
|-----------------------|--|
| \dot{Q}_{krit}^{KF} | Toleranzparameter für die bezüglich der Kopf-Fußabweichung einzuhaltenden Wärmestromverhältnisse |
| r_j | Bereitstellungszeitpunkt von Auftrag j |
| $restart^{best}$ | Parameter für den Neustart des <i>HTFSPD-TS</i> mit einer Lösung aus dem aktuellen Qualitätsspeicher |
| S | Stapelmenge |
| s | Stapel |
| s^d | Aktuelle Stapelhöhe von Stapel s |
| s^{bmin} | Breite der schmalsten in Stapel s enthaltenen Wabe |
| s^{EZiel} | Stapel, welcher die nächste zu entnehmende Wabe enthält |
| s^{fmin+} | Hilfsvariable zur Ermittlung von s^{UZiel} |
| s^{fmin-} | Hilfsvariable zur Ermittlung von s^{UZiel} |
| s^{fmin} | Minimale Reihenfolgeposition der in Stapel s enthaltenen Waben |
| s^{lmin} | Länge der kürzesten in Stapel s enthaltenen Wabe |
| s^{UZiel} | Zielstapel für eine umzustapelnde Wabe |
| s^{x1} | x_1 -Koordinate von Stapel s |
| s^{x2} | x_2 -Koordinate von Stapel s |
| s^{y1} | y_1 -Koordinate von Stapel s |
| s^{y2} | y_2 -Koordinate von Stapel s |
| \bar{s}^{EZiel} | Oberste Wabe in Stapel s^{EZiel} |
| σ | Stefan-Boltzmann-Konstante |
| $sort_{ab}$ | Indikatorvariable für die absteigende Sortierung nach den Ofentemperaturen |
| $sort_{auf}$ | Indikatorvariable für die aufsteigende Sortierung nach den Ofentemperaturen |
| $st_{gg'}$ | Reihenfolgeabhängige Rüstzeit zwischen Auftrags-Familie g und Auftrags-Familie g' |
| $st_{jj'}$ | Reihenfolgeabhängige Rüstzeit zwischen Auftrag j und Auftrag j' |
| t | Laufindex für die Zeit |
| t^u | Dauer einer Transport- bzw. Umstapelaktion |
| T^{max} | Maximale Länge des Planungshorizonts |

| | |
|----------------------|--|
| t^{min} | Mindestwartezeit zur Vorgängerwabe bzw. Einfahrdauer |
| t^{trans} | Voraussichtliche (Lager-)Entnahmezeit für die nächste an den Ofen zu transportierende Wabe |
| t_e | Eintrittszeitpunkt von Ereignis e |
| T_j | Verspätung von Auftrag j |
| TA | Temperatur einer Fläche |
| $tabulist$ | Tabu-Liste |
| $tabulist^{max}$ | Speicherkapazität der Tabu-Liste |
| \bar{T} | Menge potentieller Starttemperaturen |
| $templ^v$ | Länge der künstlichen Verschiebung des Temperaturwechsels |
| $templ^{min}$ | Mindestabstand zwischen zwei Waben im Ofen |
| $templ_{max}^{unvp}$ | Maximale Ofentemperaturabweichung zwischen der letzten Wabe aus $folge'$ und einer Wabe aus $unvp$ |
| $templ_j$ | Temperaturlücke zwischen Wabe j und ihrer Vorgängerwabe |
| $temptol_j^+$ | Obere Temperaturtoleranz von Wabe j |
| $temptol_j^-$ | Untere Temperaturtoleranz von Wabe j |
| $time^{GA}$ | Statisches Zeitlimit im <i>HTFSPD-GA</i> |
| $time^{HGA}$ | Statisches Zeitlimit im <i>HTFSPD-HGA</i> |
| $time^{TS}$ | Statisches Zeitlimit im <i>HTFSPD-TS</i> |
| tmp^{beg} | Betrachtete Starttemperatur |
| tmp^{end} | Betrachtete Endtemperatur |
| $tmpd$ | Hilfsvariable bei der Berechnung von Temperaturlücken |
| $tsize^{GA}$ | Parameter für die Turniergröße (für die Selektion im <i>HTFSPD-GA</i>) |
| $tsize^{nbsuch}$ | Parameter für die Turniergröße (für die Auswahl der Reihenfolgeposition im Rahmen der Nachbarschaftssuche des <i>HTFSPD-TS</i>) |
| $tsize^{nbwahl}$ | Parameter für die Turniergröße (für die Auswahl einer Lösung im Rahmen der Nachbarschaftssuche des <i>HTFSPD-TS</i>) |
| $tsize^{TS}$ | Parameter für die Turniergröße (für die Selektion im <i>HTFSPD-TS</i>) |
| $turnus^{TS}$ | Iterationsturnus für den Aufruf des <i>HTFSPD-TS</i> |
| u | Laufindex für Teilstücke der fiktiven Waben \tilde{j} |

