



FH Bielefeld
University of
Applied Sciences



Angewandte Mathematische Modellierung & Optimierung

AMMO – Berichte aus Forschung und Technologietransfer

Programm zur Entscheidungsunterstützung für eine Klasse von WzP-Kommissionierungsproblemen

David Nehab, Hermann-Josef Kruse, Timo Lask

Heft Nr. 9

März 2018

Veröffentlichungsreihe (Onlinepublikation):

AMMO – Berichte aus Forschung und Technologietransfer

ISSN

2198-4824

Erscheinungsort

<https://www.fh-bielefeld.de/iium/forschung/forschungsschwerpunkte/ammo/veroeffentlichungen>

Herausgeber

Sprecher FSP AMMO, Fachhochschule Bielefeld

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

FSP Angewandte Mathematische Modellierung und Optimierung

Interaktion 1

33619 Bielefeld

Vorwort

Im AMMO-Heft Nr. 7 wurden bereits Lösungsstrategien für eine (noch recht einfache) Klasse von WzP-Kommissionierungsproblemen¹ vorgestellt und diskutiert (siehe [11]). Dabei fand eine Zweiteilung in Probleme mit und ohne Stellplatzbegrenzungen statt, wobei die Klasse ohne Stellplatzbegrenzung eher eine „akademische Fingerübung“ darstellte, um grundsätzlich die Lösungsgüte von einfachen und intuitiven Reihenfolgestrategien auszutesten. In der Praxis wird nahezu ausnahmslos eine Stellplatzbegrenzung vorzufinden sein. Zudem wurde durchweg eine Offline-Situation unterstellt, d.h. sämtliche Kommissionierungsaufträge sind zum Entscheidungszeitpunkt für die Festlegung der Artikel- bzw. Auftragsreihenfolge bekannt. In der Praxis ist sehr oft aber ein dynamischer Eingang von Kommissionierungsaufträgen anzutreffen, d.h. dass der Informationsstand eine begrenzte und wandelbare Online-Situation darstellt. Und auch ein weiteres praxisnahes Problemkriterium wurde noch ausgespart, nämlich der entscheidungsrelevante Umstand, dass gewisse Aufträge auch mit spätesten Fertigstellungszeitpunkten (Deadlines) behaftet sein können (z.B. Eilaufträge), sodass bei gewissen Artikel- bzw. Auftragsreihenfolgen Verspätungen auftreten können, die möglichst zu vermeidende Strafkosten nach sich ziehen.

In diesem Beitrag wird ein Matlab-Programm vorgestellt², welches diese praxisrelevanten Problemerkweiterungen bei der Lösungsfindung berücksichtigt. Nach wie vor sind Lösungen für Probleme mit und ohne Stellplatzbegrenzung im Offline-Fall möglich. Zur bislang hierarchisch geordneten Abarbeitungsreihenfolge der Artikel ist alternativ noch eine rollierende und eine dynamische Artikelreihenfolge hinzugefügt. Darüber hinaus wird eine strikt vorgegebene Offline-Situation (d.h. sämtliche Kommissionierungsaufträge sind von vornherein bekannt und können bei der einmaligen Festlegung von Artikel- und Auftragsreihenfolge vor Kommissionierungsbeginn berücksichtigt werden) dahingehend „aufgeweicht“, dass während eines begonnenen Kommissionierungsprozesses weitere Aufträge eingehen können, sodass folgerichtig eine permanente Überprüfung und Nachbesserung der Reihenfolgen erfolgt (dynamische Wiederholstrategie). Für diese Überprüfungen und ggf. Reihenfolgeänderungen bietet die neue Programmversion ei-

¹ WzP ist die Abkürzung für *Ware-zur-Person*; in der einschlägigen Literatur auch mit *Ware-zum-Mensch* oder auch *Ware-zum-Mann* bezeichnet.

² Genauer gesagt: eine neue Programm-Version mit entsprechenden Erweiterungen.

ne Stufeneinstellung an. Zu guter Letzt kann zu jedem Auftrag eine Deadline angegeben werden, welche im Programm dahingehend Berücksichtigung findet, dass bei der Bildung der Auftragsreihenfolgen wahlweise versucht wird, die Anzahl der Verspätungen bzw. die Summe der Verspätungszeiten (und ggf. damit verbundene Strafkosten) möglichst gering zu halten.

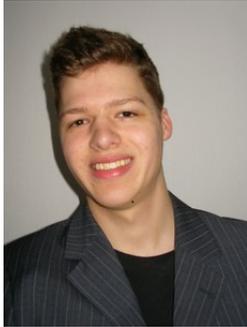
Die Entwicklung des hier vorgestellten Programms basiert auf einer Reihe von Qualifizierungsarbeiten (Bachelor-, Master- und Projektarbeiten) im Bachelor-Studiengang *Angewandte Mathematik* sowie im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik* der Fachhochschule Bielefeld, mit denen viele spezielle Untersuchungen an einer bestimmten Problemklasse aus dem Bereich der Kommissionierung (speziell *WzP-Kommissionierungsprobleme*) vorgenommen wurden.³ Hierbei wurden für unterschiedliche Fragestellungen, insbesondere im Hinblick auf verschiedene Kennzahlen bzw. Zielgrößen, spezifische Lösungsansätze entwickelt, die zum Teil bereits Eingang in die ersten Versionen 1.0-2.3 des „Kommissionier-Tools“ KMT fanden.⁴ In diesem Zusammenhang möchten sich die Autoren bei allen Mitwirkenden bedanken, die anhand ihrer Qualifikationsarbeiten einen Beitrag zur Entwicklung von Lösungsstrategien und deren Implementierung geliefert haben.

Ein(e) Leser(in), der (die) hauptsächlich am Programm KMT interessiert ist, könnte unter Umständen direkt mit der Lektüre von Kapitel 4 („Handbuch“) beginnen und bei Bedarf oder bei aufkommendem Interesse an Einzelheiten die Kapitel 2 und 3 als „Nachschlagewerk“ hinzuziehen, um sich einzelne grundlegende Begriffe und Spezifikationen in aller Ausführlichkeit klarzumachen. Einen freien Zugang zum Programm KMT bekommt man über den Link <https://github.com/tlask/KMT> (vgl. Kapitel 5).

³ Hierzu siehe [1, 3, 6, 8, 10, 13, 14, 16, 19, 20].

⁴ Hierzu siehe [5, 9, 14].

Die Autoren



David Nehab (M. Sc.)

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

dnehab@gmx.de

Hat seinen Bachelor of Engineering im Studiengang *Regenerative Energien* sowie seinen Master of Science im Studiengang *Optimierung und Simulation* an der Fachhochschule Bielefeld erlangt.



Prof. Dr. rer. pol. Hermann-Josef Kruse

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

hermann-josef.kruse@fh-bielefeld.de

Fachgebiete: Wirtschaftsmathematik, Operations Research

Seit 1995 als Professor für Wirtschaftsmathematik (insbesondere Operations Research) an der Fachhochschule Bielefeld tätig und lehrt dort im Bachelor-Studiengang *Angewandte Mathematik* und im Master-Studiengang *Optimierung & Simulation* des Fachbereichs *Ingenieurwissenschaften & Mathematik*. Gründungsmitglied des Forschungsschwerpunktes *Angewandte Mathematische Modellierung & Optimierung* (FSP AMMO) der FH Bielefeld. F&E-Projekte im Bereich der Optimierung und Simulation diskreter Systeme zur Entscheidungsunterstützung bei betrieblichen Problemstellungen.



Timo Lask (M. Sc.), geb. Kleine-Döpke

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

timo.lask@fh-bielefeld.de

Fachgebiete: Petri-Netze, Personalplanung, Echtzeitsysteme

Hat sein Master-Studium im Fach *Optimierung & Simulation* und sein Bachelor-Studium

in der *Angewandten Mathematik* an der Fachhochschule Bielefeld absolviert. Arbeitete seit der Mitte seines Bachelor-Studiums als SHK im FSP-AMMO-Projekt „Coloured Petri Nets“ (CPN) von Prof. Dr. Hermann-Josef Kruse und hat sich in den letzten Jahren unter anderem in dieser Fachrichtung spezialisiert. Seit Ende 2014 an der Fachhochschule Bielefeld als wissenschaftlicher Mitarbeiter in Forschungsprojekten beschäftigt und seit Januar 2015 ein festes AMMO-Mitglied.

Inhaltsverzeichnis

1	Einleitung	1
2	Beschreibung und Formalisierung der behandelten Problemausprägungen	3
2.1	Grundlegende Modellgrößen	3
2.2	Zielgrößen	10
2.2.1	Problemstellungen ohne Stellplatzbegrenzung	10
2.2.2	Problemstellungen mit Stellplatzbegrenzung	11
2.2.3	Deadline-orientierte Zielfunktionen	12
3	Verfahren	13
3.1	Artikelreihenfolgeverfahren	13
3.1.1	Eröffnungsstrategien	13
3.1.2	Verbesserungsstrategien	16
3.1.3	Wiederholstrategien	18
3.2	Auftragsreihenfolgeverfahren	19
3.2.1	Auftragsreihenfolgen bei spezifischen Fertigstellungszeitpunkten	19
3.2.2	Auftragsreihenfolgen bei spätesten Fertigstellungszeitpunkten . .	20
4	Das Programm KMT	22
4.1	Eingabeparameter	22
4.2	Benutzeroberfläche mit Bedienungsanleitungen	23
5	Fazit und Ausblick	37
	Literaturverzeichnis	38

1 Einleitung

In diesem Beitrag wird ein Programm vorgestellt, welches Entscheidungsunterstützung bei der Suche nach effizienten Bearbeitungsreihenfolgen für eine Klasse von bestimmten Ware-zur-Person-Kommissionierungsproblemen liefert.⁵ Dabei geht es nicht darum, die gesamte Planung und Durchführung eines konkreten Kommissionierungssystems aus der Praxis ins Auge zu fassen. Vielmehr sollen gezielte Untersuchungen an verschiedenen, sowohl artikel- als auch auftragsbezogenen Reihenfolgestrategien unter bestimmten Voraussetzungen (z.B. in Abhängigkeit von der Stellplatzanzahl im Kommissionierbereich oder vom Informationsstand über die Aufträge wie offline/online bzw. mit/ohne Deadlines) zu Aussagen über die Eignung der verschiedenen Reihenfolgestrategien führen und die Entscheidung für oder gegen deren Einsatz in einem konkreten Kommissionierungssystem unterstützen.

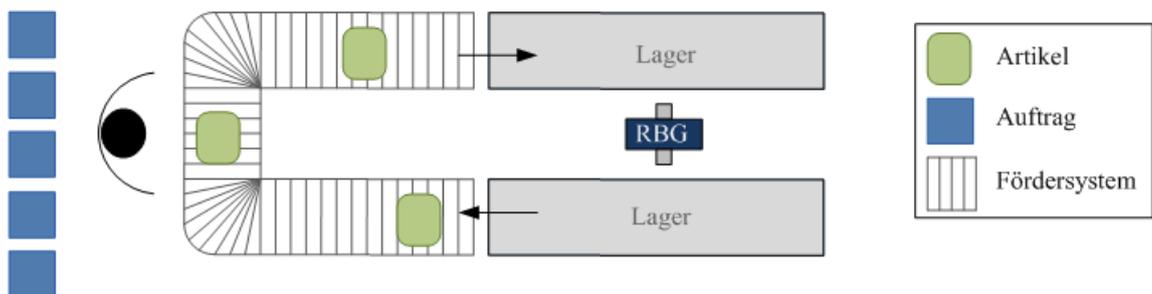


Abbildung 1: Skizzierung eines möglichen WzP-Kommissionierplatzes ([1], S. 4)

Hierzu wird ein einfaches Grundmodell erstellt (Abschn. 2.1), bei dem jeder Artikel eine vom Kommissionierungsauftrag unabhängige Transportkomponente ("Holzeit") und eine auftragsabhängige Bearbeitungskomponente ("Kommissionierzeit") zugewiesen bekommt. Für konkrete Kommissionierungsprozesse sind diese Komponenten den Gegebenheiten vor Ort entsprechend anzupassen. In Absch. 2.2 werden verschiedene praxisrelevante Kennzahlen in Form von Zielfunktionen formuliert, die es zu minimieren gilt.

In Kapitel 3 wird eine Vielzahl von Verfahren zur Bestimmung effizienter Artikel- und Auftragsreihenfolgen unter verschiedenen Problemausprägungen vorgestellt.

⁵ Zur genaueren Begriffsdefinition und Einbettung in den allgemeinen Bereich der betrieblichen Kommissionierung sei auf einschlägige Literatur verwiesen, z.B. [2, 4, 7, 12, 15, 17] oder auch [18].

1 Einleitung

Das Kernstück dieses Artikels ist das Matlab-Programm KMT in der aktuellen Version 3.0. Im Kapitel 4 wird eine ausführliche Programmbeschreibung im Stile eine Anwenderhandbuches geliefert. Damit sollte eine Leserin bzw. ein Leser ohne tiefergehende Matlab-Kenntnisse in der Lage sein, das Programm KMT 3.0 ziel- und zweckorientiert zu benutzen und eigenständige Untersuchungen und Analysen an „ihrem/seinem“ individuellen WzP-Kommissionierungsproblem durchzuführen.

Im abschließenden Kapitel 5 werden einige allgemeine Aussagen zur Effizienz der implementierten Reihenfolgestrategien gemacht und konkrete Hinweise auf einen freien Zugang zum Programm KMT gegeben.

2 Beschreibung und Formalisierung der behandelten Problemausprägungen

Die hier untersuchten Kommissionierungsprobleme bilden eine Klasse von kombinatorischen Optimierungsproblemen, die im Folgenden als mathematische Optimierungsmodelle formalisiert werden.⁶

2.1 Grundlegende Modellgrößen

Im Lager befinden sich M verschiedene Artikel P_i . Diese Artikel (oder ein Teil davon) werden innerhalb einer betrachteten Planungsperiode nachgefragt. Diese Nachfragen liegen in Form von n Kommissionierungsaufträgen A_j vor und sind in einer Auftragsserie $E = \{A_1, \dots, A_n\}$ zusammengefasst. Diese Auftragsserie enthält insgesamt m verschiedene Artikel ($m \leq M$); o. B. d. A. handelt es sich dabei um die Artikel P_1, \dots, P_m .

Die Grunddaten aller hier betrachteten Kommissionierungsprobleme lassen sich in Matrixform wiedergeben.⁷

		Aufträge				Häufigkeit
		w_{ij}	A_1	A_2	\dots	
Artikel	P_1	w_{11}	w_{12}	\dots	w_{1n}	h_1
	P_2	w_{21}	w_{22}	\dots	w_{2n}	h_2
	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
	P_m	w_{m1}	w_{m2}	\dots	w_{mn}	h_m
Auftragslänge		l_1	l_2	\dots	l_n	

Abbildung 2: Grunddaten des Kommissionierungsproblems

⁶ Vgl. [11].

⁷ An dieser Stelle sei auf [9] verwiesen; dort wird ein konkretes Beispiel mit expliziten Berechnungen der im Folgenden entwickelten Modellgrößen vorgestellt.

Eine Zuweisungsmatrix $W = (w_{ij})_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$ mit $w_{ij} \in \mathbb{N}_0$ enthält die Stückzahl⁸ von Artikel P_i im Auftrag A_j . Aus dieser Zuweisungsmatrix W entsteht auf kanonische Weise die zugehörige binäre Zuweisungsmatrix $\Omega = (\omega_{ij})_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$ durch die Setzung

$$\omega_{ij} := \begin{cases} 1 & \text{falls } w_{ij} > 0, \\ 0 & \text{falls } w_{ij} = 0. \end{cases} \quad (2.1)$$

Die binären Werte ω_{ij} geben an, ob der Artikel P_i im Auftrag A_j vorkommt oder nicht; dabei spielen die Stückzahlen eine untergeordnete Rolle. Die Häufigkeit h_i eines Artikels P_i gibt an, in wie vielen Aufträgen dieser Artikel vorkommt ($i = 1, \dots, m$). Die Häufigkeitswerte ergeben sich als Zeilensummen der binären Zuweisungsmatrix:

$$h_i := \sum_{j=1}^n \omega_{ij}, \quad i = 1, \dots, m. \quad (2.2)$$

Die Anzahl von verschiedenen Artikeln pro Auftrag (d. h. Anzahl der Positionen eines Auftrages) werden als Auftragslänge l_j bezeichnet ($j = 1, \dots, n$). Diese Werte ergeben sich als Spaltensummen der binären Zuweisungsmatrix:

$$l_j := \sum_{i=1}^m \omega_{ij}, \quad j = 1, \dots, n. \quad (2.3)$$

Im Kommissionierungsbereich stehen X Stellplätze zur Verfügung. Je Auftrag wird genau ein Stellplatz beansprucht. Es wird im Folgenden vereinbart, dass ein Auftrag A_j , dem ein Stellplatz zugewiesen worden ist, solange diesen Stellplatz in Anspruch nimmt, bis der Auftrag abgeschlossen ist, d. h. sämtliche Artikel P_i , die im Auftrag A_j vorkommen, kommissioniert worden sind.

Hinsichtlich der Stellplätze sind zwei Fälle zu unterscheiden: Die Anzahl der Stellplätze kann hinreichend groß sein, sodass jeder Auftrag unmittelbar einen solchen zugewiesen bekommt ($n \leq X$). In diesem Fall wird von „unbegrenzter Stellplatzanzahl“ gesprochen⁹, sodass sämtliche Aufträge einem parallelen Kommissionierungsprozess unterzogen werden können. Umgekehrt kann die Anzahl der Aufträge größer als die Anzahl der Stellplätze sein ($n > X$). In diesem Fall („begrenzte Stellplatzanzahl“) können nicht alle Aufträge „gleichzeitig“ kommissioniert werden, vielmehr müssen einige Aufträge zurückgestellt werden, bis andere Aufträge fertig kommissioniert sind und die jeweils frei gewordenen Stellplätze belegt werden können. Konsequenterweise wird im ersten Fall jeder benötigte Artikel nur einmal in den Kommissionierungsbereich geholt, um

⁸ Der Einfachheit halber wird hier von „Stückzahl“ gesprochen, wodurch die handelsübliche Mengeneinheit eines Artikels verallgemeinert wird.

⁹ Grundsätzlich ist die Anzahl der Stellplätze auf natürliche Weise stets begrenzt (d. h. $X < \infty$); die Sprechweisen „unbegrenzte“ bzw. „begrenzte Stellplatzanzahl“ deuten also lediglich auf die Problemrelevanz hin.

damit sämtliche Aufträge, in denen der jeweilige Artikel nachgefragt ist, zu bestücken. Im zweiten Fall wird es in der Regel nötig sein, einige Artikel mehrfach in den Kommissionierungsbereich zu holen.

Bei begrenzter Anzahl an Stellplätzen wird eine $m \times X$ -Matrix W^* benötigt:

$$W^* = (w_{ij_k}^*)_{\substack{i=1,\dots,m \\ k=1,\dots,X}} \text{ mit } j_k \in \{1, \dots, n\}. \quad (2.4)$$

Diese besteht aus X Spalten aus der Zuweisungsmatrix W , wobei die Aufträge, also die Spalten, in beliebiger Reihenfolge aufgeführt sind. Diese bestehen aus den Aufträgen, die aktuell zur Kommissionierung anstehen. Die Artikelreihenfolge (von oben nach unten) wird dadurch nicht beeinflusst, wohl aber die Auftragsreihenfolge (von links nach rechts). O. B. d. A. soll der Bequemlichkeit halber auf die Doppelindizierung $j_k \in \{1, \dots, n\}$ für $k = 1, \dots, X$ verzichtet und entsprechende Umnummerierungen $j = 1, \dots, X$ vereinbart werden.

Jeder Artikel P_i hat eine *Holzzeit* H_i und eine *Kommissionierzeit* K_i . Die Holzzeit eines Artikels ist unabhängig von der benötigten Stückzahl, da die Artikel in einer Bereitstellungseinheit zur Verfügung stehen. Die Kommissionierzeit gilt für eine Mengeneinheit eines Artikels und wird deshalb mit der Stückzahl des Artikels multipliziert. Somit lässt sich ohne Stellplatzbegrenzung die *Bearbeitungszeit* B_i eines Artikels P_i wie folgt berechnen:

$$B_i = H_i + K_i \cdot \sum_{j=1}^n w_{ij} \text{ für } i = 1, \dots, m. \quad (2.5)$$

Wenn die Stellplätze begrenzt sind, muss bei der Berechnung der Bearbeitungszeit darauf geachtet werden, ob die aktuell zu bearbeitenden Aufträge den Artikel enthalten. Ist der Artikel nicht enthalten, muss er auch nicht zur Kommissionierung bereitgestellt werden. Die Bearbeitungszeit ist somit von den Aufträgen, die sich aktuell im Kommissionierungsbereich befinden, abhängig und wird folgendermaßen berechnet:

$$B_i^* = \begin{cases} H_i + K_i \cdot \sum_{j=1}^X w_{ij}^* & \text{falls } \sum_{j=1}^X w_{ij}^* \neq 0 \\ 0 & \text{sonst} \end{cases} \text{ für } i = 1, \dots, m. \quad (2.6)$$

Die *Auftragsreihenfolge* ist eine Permutation $\alpha = (\alpha(1), \dots, \alpha(n))$ der Menge $\{1, \dots, n\}$, wobei gilt: $\alpha(k) \neq \alpha(l)$ für $k \neq l$. Sie gibt an, in welcher Reihenfolge die Aufträge bearbeitet werden. Die *Artikelreihenfolge* ist eine Permutation $\pi = (\pi(1), \dots, \pi(m))$ der Menge $\{1, \dots, m\}$, wobei ebenso gilt: $\pi(k) \neq \pi(l)$ für $k \neq l$. Mit Π_m wird die Menge aller Permutationen der Menge $\{1, \dots, m\}$ bezeichnet. Die Mächtigkeit von Π_m ist $|\Pi_m| = m!$. Dies ist die Anzahl der verschiedenen Artikelreihenfolgen. Die Anzahl der

verschiedenen Auftragsreihenfolgen ist $n!$ und wird analog berechnet. Aus diesem Grund werden heuristische Verfahren benötigt, um eine geeignete Artikel- und Auftragsreihenfolge zu bestimmen.