| | |
|--------------------------|--|
| U_j | Termineinhaltungsstatus von Auftrag j |
| $unvp$ | Menge unverplanter Waben |
| V | Verbandmenge |
| v | Laufindex für Verbände |
| v^p | Produktionsgeschwindigkeit |
| vb_v | Menge, der Verband v zugeordneten Waben |
| VNB | Menge der Vorgänger-Nachfolger-Beziehungen |
| w | Laufindex für Wabengruppen |
| W^{ARG} | Geordnete Menge der aktuell auf dem Ofenauslauf- rollgang befindlichen Waben |
| W^{BR} | Geordnete Menge der aktuell in den Brennanlagen befindlichen Waben |
| W^{Ofen} | Geordnete Menge der aktuell im Ofen befindlichen Waben |
| w_j | Prioritätsfaktor von Auftrag j |
| $warte_j$ | Wartezeit von Wabe j |
| wbh_j | Wärmebehandlungsverfahren von Wabe j |
| $wbht_{max}^{Ofen,unvp}$ | Maximale Ofentemperatur der unverplanten Waben |
| $wbht_{min}^{Ofen,unvp}$ | Minimale Ofentemperatur der unverplanten Waben |
| $wbht^{Ofen^{VW1}}$ | Ofentemperatur für Vorwärmer in der ersten Erwärmphase |
| $wbht^{Ofen^{VW2}}$ | Ofentemperatur für Vorwärmer in der zweiten Erwärmphase |
| $wbht_j^{Ofen}$ | Ofentemperatur von Wabe j |
| $wbht_{jz}^{Ofen}$ | Ofentemperatur von Wabe j bezüglich Temperatur-Zeit-Kombination z |
| $wbht_{min}^{Ofen}$ | Minimale Ofentemperatur der betrachteten Waben |
| $wbht^{Ziel,start}$ | Startvorgabe bezüglich der Ziel- bzw. Haltetemperatur |
| $wbht_j^{Ziel}$ | Zieltemperatur von Wabe j |
| $wbht_{jz}^{Ziel}$ | Ziel- bzw. Haltetemperatur von Wabe j bezüglich Temperatur-Zeit-Kombination z |
| $wbht_j$ | Wärmebehandlungstemperatur von Wabe j |
| WG | Wabengruppenmenge |
| wg_w | Menge, der Wabengruppe w zugeordneten Waben |
| wk^{HOM} | Eintrittswahrscheinlichkeit für HOM |

| | |
|-------------|---|
| wk^{INSM} | Eintrittswahrscheinlichkeit für INSM |
| wk^{INVM} | Eintrittswahrscheinlichkeit für INVM |
| wk^{NOM} | Eintrittswahrscheinlichkeit für das Entfallen der Mutation |
| wk^{NOX} | Eintrittswahrscheinlichkeit für das Entfallen der Rekombination |
| wk^{OBX} | Eintrittswahrscheinlichkeit für OBX |
| wk^{PBX} | Eintrittswahrscheinlichkeit für PBX |
| wk^{SCRM} | Eintrittswahrscheinlichkeit für SCRM |
| wk^{SWM} | Eintrittswahrscheinlichkeit für SWM |
| wl_j | Wartezeit aufgrund der Länge für Wabe j |
| x^{min} | Längsseitiger Mindestabstand zwischen zwei Stapeln |
| $x_{1,j}$ | x_1 -Koordinate von Wabe j |
| $x_{2,j}$ | x_2 -Koordinate von Wabe j |
| x_{jb} | Binäre Entscheidungsvariable für die Zuordnung von Auftrag j zu <i>Batch</i> b |
| $x_{jj'e}$ | Binäre Entscheidungsvariable für die Ofeneinfahrt von Wabe j zum Ereignis e mit Wabe j' als direktem Vorgänger |
| $x_{jj't}$ | Binäre Entscheidungsvariable für die Ofeneinfahrt von Wabe j zum Zeitpunkt t mit Wabe j' als direktem Vorgänger |
| x_{jt} | Binäre Entscheidungsvariable für den Start von Auftrag j zum Zeitpunkt t |
| x_{jv} | Binäre Entscheidungsvariable für die Zuordnung von Wabe j in Verband v |
| y^{min} | Breitseitiger Mindestabstand zwischen zwei Stapeln |
| y_b | Binäre Entscheidungsvariable für die Verwendung von <i>Batch</i> b |
| $y_{1,j}$ | y_1 -Koordinate von Wabe j |
| $y_{2,j}$ | y_2 -Koordinate von Wabe j |
| $y_{jj'e}$ | Binäre Entscheidungsvariable für die Ofenausfahrt von Wabe j zum Ereignis e mit Wabe j' als direktem Nachfolger |