Die Auftrags- oder Artikelreihenfolge ohne Umsortierung wird als *topologische Reihenfolge* bezeichnet, d. h. $\alpha = (1, \dots, n)$ und $\pi = (1, \dots, m)$. Wird nun die Reihenfolge der Artikel oder Aufträge verändert, zieht es eine Veränderung der Zuweisungsmatrix mit sich:

$$W(\pi, \alpha) = (w_{ij}^{\pi\alpha})_{\substack{i=1,\dots,m \\ j=1,\dots,n}} \text{ mit } w_{ij}^{\pi\alpha} = w_{\pi(i),\alpha(j)}. \quad (2.7)$$

Die Zuweisungsmatrix ist immer abhängig von der jeweils betrachteten Artikel- und Auftragsreihenfolge. O. B. d. A. soll die Verdeutlichung, dass es sich um diese Abhängigkeit handelt, vernachlässigt werden. Somit wird nur die Bezeichnung W für die Zuweisungsmatrix genutzt.

Wie bereits oben erwähnt, wird ein Kommissionierbehälter als geöffnet angesehen, sobald der erste Artikel für diesen Auftrag kommissioniert wird. Es existiert für jeden Auftrag ein Artikel, der den Auftrag beginnt, und ein Artikel, der den Auftrag abschließt. Bzgl. der Permutation $\pi = (\pi(1), \dots, \pi(m))$ der Artikelreihenfolge wird für jeden Auftrag A_j auf kanonische Weise ein Index $s_\pi(j) \in \{1, \dots, m\}$ induziert, sodass der Artikel $P_{\pi(s_\pi(j))}$ den Auftrag *beginnt*, wobei $w_{\pi(s_\pi(j)), j} > 0$ und $w_{\pi(k), j} = 0$ für alle $k < s_\pi(j)$ gilt. Der Artikel $P_{\pi(s_\pi(j))}$ wird als $s_\pi(j)$ -ter Artikel kommissioniert bzw. hat die Stelle $s_\pi(j)$ in der Artikelreihenfolge. Außerdem wird anhand der Artikelreihenfolge π für jeden Auftrag A_j ein Index $\tilde{s}_\pi(j) \in \{1, \dots, m\}$ bestimmt, mit dem der Artikel $\pi(\tilde{s}_\pi(j))$ den Auftrag *abschließt*. Es muss $w_{\pi(\tilde{s}_\pi(j)), j} > 0$ und $w_{\pi(k), j} = 0$ für alle $k > \tilde{s}_\pi(j)$ gelten.

Der Index $s_\pi(j)$ gibt das erste Nichtnullelement in der Spalte j der gemäß Artikelreihenfolge π umgestellten Zuweisungsmatrix an und ist der *Anfangszeitpunkt* s_j des Auftrags A_j , d. h. $s_j = s_\pi(j)$. Das letzte Nichtnullelement wird durch den Index $\tilde{s}_\pi(j)$ angegeben. Der *Endzeitpunkt* der Kommissionierung ist $\tilde{s}_j = \tilde{s}_\pi(j)$.

$$w_{\pi(k), j} \begin{cases} = 0 & \text{falls } k < s_j \\ > 0 & \text{falls } k = s_j \\ \geq 0 & \text{falls } s_j < k < \tilde{s}_j \\ > 0 & \text{falls } k = \tilde{s}_j \\ = 0 & \text{falls } k > \tilde{s}_j \end{cases} \quad (2.8)$$

Es gibt zwei Arten, den Fertigstellungszeitpunkt eines Auftrags zu bestimmen. Entweder betrachtet man die zeitgleiche Fertigstellung oder die spezifizierte Fertigstellung. Bei der *zeitgleichen Fertigstellung* wird ein Auftrag A_j als fertig angesehen, wenn der letzte Artikel des Auftrags in alle geöffneten Kommissionierbehälter gepackt ist. Der *Fertigstellungszeitpunkt* des Auftrags wird mit f_j bezeichnet. Die Auftragsreihenfolge ist nicht relevant, wenn alle Aufträge gleichzeitig bearbeitet werden. Sind die Stellplätze begrenzt, ist die Auftragsreihenfolge von Bedeutung, da nur ein Teil der Aufträge gleichzeitig bearbeitet wird und somit der Fertigstellungszeitpunkt des Auftrags von der Auftragsreihenfolge wesentlich abhängt. Werden alle Aufträge gleichzeitig bearbeitet, berechnet sich der Fertigstellungszeitpunkt für einen Auftrag A_j wie folgt:

$$f_j = \sum_{k=1}^{\tilde{s}_j} B_{\pi(k)} = \sum_{k=1}^{\tilde{s}_j} \left(H_{\pi(k)} + K_{\pi(k)} \cdot \sum_{l=1}^n w_{\pi(k),l} \right) \quad \text{für } j = 1, \dots, n \quad (2.9)$$

Bei begrenzter Anzahl an Stellplätzen muss bei der Formel für den Fertigstellungszeitpunkt eines Auftrags einiges beachtet werden. Da nicht alle Aufträge gleichzeitig bearbeitet werden können, müssen die Artikel gegebenenfalls mehrfach geholt werden. Die Bearbeitung der Aufträge erfolgt in mehreren Schritten. In jedem Schritt befinden sich bis zu X Aufträge im Kommissionierbereich, wodurch die Artikel eine angepasste Bearbeitungszeit haben, wie in Formel (2.6) angegeben. In jedem Schritt $k = 1, \dots, K$ wird mindestens ein Auftrag fertig kommissioniert, sodass für die Anzahl der Schritte gilt: $K \leq n$. In jedem Schritt k wird zunächst ermittelt, welcher der X Aufträge den geringsten Endzeitpunkt hat. Dieser wird mit $\tilde{g}^{(k)}$ bezeichnet. Für den Schritt k werden in Anlehnung an (2.6) folgende Teilbearbeitungszeiten ermittelt:

$$B_{\pi(i)}^{(k)} = \begin{cases} H_{\pi(i)} + K_{\pi(i)} \cdot \sum_{l=1}^X w_{\pi(i),l}^* & \text{falls } \sum_{l=1}^X w_{\pi(i),l}^* \neq 0 \\ 0 & \text{sonst} \end{cases} \quad \text{für } i = 1, \dots, \tilde{g}^{(k)}. \quad (2.10)$$

Anschließend wird die Matrix W^* aktualisiert, indem die Bedarfswerte $w_{\pi(i),l}^*$ für $i = 1, \dots, \tilde{g}^{(k)}$ und $l = 1, \dots, X$ auf null gesetzt und die Spalten der abgeschlossenen Aufträge durch neue Aufträge gemäß der Auftragsreihenfolge ersetzt werden. Durch schrittweise Fortsetzung ($k = 1, \dots, K$) wird jeder Auftrag A_j abgeschlossen, und zwar im k_j -ten Schritt mit $k_j \in \{1, \dots, K\}$.

Unter diesen Voraussetzungen ergibt sich der *Fertigstellungszeitpunkt* f_j^* für die Auf-

träge A_j als Summen der Teilbearbeitungszeiten:

$$f_j^* = \sum_{k=1}^{k_j} \sum_{g=1}^{\tilde{g}^{(k)}} B_{\pi(g)}^{(k)} \quad \text{für } j = 1, \dots, n \quad (2.11)$$

$$\text{mit } B_{\pi(g)}^{(k)} = \begin{cases} H_{\pi(g)} + K_{\pi(g)} \cdot \sum_{l=1}^X w_{\pi(g),l}^* & \text{falls } \sum_{l=1}^X w_{\pi(g),l}^* \neq 0, \\ 0 & \text{sonst.} \end{cases}$$

Bei der *spezifizierten Fertigstellung* ist ein Auftrag fertiggestellt, sobald der letzte Artikel in den Kommissionierbehälter gepackt wird. Somit verkürzt sich die Kommissionierzeit des letzten Artikels um die Zeit, die benötigt wird, um die Kommissionierbehälter der Aufträge innerhalb der Auftragsreihenfolge hinter dem Auftrag $A_{\alpha(j)}$ mit dem jeweiligen Artikel zu bestücken. Dadurch ist die Auftragsreihenfolge von Bedeutung. Diese Art der Fertigstellung wird nur betrachtet, wenn die Stellplätze unbegrenzt sind. Der *spezifische Fertigstellungszeitpunkt* f_j^+ des Auftrags $A_{\alpha(j)}$ berechnet sich folgendermaßen:

$$f_j^+ = \sum_{k=1}^{\tilde{s}_j-1} B_{\pi(k)} + H_{\pi(\tilde{s}_j)} + K_{\pi(\tilde{s}_j)} \cdot \sum_{l=1}^j w_{\pi(\tilde{s}_j),l} \quad \text{für } j = 1, \dots, n. \quad (2.12)$$

Ein weiteres Maß für die Dauer eines Auftrags A_j ist die *Durchlaufzeit* D_j . Bei gleichzeitiger Bearbeitung aller Aufträge unterschiede sich die Durchlaufzeit nicht vom Fertigstellungszeitpunkt. Ist die Anzahl der Stellplätze jedoch begrenzt, dann ist die Durchlaufzeit geringer oder gleich dem Fertigstellungszeitpunkt, d. h. $D_j \leq f_j^*$. Es gelten dieselben Voraussetzungen wie bei der Berechnung des Fertigstellungszeitpunkts, allerdings beginnt bei den Durchlaufzeiten die Summierung erst dann, wenn der Kommissionierbehälter für den Auftrag A_j auf einen freien Stellplatz gestellt und der Auftrag bearbeitet wird. In Analogie zur Bestimmung des Schrittes k_j , in welchem der Auftrag A_j abgeschlossen wird, muss nun auch der Schritt κ_j bestimmt werden, in welchem der Auftrag A_j einen Stellplatz zugewiesen bekommt. Die Formel für die Durchlaufzeit D_j für den Auftrag A_j ergibt sich dann wie folgt:

$$D_j = \sum_{k=\kappa_j}^{k_j} \sum_{g=1}^{\tilde{g}^{(k)}} B_{\pi(g)}^{(k)} \quad \text{für } j = 1, \dots, n \quad (2.13)$$

$$\text{mit } B_{\pi(g)}^{(k)} = \begin{cases} H_{\pi(g)} + K_{\pi(g)} \cdot \sum_{l=1}^X w_{\pi(g),l}^* & \text{falls } \sum_{l=1}^X w_{\pi(g),l}^* \neq 0, \\ 0 & \text{sonst.} \end{cases}$$

Demgegenüber wird bei gleichzeitiger Bearbeitung der Aufträge eine *Kommissionierdauer* d_j eines Auftrags A_j betrachtet. Dies ist die Zeit, die der Kommissionierbehälter eines Auftrags auf einem Stellplatz verweilt. Deswegen wird die Kommissionierdauer auch *Stellplatzzeit* genannt. Für die Kommissionierdauer d_j wird die Bearbeitungszeit vom Anfangszeitpunkt s_j bis zum Endzeitpunkt \tilde{s}_j der Kommissionierung des Auftrags A_j

aufsummiert. Dabei ist es irrelevant, ob der Auftrag alle Artikel enthält, die in der Zeit kommissioniert werden.

$$d_j = \sum_{k=s_j}^{\tilde{s}_j} B_{\pi(k)} = \sum_{k=s_j}^{\tilde{s}_j} \left(H_{\pi(k)} + K_{\pi(k)} \cdot \sum_{l=1}^n w_{\pi(k), l} \right) \quad \text{für } j = 1, \dots, n \quad (2.14)$$

Es wird angestrebt, dass in den Spalten der Zuweisungsmatrix W keine Nullelemente zwischen dem ersten und dem letzten Nichtnullelement stehen, um den Auftrag ohne Zeit verursachende Unterbrechungen zu bearbeiten. Diese Zeit verursachenden Unterbrechungen in einem Auftrag A_j ist die *Unproduktivzeit* u_j . Dabei wird zwischen dem Anfangs- und Endzeitpunkt eines Auftrags die Bearbeitungszeit aufsummiert, die benötigt wird, um die Artikel zu kommissionieren, die nicht in dem jeweiligen Auftrag enthalten sind. Die Formel für die Unproduktivzeit u_j eines Auftrags A_j lautet:

$$u_j = \sum_{k=s_j}^{\tilde{s}_j} \tilde{B}_{\pi(k)} \quad \text{für } j = 1, \dots, n \quad (2.15)$$

$$\text{mit } \tilde{B}_{\pi(k)} = \begin{cases} H_{\pi(k)} + K_{\pi(k)} \cdot \sum_{l=1}^n w_{\pi(k), l} & \text{falls } w_{\pi(k), j} = 0 \\ 0 & \text{sonst} \end{cases}$$

Bei Betrachtung der Formeln (2.14) und (2.15) ist zu erkennen, dass für einen Auftrag A_j die Unproduktivzeit u_j immer kürzer ist als die Kommissionierdauer d_j . Der Wert für die Kommissionierdauer d_j eines Auftrags A_j ist wiederum kleiner oder gleich dem Wert des Fertigstellungszeitpunkts f_j . Demnach gilt für diese drei Werte:

$$u_j < d_j \leq f_j \quad \text{für } j = 1, \dots, n \quad (2.16)$$

Verfügt ein Auftrag A_j über einen spätesten Fertigstellungszeitpunkt¹⁰ DL_j , so kann sich in Abhängigkeit von Artikel- und Auftragsreihenfolge eine zeitliche Überschreitung e_j^+ dieser vorliegenden Deadline ergeben:

$$e_j^+ = \max\{f_j^*(\pi, \alpha) - DL_j; 0\}. \quad (2.17)$$

Zusätzlich zu dieser in Zeiteinheiten gemessenen Kenngröße kann auch grundsätzlich interessieren, ob ein Auftrag A_j in Abhängigkeit von Artikel- und Auftragsreihenfolge überhaupt pünktlich erledigt sein wird oder nicht; hierzu wird eine binäre Kenngröße eingeführt:

$$\tilde{e}_j^+ = \begin{cases} 1, & \text{falls } e_j^+ > 0 \\ 0, & \text{sonst} \end{cases} \quad (2.18)$$

¹⁰ Engl. *deadline* oder *due date*.

2.2 Zielgrößen

Anhand dieser eingeführten Zeiten lassen sich verschiedene Zielgrößen (oder Zielfunktionen) ableiten, die aus der Praxis kommende Kennzahlen widerspiegeln (siehe Tab. 1 - Tab. 3). Alle im Folgenden betrachteten Zielgrößen werden als Minimierungsproblem formuliert. Dabei wird zwischen Zielfunktionen für Problemstellungen *ohne* oder *mit* Stellplatzbegrenzungen unterschieden (siehe Tab. 1 bzw. Tab. 2). Für den Falle von vorliegenden spätesten Fertigstellungszeitpunkten werden auch verschiedene *deadline-orientierte* Zielgrößen eingeführt und zur Untersuchung mittels KMT angeboten (siehe Tab. 3).

2.2.1 Problemstellungen ohne Stellplatzbegrenzung

Die folgenden Zielfunktionen sind nur auf Problemstellungen anzuwenden, bei denen die Stellplätze unbegrenzt sind ($X \geq n$) und somit alle Aufträge gleichzeitig bearbeitet werden können. Die Artikel werden einmalig zur Kommissionierung bereitgestellt, da sie auf alle Kommissionierbehälter, deren Aufträge den Artikel enthalten, verteilt werden können. Abgesehen von der Zielfunktion *Minimierung der summierten spezifischen Fertigstellungszeitpunkte* ist die Auftragsreihenfolge für den Zielfunktionswert nicht relevant.¹¹

z_i	Abkürzung	Zielfunktion
z_1	$\sum f$	Minimierung der summierten Fertigstellungszeitpunkte $\min! z_1(\pi) = \sum_{j=1}^n f_j$; vgl. (2.9)
z_2	$\sum B$	Minimierung des maximalen Fertigstellungszeitpunktes $\min! z_2(\pi) = \max\{f_1, \dots, f_n\}$; vgl. (2.9)
z_3	$\sum f^+$	Minimierung der summierten spezifischen Fertigstellungszeitpunkte $\min! z_3(\pi, \alpha) = \sum_{j=1}^n f_j^+$; vgl. (2.12)
z_4	$\sum u$	Minimierung der summierten Unproduktivzeiten $\min! z_4(\pi) = \sum_{j=1}^n u_j$; vgl. (2.15)
z_5	$\max u$	Minimierung der maximalen Unproduktivzeit $\min! z_5(\pi) = \max\{u_1, \dots, u_n\}$; vgl. (2.15)
z_6	$\sum d$	Minimierung der summierten Stellplatzzeiten $\min! z_6(\pi) = \sum_{j=1}^n d_j$; vgl. (2.14)
z_7	$\max x$	Minimierung der Anzahl der Stellplätze $\min! z_7(\pi) = \max\{x_1, \dots, x_m\}$ mit $x_i = \sum_{\substack{j=1 \\ s_j \leq i \leq \bar{s}_j}}^n 1$

Tabelle 1: Zielfunktionen für Problemstellungen ohne Stellplatzbegrenzung

¹¹ Die explizite Berechnung dieser Zielfunktionswerte an einem Beispiel findet man in [9].

Anmerkungen:

- Durch die Zielfunktion z_1 wird implizit auch der durchschnittliche Fertigstellungszeitpunkt aller Aufträge minimiert. Der Unterschied liegt an einem konstanten Faktor, der allerdings für die Minimierung irrelevant ist. Um den *durchschnittlichen Fertigstellungszeitpunkt* über alle Aufträge zu erhalten, wird der Zielfunktionswert durch die Anzahl der Aufträge dividiert: $\tilde{z}_1(\pi) = \frac{1}{n} \cdot z_1(\pi)$.
- Da die Zielfunktion z_2 für jede beliebige Artikelreihenfolge stets denselben Wert liefert, stellt sie ein triviales Optimierungsproblem dar und bleibt im Folgenden weitgehend unbeachtet.
- Werden bei Verfolgung der Zielgröße z_4 nicht nur Zeiten, sondern auch Kosten betrachtet, ist diese Zielfunktion die Summe der unproduktiven Stellplatzmieten. Dabei wird der ursprüngliche Zielfunktionswert mit einem konstanten Faktor, welcher die Stellplatzmiete pro Zeiteinheit ist, multipliziert.
- Die Zielfunktion z_6 ist ähnlich zu der Zielfunktion z_4 . Der Unterschied liegt darin, dass jetzt alle Bearbeitungszeiten vom Artikel des Anfangszeitpunktes des Auftrags bis zum Artikel des Endzeitpunktes über alle Aufträge summiert werden und nicht nur die, bei denen der zugehörige Zuweisungsmatrixwert null ist. Aufgrund der Relation (2.16) ist die Summe der Stellplatzzeiten immer größer als die Summe der Unproduktivzeiten, allerdings auch kleiner oder gleich der Summe der Fertigstellungszeitpunkte: $z_4(\pi) < z_6(\pi) \leq z_1(\pi)$.
- Zwar wird anhand der Zielgröße z_7 die Anzahl der Stellplätze gezielt eingegrenzt, auf Grund der Ausgangssituation gehört diese Zielfunktion allerdings zu den Problemstellungen ohne Stellplatzbegrenzung.

2.2.2 Problemstellungen mit Stellplatzbegrenzung

Auf Problemstellungen mit begrenzter Anzahl an Stellplätzen sind die folgenden drei Zielfunktionen anwendbar.¹² Durch die begrenzte Anzahl an Stellplätzen kann nur ein Teil der Aufträge gleichzeitig bearbeitet werden, daher ist neben der Artikelreihenfolge auch die Auftragsreihenfolge von Bedeutung. Da nicht zwangsläufig alle Aufträge, die den gerade zu kommissionierenden Artikel enthalten, gleichzeitig bearbeitet werden, müssen die Artikel gegebenenfalls mehrmalig zur Kommissionierung bereitgestellt werden. Hierbei ergibt sich eine erweiterte Artikelreihenfolge, die i.d.R. keine Permutation π der Menge $\{1, \dots, m\}$ mehr darstellt.

z_i	Abkürzung	Zielfunktion
z_8	$\sum f^*$	Minimierung der summierten Fertigstellungszeitpunkte $\min! z_8(\pi, \alpha) = \sum_{j=1}^n f_j^*$; vgl. (2.11)
z_9	$\max f^*$	Minimierung des maximalen Fertigstellungszeitpunktes $\min! z_9(\pi, \alpha) = \max\{f_1^*, \dots, f_n^*\}$; vgl. (2.11)
z_{10}	$\sum D$	Minimierung der summierten Durchlaufzeiten $\min! z_{10}(\pi, \alpha) = \sum_{j=1}^n D_j$; vgl. (2.13)

Tabelle 2: Zielfunktionen für Problemstellungen mit Stellplatzbegrenzung

¹² Die explizite Berechnung der zugehörigen Zielfunktionswerte an einem Beispiel findet man ebenfalls in [9].

Anmerkungen:

- Die Zielfunktion z_8 summiert die Fertigstellungszeitpunkte aller Aufträge unter Berücksichtigung der Stellplatzbegrenzung. Dabei wird eine zeitgleiche Fertigstellung der Aufträge angenommen. Es werden X Aufträge gemäß der Auftragsreihenfolge im Kommissionierbereich bearbeitet. Aus den Aufträgen wird zuerst der mit der kürzesten Bearbeitungszeit fertiggestellt, d. h. der Auftrag mit dem minimalen Endzeitpunkt aller Aufträge im Kommissionierbereich. Dazu werden alle Artikel, die zur Fertigstellung des Auftrags benötigt werden, auf die jeweiligen Kommissionierbehälter der Aufträge im Kommissionierbereich verteilt. Nicht zwingend wird der Auftrag zuerst fertiggestellt, der in der Auftragsreihenfolge vor einem anderen steht.
- Da bei der Berechnung nicht alle Aufträge gleichzeitig bearbeitet werden, handelt es sich im Gegensatz zu der Zielfunktion z_2 um kein triviales Problem.
- Anders als bei der Berechnung des Fertigstellungszeitpunkts beginnt die Summierung der kumulierten Bearbeitungszeiten erst, wenn der Kommissionierbehälter des jeweiligen Auftrags im Kommissionierbereich einen Stellplatz zugewiesen bekommt. Da die Durchlaufzeit eines Auftrags immer kleiner oder gleich dem Fertigstellungszeitpunkt des Auftrags ist, so gilt dies auch bei der Summe aller Aufträge und damit: $z_{10}(\pi, \alpha) \leq z_8(\pi, \alpha)$.