| | |
|--------------|--|
| $y_{jj't}$ | Binäre Entscheidungsvariable für die Ofenausfahrt von Wabe j zum Zeitpunkt t mit Wabe j' als direktem Nachfolger |
| z_j^{glob} | Optimale Temperatur-Zeit-Kombination für Wabe j |
| z^{max} | Maximale Anzahl betrachteter Temperatur-Zeit-Kombinationen |
| zul_s | Indikatorvariable für die Zulässigkeit von Stapel s |

Literaturverzeichnis

- [1] ARNOLD, D. ; ISERMANN, H. ; KUHN, A. ; TEMPELMEIER, H. ; FURMANS, K.: *Handbuch Logistik*. Springer, Berlin, Heidelberg, 2008
- [2] ARTIGUES, C. ; DEMASSEY, S. ; NÉRON, E.: *Resource-constrained project scheduling*. John Wiley & Sons, Hoboken, New Jersey, 2008
- [3] ATIGHEHCHIAN, A. ; BIJARIA, M. ; TARKESHA, H.: A novel hybrid algorithm for scheduling steel-making continuous casting production. In: *Computers & Operations Research* 36 (2009), Nr. 8, S. 2450–2461
- [4] BAKER, K.R. ; TRIETSCH, D.: *Principles of sequencing and scheduling*. John Wiley & Sons, Hoboken, New Jersey, 2009
- [5] BARGEL, H.J. ; HILBRANS, H. ; SCHULZE, G.: *Werkstoffkunde*. Springer, Berlin, Heidelberg, 2008
- [6] BELLABDAOUI, A. ; TEGHEM, J.: A mixed-integer linear programming model for the continuous casting planning. In: *International Journal of Production Economics* 104 (2006), Nr. 2, S. 260–270
- [7] BELLMAN, R.: *Dynamic programming*. Princeton University Press, Princeton, New Jersey, 1957
- [8] BERTSEKAS, D.P.: *Dynamic programming and optimal control*. Athena Scientific, Belmont, 2007
- [9] BLAZEWICZ, J. ; ECKER, K. ; PESCH, E. ; SCHMIDT, G. ; WEGLARZ, J.: *Handbook on scheduling: From theory to applications*. Springer, Berlin, Heidelberg, 2007