2.2.3 Deadline-orientierte Zielfunktionen

Die folgenden Zielgrößen beziehen sich auf die spätest erlaubten Fertigstellungszeitpunkte (Deadlines) der Aufträge, soweit diese angegeben sind. Jeder Auftrag ohne explizite Deadline bekommt implizit den fiktiven Deadline-Wert ∞ zugewiesen.

z_i	Abkürzung	Zielfunktion
z_{11}	$\sum e^+$	Minimierung der summierten Überschreitungen der Deadlines $min! z_{11}(\pi, \alpha) = \sum_{j=1}^n e_j^+$; vgl. (2.17)
z_{12}	$\sum \frac{1}{n} e^+$	Minimierung der durchschnittlichen Überschreitung der Deadlines $min! z_{12}(\pi, \alpha) = \frac{1}{n} \cdot \sum_{j=1}^n e_j^+$; vgl. (2.17)
z_{13}	$max e^+$	Minimierung der maximalen Überschreitung der Deadlines $min! z_{13}(\pi, \alpha) = max\{e_1^+, \dots, e_n^+\}$; vgl. (2.17)
z_{14}	$\sum \tilde{e}^+$	Minimierung der Anzahl an Überschreitungen von Deadlines $min! z_{14}(\pi, \alpha) = \sum_{j=1}^n \tilde{e}_j^+$; vgl. (2.18)
z_{15}	$\sum g \cdot \tilde{e}^+$	Minimierung der summierten Überschreitungsgewichte $min! z_{15}(\pi, \alpha) = \sum_{j=1}^n g_j \cdot \tilde{e}_j^+$; vgl. (2.18)

Tabelle 3: Deadline-orientierte Zielfunktionen

Anmerkungen:

- Die Zielfunktion z_{12} ist zwar eine einfache Herleitung aus z_{11} , sie wird aber in KMT explizit angeboten.
- Bei den Gewichten g_j in z_{15} kann es sich um individuelle Strafkosten bei Terminüberschreitungen der einzelnen Aufträge A_j handeln (z.B. Konventionalstrafen), die möglichst gering (im Idealfall gleich null) sein sollen.

3 Verfahren

In diesem Kapitel werden verschiedene heuristische Verfahren vorgestellt, um effiziente Artikel- und Auftragsreihenfolgen zu ermitteln. Dabei handelt es sich um einfache Anordnungsstrategien nach bestimmten Auswahlkriterien.

3.1 Artikelreihenfolgeverfahren

Zunächst werden intuitive Artikelanordnungen nach bestimmten Auswahlkriterien vorgestellt (Abschn. 3.1.1). Danach wird auf *Verbesserungsverfahren* eingegangen, die zur Optimierung der Artikelreihenfolge verwendet werden, wobei als „Startlösung“ für diese Verbesserungsverfahren die Lösung eines *Eröffnungsverfahrens* genutzt wird (Abschn. 3.1.2). Diese Strategien sind unter der Prämisse anwendbar, dass keine Stellplatzbegrenzung und eine Offline-Situation vorliegt. Im Falle einer Stellplatzbegrenzung oder bei begrenzter Information über die Aufträge (Online-Situation) werden *Wiederholstrategien* benötigt, die in Abschn. 3.1.3 behandelt werden.

3.1.1 Eröffnungsstrategien

Bei diesen Verfahren werden die Artikel nach gewissen Kriterien sortiert. Dabei weisen die Kriterien unterschiedliche Prioritäten auf. Es werden ein bis zwei Kriterien für die Sortierung genutzt (*Primär-* bzw. *Sekundärkriterium*). Implizit wird die Eindeutigkeit der Lösung durch ein abschließendes *Indexkriterium* erzwungen, d.h. dass im Falle einer Mehrfachlösung anhand von Primär- (und ggf. Sekundär-) Kriterium der vorliegende Index entscheidet: $P_i \prec P_k \Leftrightarrow i < k$ (\prec bedeutet „kommt vor“).¹³

Überwiegend werden Verfahren vorgestellt, die eine effiziente Lösung liefern. Allerdings gibt es auch Verfahren, deren Lösungen per se ineffizient sind. Dadurch soll verdeutlicht werden, dass *keine* Sortierung der topologischen Reihenfolge (d.h. allein gemäß Indexkriterium) besser ist als eine nach „verschlimmbessernden“ Kriterien.

In den folgenden Übersichtstabellen (siehe Tabelle 4 und Tabelle 5) werden zunächst

¹³ Analog für Auftragsreihenfolgen: $A_j \prec A_k \Leftrightarrow j < k$.

19 verschiedene Artikelreihenfolgeverfahren vorgestellt.¹⁴ Durch die Verfahren werden verschiedene Artikelreihenfolgen bestimmt. Einige Verfahren wurden gezielt für spezielle Zielfunktionen entwickelt.¹⁵ Grundsätzlich aber lassen sich die Verfahren generell anwenden, also auch auf Zielgrößen, für die sie nicht spezifisch entwickelt wurden. Demnach kann entsprechend der Wahl der Zielfunktion die Effizienz der Verfahren gewissen Schwankungen unterliegen.¹⁶

Abkürzung	Verfahren
PTopo	Anordnung in topologischer Reihenfolge („Indexkriterium“) $P_i \prec P_k \Leftrightarrow i < k$ (primär)
MinBMaxH	Anordnung nach minimaler Bearbeitungszeit und maximaler Häufigkeit $P_i \prec P_k \Leftrightarrow B_i < B_k$ (primär), $h_i > h_k$ (sekundär), $i < k$ (tertiär)
MaxBMaxH	Anordnung nach maximaler Bearbeitungszeit und maximaler Häufigkeit $P_i \prec P_k \Leftrightarrow B_i > B_k$ (primär), $h_i > h_k$ (sekundär), $i < k$ (tertiär)
MinHTopo	Anordnung nach minimaler Häufigkeit und topologischer Reihenfolge $P_i \prec P_k \Leftrightarrow h_i < h_k$ (primär), $i < k$ (sekundär)
MaxHTopo	Anordnung nach maximaler Häufigkeit und topologischer Reihenfolge $P_i \prec P_k \Leftrightarrow h_i > h_k$ (primär), $i < k$ (sekundär)
MaxHMinB	Anordnung nach maximaler Häufigkeit und minimaler Bearbeitungszeit $P_i \prec P_k \Leftrightarrow h_i > h_k$ (primär), $B_i < B_k$ (sekundär), $i < k$ (tertiär)
MaxHMaxB	Anordnung nach maximaler Häufigkeit und maximaler Bearbeitungszeit $P_i \prec P_k \Leftrightarrow h_i > h_k$ (primär), $B_i > B_k$ (sekundär), $i < k$ (tertiär)
ZAufH	Anordnung nach zum Zentrum aufsteigender Häufigkeit (Anmerk. 1) $P_i \prec P_k \Leftrightarrow \begin{cases} h_i < h_k & \text{für } i, k \leq \lfloor \frac{m}{2} \rfloor \\ h_i > h_k & \text{für } i, k > \lfloor \frac{m}{2} \rfloor \end{cases}$ (primär), $i < k$ (sekundär)
ZAbH	Anordnung nach zum Zentrum absteigender Häufigkeit (Anmerk. 2) $P_i \prec P_k \Leftrightarrow \begin{cases} h_i > h_k & \text{für } i, k \leq \lfloor \frac{m}{2} \rfloor \\ h_i < h_k & \text{für } i, k > \lfloor \frac{m}{2} \rfloor \end{cases}$ (primär), $i < k$ (sekundär)

Tabelle 4: Artikelreihenfolgeverfahren im Überblick (1)

Anmerkungen:

- (1) Zu Beginn des Verfahrens *Anordnung nach zum Zentrum aufsteigender Häufigkeit* (ZAufH) werden die Artikel in zwei Mengen geteilt. Die erste Teilmenge besteht aus den ersten $\lfloor \frac{m}{2} \rfloor$ Artikeln und die zweite Teilmenge aus den letzten $\lceil \frac{m}{2} \rceil$ Artikeln. Ist die Anzahl der Artikel ungerade, so enthält die erste Teilmenge ein Element weniger als die zweite Teilmenge; ansonsten sind die Teilmengen gleichmächtig. Die Artikel aus der ersten Teilmenge werden aufsteigend nach den Häufigkeiten sortiert, wodurch eine Artikelreihenfolge $\pi_1 = (\pi_1(1), \dots, \pi_1(\lfloor \frac{m}{2} \rfloor))$ entsteht. Im Gegensatz dazu werden die Artikel aus der zweiten Teilmenge absteigend nach den Häufigkeiten sortiert. Dadurch wird eine zweite Artikelreihenfolge $\pi_2 = (\pi_2(\lfloor \frac{m}{2} \rfloor + 1), \dots, \pi_2(m))$ gebildet. Bei gleicher Häufigkeit werden die Artikel topologisch angeordnet. Die Artikelreihenfolgen der beiden Teilmengen werden zu $\pi = (\pi_1, \pi_2)$ zusammengefasst. Somit stehen die Artikel mit der größten Häufigkeit in der Mitte bzw. im

¹⁴ Ausführliche Beschreibungen findet man in [11].

¹⁵ Hierzu siehe [1, 19, 16, 8, 5].

¹⁶ Näheres hierzu findet man in [10, 11].

Zentrum der Artikelreihenfolge und die Artikel mit einer niedrigen Häufigkeit am Anfang und am Ende. In analoger Weise gilt diese Prozedur für das Verfahren .

- (2) Das Verfahren *Anordnung nach zum Zentrum absteigender Häufigkeit* (ZAbH) ähnelt dem zuvor beschriebenen Verfahren. Zu Beginn des Verfahrens werden die Artikel in zwei Mengen geteilt. Die ersten $\lfloor \frac{m}{2} \rfloor$ Artikel werden der ersten Teilmenge zugeteilt. Dementsprechend besteht die zweite Teilmenge aus den letzten $\lceil \frac{m}{2} \rceil$ Artikeln. Für die Artikelreihenfolge $\pi_1 = (\pi_1(1), \dots, \pi_1(\lfloor \frac{m}{2} \rfloor))$ erfolgt die Sortierung der Artikel der ersten Teilmenge nach absteigender Häufigkeit. Im Gegensatz dazu werden die Artikel aus der zweiten Teilmenge für die Artikelreihenfolge $\pi_2 = (\pi_2(\lfloor \frac{m}{2} \rfloor + 1), \dots, \pi_2(m))$ aufsteigend nach den Häufigkeiten sortiert. Weisen mehrere Artikel dieselbe Häufigkeit auf, werden diese Artikel topologisch angeordnet. Durch die Zusammenfassung der Artikelreihenfolgen der beiden Teilmengen ergibt sich eine Artikelreihenfolge $\pi = (\pi_1, \pi_2)$. In dieser Artikelreihenfolge stehen die Artikel mit einer niedrigen Häufigkeit im Zentrum. Dementsprechend sind die Artikel nach aufsteigender Häufigkeit zum Anfang und zum Ende angeordnet.

Das in den folgenden Anordnungsstrategien verwendete Grundverfahren *Anordnung nach minimaler Auftragslänge* (MinA) mit jeweiligem Sekundärkriterium benötigt m Wiederholungen des folgenden Ablaufs, um eine Artikelreihenfolge zu ermitteln. Dazu wird eine Teilreihenfolge der Artikel genutzt, die in jedem Schritt um einen Artikel erweitert wird und am Ende die Artikelreihenfolge ergibt. In jedem Schritt werden zuerst die Aufträge mit der minimalen Auftragslänge, die jedoch größer als null ist, gesucht, also die Aufträge, die noch nicht fertiggestellt sind und möglichst wenig unterschiedliche Artikel zur Fertigstellung benötigen. Aus der Menge dieser Aufträge wird der darin enthaltene gemäß Sekundärkriterium „beste“ (d.h. topologisch erste bzw. mit der geringsten bzw. längsten Bearbeitungszeit bzw. mit der minimalen bzw. maximalen Häufigkeit) und noch nicht in der Teilreihenfolge aufgenommene Artikel gewählt und an die Teilreihenfolge „angehängt“. Daraufhin werden die Auftragslängen neu berechnet, wobei die bereits in der Teilreihenfolge enthaltenen Artikel aus der Berechnung ausgeschlossen sind. Somit wird in jedem Schritt für die noch nicht fertiggestellten Aufträge die Anzahl der unterschiedlichen Artikel, die noch zur Fertigstellung benötigt werden, aktualisiert.