- [10] BLAZEWICZ, J. ; LENSTRA, J.K. ; RINNOOY KAN, A.H.G.: Scheduling subject to resource constraints: Classification and complexity. In: *Discrete Applied Mathematics* 5 (1983), Nr. 1, S. 11–24
- [11] BOLBRINKER, A.K. ; FEURER, M. ; GARBRACHT, K. ; SCHÜRMAN, H.: *Stahl-Lexikon*. GRIPS media, Bad Harzburg, 2004
- [12] BORODIN, A. ; EL-YANIV, R.: *Online computation and competitive analysis*. Cambridge University Press, Cambridge, 2005
- [13] BRUCKER, P.: *Scheduling algorithms*. Springer, Berlin, Heidelberg, 2007
- [14] BUSCH, H.: *Stochastische Optimierungsverfahren zur Produktionssteuerung von Walzstraßen*, Technische Universität Clausthal, Diss., 2005
- [15] CASERTA, M. ; SCHWARZE, S. ; VOSS, S.: *The blocks relocation problem*. <http://iwi.econ.uni-hamburg.de/IWIWeb/Uploads/Team/augsburg0908.pdf>, . – Zuletzt aufgerufen am 21.09.2012
- [16] CASERTA, M. ; SCHWARZE, S. ; VOSS, S.: A new binary description of the blocks relocation problem and benefits in a look ahead heuristic. In: *Proceedings of the EvoCOP*, 2009, S. 37–48
- [17] CASERTA, M. ; VOSS, S.: A cooperative strategy for guiding the corridor method. In: *Studies in computational intelligence* 236 (2009), S. 273–286
- [18] CASERTA, M. ; VOSS, S.: Corridor selection and fine tuning for the corridor method. In: *Proceedings of the 3rd International Conference on Learning and Intelligent Optimization*, 2009, S. 163–175
- [19] CASERTA, M. ; VOSS, S. ; SNIEDOVICH, M.: Applying the corridor method to a blocks relocation problem. In: *OR Spectrum* 33 (2011), Nr. 4, S. 915–929
- [20] CHEN, D.-S. ; BATSON, R.G. ; DANG, Y.: *Applied integer programming: Modeling and solution*. John Wiley & Sons, Hoboken, New Jersey, 2010
- [21] CHENG, X. ; TANG, L.: A scatter search algorithm for the slab stack shuffling problem. In: *ICSI (1)*, 2010, S. 382–389

- [22] CLARKE, G. ; WRIGHT, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. In: *Operations Research* 12 (1964), S. 568–581
- [23] COWLING, P.: A flexible decision support system for steel hot rolling mill scheduling. In: *Computers & Industrial Engineering* 45 (2003), Nr. 2, S. 307–321
- [24] DAMODARAN, P. ; MANJESHWAR, P.K. ; SRIHARI, K.: Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms. In: *International Journal of Production Economics* 103 (2006), Nr. 2, S. 882–891
- [25] DANG, C. ; KANG, L.: Batch-processing scheduling with setup times. In: *Journal of Combinatorial Optimization* 8 (2004), Nr. 2, S. 137–146
- [26] DE CASTILHO, B. ; DAGANZO, C.F.: Handling strategies for import containers at marine terminals. In: *Transportation Research* 27B (1993), Nr. 2, S. 151–166
- [27] DIN EN 10001: *Begriffsabstimmung und Einteilung von Roheisen, Deutsche Fassung EN 10001:1990*, 1991
- [28] DIN EN 10020: *Begriffsabstimmungen für die Einteilung der Stähle, Deutsche Fassung EN 10020:2000*, 2000
- [29] DIN EN 10029: *Warmgewalztes Stahlblech von 3 mm Dicke an - Grenzabmaße und Formtoleranzen, Deutsche Fassung EN 10029:2010*, 2011
- [30] DIN EN 10052: *Begriffe der Wärmebehandlung von Eisenwerkstoffen, Deutsche Fassung EN 10052:1993*, 1994
- [31] DOBSON, G. ; NAMBIMADOM, R.S.: The batch loading and scheduling problem. In: *Operations Research* 49 (2001), Nr. 1, S. 52–65
- [32] DOMSCHKE, W. ; DREXL, A.: *Einführung in Operations Research*. Springer, Berlin, Heidelberg, 2004
- [33] EBERMANN, E. ; NICKEL, S.: Scheduling steel plates on a roller furnace. In: *Operations Research Proceedings 2011: Selected Papers of the International Conference on Operations Research (OR 2011)*, 2011, S. 389–394

- [34] EIBEN, A. ; SMITH, J.: *Introduction to evolutionary computing*. Springer, Berlin, Heidelberg, 2003
- [35] ELEY, M.: *Simulation in der Logistik - Eine Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges Plant Simulation*. Springer, Berlin, Heidelberg, 2012
- [36] FORSTER, F. ; BORTFELDT, A.: A tree search procedure for the container relocation problem. In: *Computers & Operations Research* 39 (2012), Nr. 2, S. 299–309
- [37] GAREY, M.R. ; JOHNSON, D.S.: *Computers and intractability - A guide to the theory of NP-completeness*. W.H. Freeman and Company, San Francisco, 1979
- [38] GENDREAU, M.: An introduction to tabu search. In: GLOVER, F. (Hrsg.) ; KOCHENBERGER, G.A. (Hrsg.): *Handbook of metaheuristics*. Kluwer Academic Publishers, Boston, 2003, S. 37–54
- [39] GLOVER, F.: Future paths for integer programming and links to artificial intelligence. In: *Computers & Operations Research* 13 (1986), Nr. 5, S. 533–549
- [40] GLOVER, F. ; LAGUNA, M.: *Tabu search*. Kluwer Academic Publishers, Boston, 1997
- [41] GOBRECHT, J. ; RUMPLER, E.: *Werkstofftechnik - Metalle*. Oldenbourg, München, 2006
- [42] GOLDBERG, D.E.: *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, Massachusetts, 1989
- [43] GRAHAM, E.L. ; LAWLER, J.K. ; LENSTRA, J.K. ; RINNOOY KAN, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: A survey. In: *Annals of Discrete Mathematics* 5 (1979), S. 287–326
- [44] HANSEN, J.: *Industrialised application of combinatorial optimization*, Technical University of Denmark, Diss., 2003

- [45] HARTMANN, S. ; BRISKORN, D.: A survey of variants and extensions of the resource-constrained project scheduling problem. In: *European Journal of Operational Research* 207 (2010), Nr. 1, S. 1–14
- [46] HIERSIG, H.M.: *Lexikon Maschinenbau*. Springer, Berlin, Heidelberg, 1995
- [47] HOLLAND, J.H.: Outline for a logical theory of adaptive systems. In: *Journal of the ACM* 3 (1962), S. 297–314
- [48] HOLLAND, J.H.: *Adaption in natural and artificial systems*. University of Michigan Press, Ann Arbor, 1975
- [49] HOLLOMON, J.H. ; JAFFE, L.D.: Time-temperature relations in tempering of steel. In: *AIME* 162 (1945), S. 223–249
- [50] IBM: *IBM ILOG CPLEX CP Optimizer*. <http://www.ibm.com/software/integration/optimization/cplex-cp-optimizer/>, . – Zuletzt aufgerufen am 21.09.2012
- [51] IBM: *IBM ILOG CPLEX Optimizer*. <http://www.ibm.com/software/integration/optimization/cplex-optimizer/>, . – Zuletzt aufgerufen am 21.09.2012
- [52] KARABOGA, D. ; AKAY, B.: A survey: algorithms simulating bee swarm intelligence. In: *Artificial Intelligence Review* 31 (2009), S. 61–85
- [53] KARCI, A.: Imitation of bee reproduction as a crossover operator in genetic algorithms. In: *Lecture Notes in Computer Science* 3157 (2004), S. 1015–1016
- [54] KELLEGÖZ, T. ; TOKLU, B. ; WILSON, J.: Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem. In: *Applied Mathematics and Computation* 199 (2008), Nr. 2, S. 590–598
- [55] KIM, K.H.: Evaluation of the number of rehandles in container yards. In: *Computers & Industrial Engineering* 32 (1997), Nr. 4, S. 701–711
- [56] KIM, K.H. ; HONG, G.-P.: A heuristic rule for relocating blocks. In: *Computers & Operations Research* 323 (2006), S. 940–954

- [57] KÖNIG, F.G.: *Sorting with objectives - Graph theoretic concepts in industrial optimization*, Technische Universität Berlin, Diss., 2009
- [58] KONÉ, O. ; ARTIGUES, C. ; LOPEZ, P. ; MONGEAU, M.: Event-based MILP models for resource-constrained project scheduling problems. In: *Computers & Operations Research* 38 (2011), Nr. 1, S. 3–13
- [59] LAW, A.M. ; KELTON, W.D.: *Simulation modeling and analysis*. McGraw-Hill, New York, 2000
- [60] LÖDDING, H.: *Verfahren der Fertigungssteuerung*. Springer, Berlin, Heidelberg, 2008
- [61] LIEDTKE, D. ; KESSLER, O. ; SOMMER, P.: *Wärmebehandlung von Eisenwerkstoffen I*. Expert Verlag, Renningen, 2010
- [62] LÄPPLE, V.: *Wärmebehandlung des Stahls*. Europa-Lehrmittel, Haan-Gruiten, 2006
- [63] LUEGER, O.: *Lexikon der gesamten Technik und ihrer Hilfswissenschaften*. DVA Stuttgart, Leipzig, 1920
- [64] MANJESHWAR, P.K. ; DAMODARAN, P. ; SRIHARI, K.: Minimizing makespan in a flow shop with batch-processing machines using simulated annealing. In: *Robotics and Computer-Integrated Manufacturing* 25 (2009), Nr. 3, S. 667–679
- [65] MELOUK, S. ; DAMODARAN, P. ; CHANG, P.-Y.: Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. In: *International Journal of Production Economics* 87 (2004), Nr. 2, S. 141–147
- [66] MÄRZ, L. ; KRUG, W. ; ROSE, O. ; WEIGERT, G.: *Simulation und Optimierung in Produktion und Logistik*. Springer, Berlin, Heidelberg, 2011
- [67] NEUMANN, K. ; SCHWINDT, C. ; ZIMMERMANN, J.: *Project scheduling with time windows and scarce resources*. Springer, Berlin, Heidelberg, 2003