Beim analogen Grundverfahren *Anordnung nach maximaler Auftragslänge* (MaxA) werden schrittweise zuerst die Aufträge gesucht werden, die möglichst viele unterschiedliche Artikel zur Fertigstellung benötigen. Dementsprechend wird aus der Menge der Aufträge mit der maximalen Auftragslänge der gemäß Sekundärkriterium „beste“ bestimmt. Dieser Artikel wird an die aktuelle Teilreihenfolge angefügt und aus der weiteren Berechnung ausgeschlossen. Daraufhin werden die Auftragslängen aktualisiert.

Abkürzung	Verfahren
MinATopo	Anordnung nach minimaler Auftragslänge und topologischer Reihenfolge $P_i \prec P_k \Leftrightarrow \text{MinA (primär)}, i < k \text{ (sekundär)}$
MinAMinB	Anordnung nach minimaler Auftragslänge und minimaler Bearbeitungszeit $P_i \prec P_k \Leftrightarrow \text{MinA (primär)}, B_i < B_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MinAMaxB	Anordnung nach minimaler Auftragslänge und maximaler Bearbeitungszeit $P_i \prec P_k \Leftrightarrow \text{MinA (primär)}, B_i > B_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MinAMinH	Anordnung nach minimaler Auftragslänge und minimaler Häufigkeit $P_i \prec P_k \Leftrightarrow \text{MinA (primär)}, h_i < h_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MinAMaxH	Anordnung nach minimaler Auftragslänge und maximaler Häufigkeit $P_i \prec P_k \Leftrightarrow \text{MinA (primär)}, h_i > h_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MaxATopo	Anordnung nach maximaler Auftragslänge und topologischer Reihenfolge $P_i \prec P_k \Leftrightarrow \text{MaxA (primär)}, i < k \text{ (sekundär)}$
MaxAMinB	Anordnung nach maximaler Auftragslänge und minimaler Bearbeitungszeit $P_i \prec P_k \Leftrightarrow \text{MaxA (primär)}, B_i < B_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MaxAMaxB	Anordnung nach maximaler Auftragslänge und maximaler Bearbeitungszeit $P_i \prec P_k \Leftrightarrow \text{MaxA (primär)}, B_i > B_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MaxAMinH	Anordnung nach maximaler Auftragslänge und minimaler Häufigkeit $P_i \prec P_k \Leftrightarrow \text{MaxA (primär)}, h_i < h_k \text{ (sekundär)}, i < k \text{ (tertiär)}$
MaxAMaxH	Anordnung nach maximaler Auftragslänge und maximaler Häufigkeit $P_i \prec P_k \Leftrightarrow \text{MaxA (primär)}, h_i > h_k \text{ (sekundär)}, i < k \text{ (tertiär)}$

Tabelle 5: Artikelreihenfolgeverfahren im Überblick (2)

3.1.2 Verbesserungsstrategien

Die vorgestellten Verbesserungsverfahren werden zur Optimierung einer Artikelreihenfolge verwendet. Als Startlösung wird die Lösung eines Eröffnungsverfahrens aus Abschn. 3.1.1 genutzt. Die Anwendung der Verbesserungsverfahren erfolgt dementsprechend auf eine Artikelreihenfolge, die als Permutation (also ohne Wiederholungen) angegeben ist. Somit sind diese Verbesserungsverfahren (in der hier behandelten Art) nur im Zusammenhang mit den Zielfunktionen für Problemstellungen ohne Stellplatzbegrenzung und im Offline-Zustand anzuwenden.

Die Tabelle 6 zeigt eine Übersicht der sechs Verbesserungsverfahren mit Kurzbeschreibungen; ausführliche Erläuterungen mit jeweiligem Pseudocode findet man in [11]. Dabei werden drei verschiedene Grundsatzverfahren vorgestellt, die wiederum in jeweils zwei Ausprägungen unterteilt sind, indem zwischen der Variante *First-Fit* und *Best-Fit* unterschieden wird; entsprechend bekommen die Verfahren den Namenszusatz *First* bzw. *Best*. Mit diesen Verfahren wird systematisch in der Nachbarschaft der Artikelreihenfolge eine bessere Lösung gesucht. Als *Nachbarschaft* werden alle Artikelreihenfolgen angesehen, die durch eine bestimmte Modifikation erzeugt werden können. In allen Verfahren ist die Anzahl der Nachbarn $N = \frac{m \cdot (m-1)}{2}$, allerdings mit unterschiedlichen Nachbarschaften.

Bei der Variante *First-Fit* wird das systematische Prüfen der Nachbarschaft abgebrochen, sobald eine bessere Lösung ermittelt wird. Daraufhin beginnt das Verfahren mit der verbesserten Artikelreihenfolge als neue Startlösung von Neuem. Im Gegensatz dazu wird bei der Variante *Best-Fit* zunächst die gesamte Nachbarschaft auf bessere Lösungen geprüft. Wenn bessere Artikelreihenfolgen ermittelt werden, wird die mit der größten Verbesserung als neue Startlösung gewählt und das Verfahren erneut gestartet.

Verfahren	Kurzbeschreibung
2-Tausch	<p>In der aktuellen Artikelreihenfolge werden systematisch zwei Artikel vertauscht. Dabei bleibt die Reihenfolge der übrigen Artikel erhalten. Daraufhin wird geprüft, ob durch den Tausch eine Verbesserung des Zielfunktionswertes entsteht. Sobald eine Verbesserung festgestellt wird, wird die verbesserte Lösung zur aktuellen Artikelreihenfolge erklärt und das Verfahren wird mit dieser fortgesetzt.</p> <p>First-Fit-Variante: 2-Tausch First (TF) Best-Fit-Variante: 2-Tausch Best (TB)</p>
1-Verschieben	<p>Aus der aktuellen Artikelreihenfolge wird systematisch ein Artikel entnommen und an einer anderen Stelle platziert. Dabei erfolgt die Verschiebung des ausgewählten Artikels in der Artikelreihenfolge ausschließlich nach hinten. Die Reihenfolge der anderen Artikel bleibt dabei erhalten. Daraufhin wird geprüft, ob durch die Verschiebung eine Verbesserung des Zielfunktionswertes erzielt wird. In diesem Fall wird die verbesserte Lösung zur aktuellen Artikelreihenfolge erklärt und das Verfahren wird mit dieser fortgesetzt.</p> <p>First-Fit-Variante: 1-Verschieben First (VF) Best-Fit-Variante: 1-Verschieben Best (VB)</p>
Drehung	<p>Aus der aktuellen Artikelreihenfolge wird systematisch eine Teilkette von Artikeln in der Artikelreihenfolge gedreht. Dazu werden zwei Artikel bestimmt, die mit den dazwischenliegenden Artikeln eine Teilkette bilden. Diese Teilkette wird einmal gedreht bzw. gespiegelt. Daraufhin wird geprüft, ob durch die Drehung eine Verbesserung des Zielfunktionswertes erzielt wird. In diesem Fall wird die verbesserte Lösung zur aktuellen Artikelreihenfolge erklärt und das Verfahren wird mit dieser fortgesetzt. Diese Vorgehensweise basiert auf dem <i>2-opt-Verfahren</i> zur Lösung von Traveling-Salesman-Problemen.</p> <p>First-Fit-Variante: Drehung First (DF) Best-Fit-Variante: Drehung Best (DB)</p>

Tabelle 6: Verbesserungsverfahren im Überblick

Die Laufzeit der Verbesserungsverfahren kann enorm lang sein, besonders bei großen

Auftragsserien. Aus diesem Grund werden bei der Implementierung dieser Verfahren zusätzliche Abbruchkriterien berücksichtigt.¹⁷

3.1.3 Wiederholstrategien

Im Falle von Problemstellungen ohne Stellplatzbegrenzung (d.h. $X \geq n$) unter der zusätzlichen Annahme, dass zum Zeitpunkt der Disposition einer effizienten Artikelreihenfolge *alle* Kommissionierungsaufträge der Planungsperiode bekannt sind (*Offline-Situation*), kann die Lösung als eine Permutation π über der Indexmenge $\{1, \dots, m\}$ angegeben werden. Dabei wird jeder Artikel nur einmal geholt und dem Kommissionierungsprozess zugeführt: $P_{\pi(1)} \prec P_{\pi(2)} \prec \dots \prec P_{\pi(m-1)} \prec P_{\pi(m)}$.

Falls allerdings eine Stellplatzbegrenzung zu beachten ist ($X < n$) oder der Informationsstand über die zu disponierenden Aufträge ist eingeschränkt, indem während der Kommissionierungsvorgänge weitere Aufträge hinzukommen und im laufenden Prozess eingeplant werden sollen (*Online-Situation*), müssen gewisse Artikel wiederholt zur Kommissionierung bereitgestellt werden, sodass die Lösung als eine endliche Folge anzugeben ist, in der einige Indizes mehrfach vorkommen (können). Für diesen Fall werden drei verschiedene *Wiederholstrategien* zur Auswahl gestellt¹⁸. Dabei wird jeweils eine zuvor ausgewählte Artikelreihenfolgestrategie aus Abschn. 3.1.1 zu Grunde gelegt.

Hierarchische Variante:

Bei Problemstellungen mit Stellplatzbegrenzung wird im Offline-Zustand zu Beginn anhand der ausgewählten Artikelreihenfolgestrategie eine Permutation π ermittelt. Gemäß dieser Artikelreihenfolge werden die X Aufträge auf den belegten Stellplätzen kommissioniert. Sobald ein Auftrag fertig ist, bekommt der nächste Auftrag $X + 1$ den frei gewordenen Stellplatz. Als nächster Artikel wird gemäß der festgelegten Hierarchie $P_{\pi(1)} \prec P_{\pi(2)} \prec \dots \prec P_{\pi(m-1)} \prec P_{\pi(m)}$ der erste Artikel, der in der aktualisierten Auftragsliste vorkommt, bestimmt, d.h. es wird grundsätzlich immer „wieder von vorne“ begonnen.

Rollierende Variante:

Im Gegensatz zur hierarchischen Variante wird unter denselben Ausgangsvoraussetzungen stets der nächste Artikel geholt; d.h. falls bei Fertigstellung eines Auftrages der zuletzt kommissionierte Artikel $P_{\pi(i)}$ ist, wird mit $P_{\pi(i+1)}$ fortgefahren, und erst nach $P_{\pi(m)}$ wird wieder mit $P_{\pi(1)}$ begonnen.

¹⁷ Genaueres hierzu wird in Kapitel 4 beschrieben.

¹⁸ Das Programm KMT bietet zu Vergleichszwecken auch die Option an, alle Wiederholstrategien parallel durchzuführen.

Dynamische Variante:

Während in den beiden obigen Varianten die Artikelreihenfolge π gemäß der ausgewählten Artikelreihenfolgestrategie einmalig festgelegt und im weiteren Verlauf (d.h. auch nach Neubelegungen von Stellplätzen) beibehalten wird, erfolgt nun eine permanente Aktualisierung der Artikelreihenfolge. Bereits zu Beginn werden nicht sämtliche n Aufträge ins Auge gefasst, sondern immer nur diejenigen X Aufträge, welche aktuell Stellplätze belegen. Die Häufigkeit der Aktualisierungen der Artikelreihenfolge kann anhand eines Eingabeparameters gesteuert werden.¹⁹

Im Offline-Fall mit Stellplatzbegrenzung sind alle drei Varianten auswählbar. Im Online-Fall kommt automatisch die dynamische Variante zum Einsatz.²⁰

3.2 Auftragsreihenfolgeverfahren

In diesem Abschnitt werden verschiedene heuristische Verfahren vorgestellt, um eine effiziente Auftragsreihenfolge zu ermitteln. Diese ist dann von entscheidender Bedeutung, wenn es sich um spezielle Zielgrößen handelt.

3.2.1 Auftragsreihenfolgen bei spezifischen Fertigstellungszeitpunkten

Zunächst werden Auftragsreihenfolgestrategien vorgestellt, die im Wesentlichen bei Zielgröße z_3 mit spezifischen Fertigstellungszeitpunkten (vgl. Tab. 1) und für Problemstellungen mit Stellbegrenzung (vgl. Tab. 2) zur Anwendung kommen (siehe Tab. 7).

Abkürzung	Verfahren
ATopo	Anordnung in topologischer Reihenfolge $A_j \prec A_k \Leftrightarrow j < k$ (primär)
AufSF	Anordnung nach aufsteigenden spezifischen Fertigstellungszeitpunkten $A_j \prec A_k \Leftrightarrow f_j^+ < f_k^+$ (primär), $j < k$ (sekundär); vgl. (2.12)
AbSF	Anordnung nach absteigenden spezifischen Fertigstellungszeitpunkten $A_j \prec A_k \Leftrightarrow f_j^+ > f_k^+$ (primär), $j < k$ (sekundär); vgl. (2.12)

Tabelle 7: Auftragsreihenfolgeverfahren im Überblick

Die Auftragsreihenfolgeverfahren werden zusammen mit einem Artikelreihenfolgeverfahren genutzt. Dabei wird zuerst die Artikelreihenfolge und danach die Auftragsreihenfolge bestimmt.

¹⁹ Da die Bestimmung der Artikelreihenfolge eine nicht zu vernachlässigende Rechenzeit in Anspruch nimmt, kann vorweg eingestellt werden, nach wie vielen neu hinzukommenden Aufträgen eine Aktualisierung stattfinden soll.

²⁰ Weitere Einzelheiten zu den Wiederholstrategien, insbesondere im Online-Fall, findet man in [3].

3.2.2 Auftragsreihenfolgen bei spätesten Fertigstellungszeitpunkten

In diesem Abschnitt werden drei Konstruktionsverfahren vorgestellt, mit denen effiziente Auftragsreihenfolgen für Aufträge A_j , die mit spätesten Fertigstellungszeitpunkten DL_j versehen sind, im Hinblick auf deadline-orientierte Zielfunktionen erstellt werden (vgl. Abschn. 2.2.3). Dabei wird eine Problemstellung mit Stellplatzbegrenzung und Offline-Situation vorausgesetzt. Anschließend werden zwei Verfahren diskutiert, mit deren Hilfe versucht wird, vorbestimmte Auftragsreihenfolgen zu verbessern.

Earliest deadline first (EDF)

Die Aufträge werden aufsteigend nach ihren Deadlines angeordnet:

$$A_j \prec A_k \Leftrightarrow DL_j < DL_k \text{ (primär), } f_j^* < f_k^* \text{ (sekundär), } j < k \text{ (tertiär); vgl. (2.11).}$$

Least slacktime (LST)

Die Aufträge werden aufsteigend nach ihren Pufferzeiten angeordnet, wobei unter der Pufferzeit eines Auftrages die Zeitspanne zwischen dem geplanten und spätesten Fertigstellungszeitpunkt zu verstehen ist:

$$A_j \prec A_k \Leftrightarrow DL_j - f_j^* < DL_k - f_k^* \text{ (primär), } j < k \text{ (sekundär); vgl. (2.11).}$$

Diese *statische* Variante (LST statisch) lässt sich noch durch eine *dynamische* Variante (LST dynamisch) verfeinern. Während bei der statischen Variante die Pufferzeiten nach obiger Formel zu Beginn des Kommissionierungsprozesses (also zum Zeitpunkt $t = 0$) einmalig berechnet werden, wobei zudem die Stellplatzbegrenzung außer Acht gelassen wird, erfolgt bei der dynamischen Variante eine Berücksichtigung der Stellplatzbegrenzung. In einem ersten Schritt werden die Aufträge wie bei der statischen Variante angeordnet und zum Teil den X Stellplätzen zugeführt. Danach wird jedes Mal, wenn ein Auftrag fertiggestellt ist, dynamisch überprüft, welcher der noch zu bearbeitenden Aufträge bei der aktuellen Artikelreihenfolge die niedrigste Pufferzeit aufweist. Dieser wird dann auf den frei gewordenen Kommissionierplatz gestellt.²¹

2-Tausch lite

Dieses Verfahren verbessert eine bereits bestehende Lösung, indem Aufträge, die ihre Deadlines überschreiten, „nach vorne getauscht“ werden, falls möglich und effektiv. Letzteres wird zu erreichen versucht, indem derjenige Auftrag „nach hinten rückt“, der die größte Pufferzeit besitzt. Ist also α die aktuelle Auftragsreihenfolge, Φ die Menge aller Aufträge mit Deadlineüberschreitung und SL_j die Pufferzeit von Auftrag A_j , so wird ein Auftrag $A_{\alpha(k)} \in \Phi$ mit dem Auftrag $A_{\alpha(\ell)}$ mit $SL_{\alpha(\ell)} = \max\{SL_{\alpha(j)} \mid j < k\}$

²¹ Eine genauere Beschreibung und zusätzliche Einzelheiten findet man in [14], S. 29ff.

getauscht. Danach wird der Zielfunktionswert dieser neuen Auftragsreihenfolge berechnet. Falls ein besserer Zielfunktionswert gefunden ist, liegt eine neue Auftragsreihenfolge vor; diese ist dann neuer Ausgangspunkt des Tauschverfahrens. Dieser Tauschprozess wird solange wiederholt, bis alle Aufträge aus Φ durchprobiert worden sind.

Der Tauschprozess kann dahingehend verfeinert werden, dass nicht nur der Auftrag mit der größten Pufferzeit als Tauschkandidat ins Auge gefasst wird, sondern im Falle keiner Verbesserung auch die Aufträge mit der zweit- oder drittgrößten Pufferzeit ausprobiert werden; allgemein $T = 1, 2, 3, \dots$.²²

Insert

Dieses alternative Verbesserungsverfahren handelt nach dem „Prinzip der besten Einbettung“ (vgl. 1-Verschieben). Die Aufträge mit Deadlineüberschreitungen werden nacheinander aus der Auftragsreihenfolge α entnommen und zunächst vor dem direkten Vorgänger eingefügt, wodurch im ersten Schritt ein unmittelbarer Tausch entsteht:

$\dots, A_{\alpha(k-2)}, A_{\alpha(k)}, A_{\alpha(k-1)}, A_{\alpha(k+1)}, \dots$

Falls die hierdurch entstandene Auftragsreihenfolge keine Verbesserung des Zielfunktionswertes erzielt wird, so wird dieser Einfügeschritt bis zu T mal wiederholt:

$\dots, A_{\alpha(k-t-1)}, A_{\alpha(k)}, A_{\alpha(k-t)}, A_{\alpha(k-t+1)}, \dots, A_{\alpha(k-1)}, A_{\alpha(k+1)}, \dots$ für $t \in \{1, \dots, T\}$.²³

²² Eine ausführlichere Beschreibung mit Beispielrechnung findet man in [14], S. 62ff.

²³ Eine ausführlichere Beschreibung mit Beispielrechnung findet man in [14], S. 68ff.

4 Das Programm KMT

Für die oben beschriebene Klasse von WzP-Kommissionierungsproblemen ist ein Optimierungstool entwickelt worden, welches in diesem Kapitel ausführlich beschrieben wird. Das Optimierungstool wird im Folgenden auch *Kommissioniertool (KMT)* genannt. Auf die benötigten Eingabeparameter wird im ersten Abschnitt eingegangen (Abschn. 4.1). Daraufhin wird die Oberfläche des KMT beschrieben (Abschn. 4.2). Dabei stehen der Aufbau und die Benutzung im Vordergrund. Als Software zur Erstellung des KMT wird MATLAB R2017a verwendet .

4.1 Eingabeparameter

Im KMT wird zu einer Auftragsserie E eine effiziente Artikel- und Auftragsreihenfolge ermittelt. Dazu wird die Zuweisungsmatrix A benötigt, welche die Stückzahlen des Artikels P_i im Auftrag A_j enthält. Somit ist A eine $m \times n$ -Matrix, wobei m die Anzahl der Artikel und n die Anzahl der Aufträge ist (vgl. Abb. 2). Des Weiteren werden die Hol- und Kommissionierzeiten der Artikel benötigt, welches zwei m -dimensionale Spaltenvektoren HZ und KZ sind. Außerdem können die Deadlines der Aufträge in einem n -dimensionalen Spaltenvektor DL hinterlegt werden. Zusätzlich kann die Gewichtung der Aufträge in einem n -dimensionalen Spaltenvektor g hinterlegt werden. Dieser Vektor wird nur bei Auswahl der Zielfunktion „Min! Sum. Gewichtete Überschreitung der Deadline“ verwendet und kann auch manuell nachgetragen werden. Die Eingabeparameter A , HZ , KZ und DL werden mit genau diesen Bezeichnungen in einer MAT-Datei hinterlegt, wie in der folgenden Abbildung veranschaulicht.

Name	Value
A	5x7 double
DL	[100;96;50;Inf;136;Inf;40]
HZ	[7.7000;5.0300;1.4100;2.6400;3.4700]
KZ	[0.4200;0.1000;0.2100;0.1600;0.3600]

Abbildung 3: Eingabeparameter in einer MAT-Datei

Wichtig ist, dass bei der Eingabe von Zahlen die englische Schreibweise genutzt wird. Ansonsten wird die Zahl von dem Programm fehlerhaft interpretiert. Außerdem dürfen in der Zuweisungsmatrix keine Nullzeilen oder -spalten enthalten sein. Darüber hinaus müssen alle Werte der Zuweisungsmatrix ganzzahlig und nichtnegativ sein, da es sich um

Stückzahlen handelt. Für die Hol- und Kommissionierzeiten gilt ebenfalls die Nichtnegativität der Werte. Die Deadlines dürfen aus dem Intervall $[0; \infty)$ stammen.

4.2 Benutzeroberfläche mit Bedienungsanleitungen

Der Aufbau des KMT ist in Abb. 4 veranschaulicht. In den entsprechenden Fenstern werden Eingaben getätigt, Auswahlen getroffen und Ergebnisse ausgegeben. Im Folgenden werden der Aufbau und die Benutzeroberfläche beschrieben. Dabei wird zuerst auf die Wahl der Auftragsserie eingegangen. Daraufhin wird im nächsten Fenster zwischen einer oder keiner Stellplatzbegrenzung gewählt. Entscheidet man sich für eine Stellplatzbegrenzung, muss im Anschluss ausgewählt werden, ob den Aufträgen eine jeweilige Deadline zugeordnet werden soll oder nicht, wodurch der weitere Verlauf des Programms bestimmt wird.