- [68] N.N.: Dillinger Hütte investiert. In: *Frankfurter Allgemeine Zeitung* 123 (2011), S. 18
- [69] PACCIARELLI, D. ; PRANZO, M.: Production scheduling in a steelmaking-continuous casting plant. In: *Computers & Chemical Engineering* 28 (2004), Nr. 12, S. 2823–2835
- [70] PAN, C. ; YANG, G.K.: A method of solving a large-scale rolling batch scheduling problem in steel production using a variant of column generation. In: *Computers & Industrial Engineering* 56 (2009), Nr. 1, S. 165–178
- [71] PAPADIMITRIOU, C.H.: *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994
- [72] PINEDO, M.L.: *Scheduling - Theory, algorithms, and systems*. Springer, New York, 2008
- [73] POTTS, C.N. ; KOVALYOV, M.Y.: Scheduling with batching: A review. In: *European Journal of Operational Research* 120 (2000), S. 228–249
- [74] PRITSKER, A.A.B. ; WAITERS, L.J. ; WOLFE, P.M.: Multiproject scheduling with limited resources: A zero-one programming approach. In: *Management Science* 16 (1969), Nr. 1, S. 93–107
- [75] ROSSI, F. ; VAN BEEK, P. ; WALSH, T.: *Handbook of constraint programming*. Elsevier, Amsterdam, 2006
- [76] SCHUH, G.: *Produktionsplanung und -steuerung*. Springer, Berlin, Heidelberg, 2006
- [77] SILVER, E.A. ; MEAL, H.C.: A heuristic for selecting lot size quantities for the case of a deterministic varying demand rate and discrete opportunities for replenishment. In: *Production and Inventory Management* 14 (1973), Nr. 2, S. 64–74
- [78] SIVANANDAM, S.N. ; DEEPA, S.N.: *Introduction to genetic algorithms*. Springer, Berlin, Heidelberg, New York, 2008

- [79] SRINIVAS, K.A. S. ; TIWARI, M.K.: Modelling the slab stack shuffling problem in developing steel rolling schedules and its solution using improved parallel genetic algorithms. In: *International Journal of Production Economics* 91 (2003), S. 135–147
- [80] STAHL-INSTITUT VDEH: *Stahlfibel*. Stahleisen GmbH, Düsseldorf, 2007
- [81] STAHL-ZENTRUM: *Die größten Stahlerzeuger in Deutschland*. http://www.stahl-online.de/deutsch/Linke_Navigation/Stahl_in_Zahlen/_Dokumente/Die_groessten_Stahlerzeuger_in_Deutschland_2011.pdf, . – Zuletzt aufgerufen am 21.09.2012
- [82] STAHL-ZENTRUM: *Welt-Rohstahlproduktion nach Regionen*. http://www.stahl-online.de/deutsch/Linke_Navigation/Stahl_in_Zahlen/_Dokumente/Welt_Rohstahlproduktion_nach_Regionen_2011.pdf, . – Zuletzt aufgerufen am 21.09.2012
- [83] TALBI, E.-G.: *Metaheuristics - From design to implementation*. John Wiley & Sons, Hoboken, New Jersey, 2009
- [84] TANG, L. ; LIU, J. ; RONG, A. ; YANG, Z.: An effective heuristic algorithm to minimize stack shuffles in selecting steel slabs from the slab yard for heating and rolling. In: *The Journal of the Operational Research Society* 52 (2001), Nr. 10, S. 1091–1097
- [85] TANG, L. ; LIU, J. ; RONG, A. ; YANG, Z.: Modelling and a genetic algorithm solution for the slab stack shuffling problem when implementing steel rolling schedules. In: *International Journal of Production Research* 40 (2002), Nr. 7, S. 1583–1595
- [86] TANG, L. ; REN, H.: Modelling and a segmented dynamic programming-based heuristic approach for the slab stack shuffling problem. In: *Computers & Operations Research* 37 (2010), Nr. 2, S. 368–375
- [87] TANG, L. ; WANG, G.: Decision support system for the batching problems of steelmaking and continuous-casting production. In: *Omega* 36 (2008), Nr. 6, S. 976–991
- [88] TANG, L. ; ZHANG, X. ; GUO, Q.: Two hybrid metaheuristic algorithms for hot rolling scheduling. In: *ISIJ International* 49 (2009), Nr. 4, S. 529–538

- [89] TANG, L. ; ZHAO, Y.: Scheduling a single semi-continuous batching machine. In: *Omega* 36 (2008), Nr. 6, S. 992–1004
- [90] THERMPROCESS, LOI: *Taschenbuch für Thermprocess-Technik*. Vulkan-Verlag, Essen, 1999
- [91] UZSOY, R.: Scheduling a single batch processing machine with non-identical job sizes. In: *International Journal of Production Research* 32 (1994), Nr. 7, S. 1615–1635
- [92] VAHRENKAMP, R.: *Produktionsmanagement*. Oldenbourg, München, 2008
- [93] VEREIN DEUTSCHER INGENIEURE: *VDI-Wärmeatlas*. Springer, Berlin, Heidelberg, 2006
- [94] VEREIN DEUTSCHER INGENIEURE: VDI 3633, Blatt 1: Simulation von Logistik-, Materialfluss- und Produktionssystemen, Grundlagen. In: *VDI-Handbuch Materialfluss und Fördertechnik, Bd. 8*. Beuth, Berlin, 2007
- [95] WANG, G. ; JIN, C. ; DENG, X.: Modeling and optimization on steel plate pick-up operation scheduling on stackyard of shipyard. In: *Proceedings of the IEEE International Conference on Automation and Logistics*, 2008, S. 548–553
- [96] WANG, H.-F. ; WU, K.-Y.: Hybrid genetic algorithm for optimization problems with permutation property. In: *Computers & Operations Research* 31 (2004), Nr. 14, S. 2453–2471
- [97] WANG, X. ; TANG, L.: Integration of batching and scheduling for hot rolling production in the steel industry. In: *The International Journal of Advanced Manufacturing Technology* 36 (2008), Nr. 5-6, S. 431–441
- [98] WANG, X. ; TANG, L.: A simplified scatter search for a special single machine problem to minimize total weighted tardiness. In: *Proceedings of the 48th IEEE Conference on Decision and Control, combined with the 28th Chinese Control Conference*, 2009, S. 6250–6255
- [99] WEGENER, I.: *Komplexitätstheorie: Grenzen der Effizienz von Algorithmen*. Springer, Berlin, Heidelberg, New York, 2003

- [100] WINSTON, P.H.: *Artificial intelligence*. Addison-Wesley, Reading, Massachusetts, 1993
- [101] WITT, A.: *Produktionsplanung- und steuerung in der Stahlindustrie*, Technische Universität Braunschweig, Diss., 2004
- [102] WITT, G.: *Taschenbuch der Fertigungstechnik*. Hanser Verlag, Leipzig, 2006
- [103] ZHAO, J. ; WANG, W. ; LIU, Q. ; WANG, Z. ; SHI, P.: A two-stage scheduling method for hot rolling and its application. In: *Control Engineering Practice* 17 (2009), Nr. 6, S. 629–641
- [104] ZHU, D.: Intelligent optimization-based production planning and simulation analysis for steelmaking and continuous casting process. In: *Journal of Iron and Steel Research, International* 17 (2010), Nr. 9, S. 19–24

Eidesstattliche Erklärung

Ich versichere wahrheitsgemäß, die Dissertation bis auf die in der Abhandlung angegebene Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und genau kenntlich gemacht zu haben, was aus Arbeiten anderer und aus eigenen Veröffentlichungen unverändert oder mit Abänderungen entnommen wurde.

Saarbrücken,

Eric Ebermann