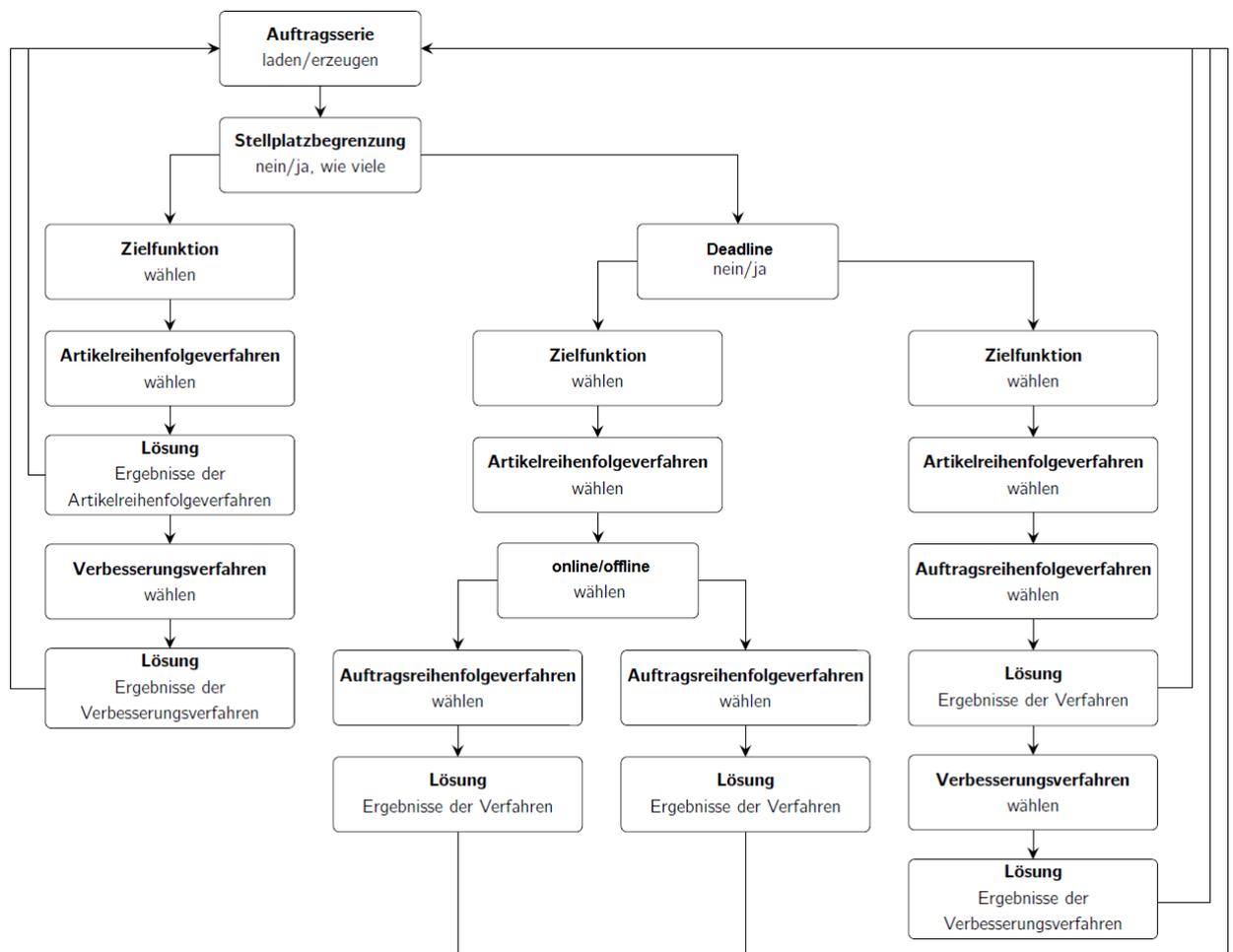


Abbildung 4: Aufbau des Kommissioniertools

In jedem Fenster der Oberfläche gibt es eine Infoleiste und einen Infobutton. Der Infobutton befindet sich oben rechts in der Ecke. Durch das Anklicken erscheint ein Infofeld mit einem kurzen Beschreibungstext zum aktuellen Fenster.

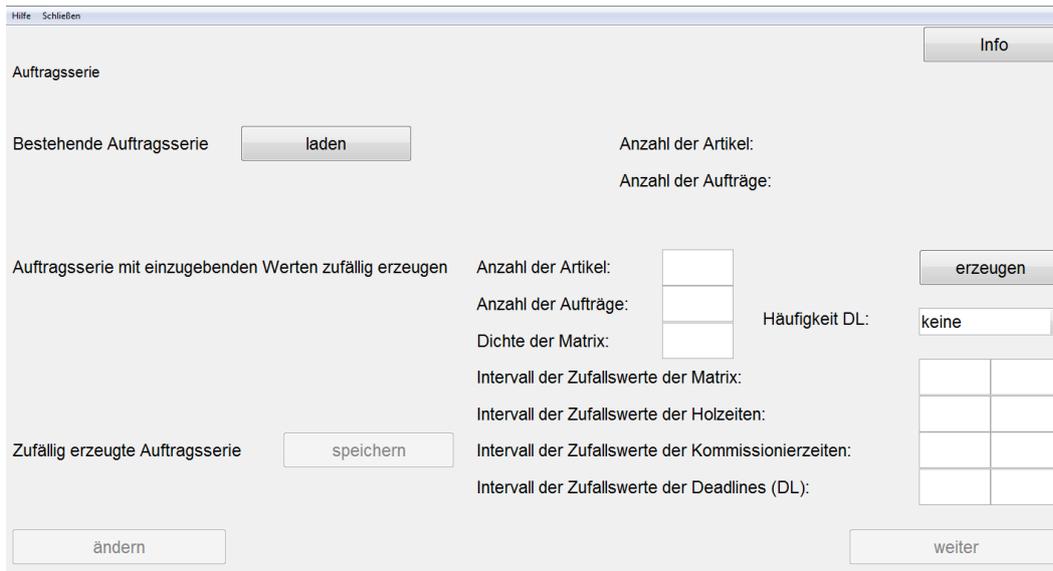


Abbildung 5: Auftragsserie laden oder erstellen

Abbildung 5 zeigt das Fenster, um eine bestehende Auftragsserie zu laden oder eine Auftragsserie mit einzugebenden Werten zufällig zu erzeugen. Die zufällig erzeugte Auftragsserie kann gespeichert werden. Im Folgenden wird zuerst das Laden einer Auftragsserie und daraufhin das Erzeugen einer Auftragsserie erläutert. Es können nur Auftragsserien geladen werden, die zuvor als MAT-Dateien gespeichert worden sind. Wie oben bereits beschrieben, sind in diesen MAT-Dateien die Eingabeparameter hinterlegt. Durch das Klicken auf den Button „laden“ kann eine Datei ausgewählt und geladen werden. Nach dem erfolgreichen Laden der Datei verändert sich die Oberfläche, wie in Abb. 6 zu sehen ist. In der Oberfläche wird der Teil zum zufälligen Erzeugen einer Auftragsserie ausgegraut. Im oberen Teil werden der Dateiname, die Anzahl der Artikel und die Anzahl der Aufträge angegeben. Mit den nun nicht mehr ausgegrauten Buttons kann entweder die Auftragsserie geändert oder zum nächsten Schritt übergegangen werden.

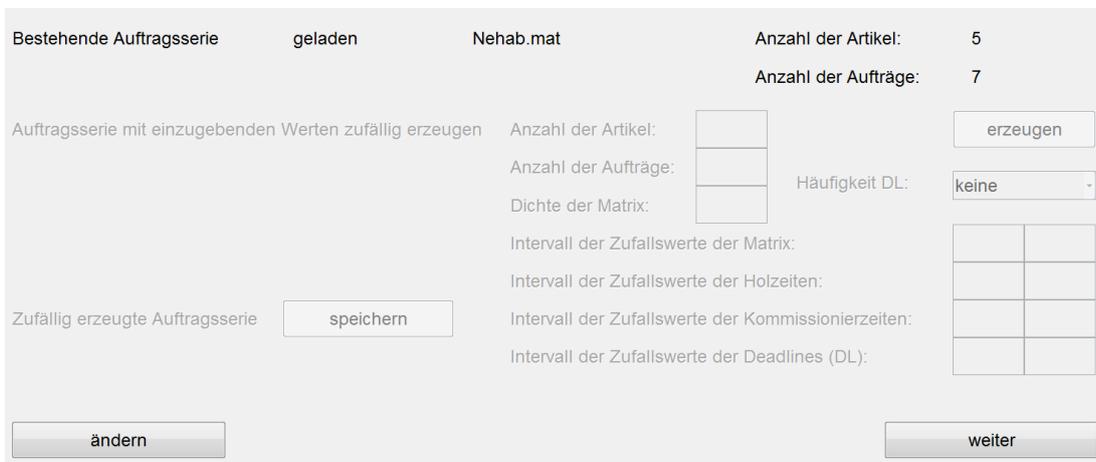


Abbildung 6: Auftragsserie wurde geladen

Um eine Auftragsserie zufällig zu erzeugen, muss bei der Eingabe der Werte auf Verschiedenes geachtet werden. Die Anzahl der Artikel und Aufträge muss ganzzahlig und

positiv sein. Als Dichte der Zuweisungsmatrix können Werte aus dem Intervall $(0; 1]$ eingegeben werden. Mit Dichte wird das Verhältnis zwischen Null- und Nichtnullelementen in der Matrix bezeichnet. Dabei gibt der Wert 1 an, dass in der Matrix keine Nullelemente vorkommen. Der eingegebene Wert für die Dichte wird als Richtwert angesehen, die tatsächliche Dichte kann davon leicht abweichen. Treten beim zufälligen Erzeugen der Matrixelemente Nullzeilen oder -spalten auf, werden diese nachträglich an einer zufälligen Stelle mit einem zufälligen Wert korrigiert. Für das Intervall der Zufallswerte der Matrix dürfen nur ganzzahlige nichtnegative Werte eingegeben werden. Die Werte für das Intervall der Zufallswerte der Hol- und Kommissionierzeiten müssen ebenfalls nichtnegativ sein. Allerdings dürfen Dezimalwerte verwendet werden, solange sie nicht mehr als drei Nachkommastellen aufweisen. Es ist darauf zu achten, dass beim Intervall der Zufallswerte kein Nachkomma-Anteil gebildet wird, wenn ganzzahlige Werte angegeben werden. Dementsprechend wird z. B. 5.00 eingegeben, wenn Zufallswerte mit zwei Nachkommastellen gewünscht sind. Nach der Eingabe aller benötigten Werte wird durch das Anklicken des Buttons „erzeugen“ eine zufällige Auftragsserie generiert. Das Speichern der erzeugten Auftragsserie ist möglich.

Im nächsten Schritt des KMT wird die Wahl der Anzahl der Stellplätze getroffen. Abbildung 7 zeigt die Wahlmöglichkeiten. Entscheidet man sich für eine Stellplatzbegrenzung, klickt man auf den Button „Ja“ und gibt daraufhin die Anzahl der Stellplätze ein. Es ist darauf zu achten, dass als Anzahl der Stellplätze eine ganzzahlige positive Zahl angegeben wird. Außerdem darf der Wert nicht die Anzahl der Aufträge überschreiten. Wird die Anzahl der Aufträge als Wert angegeben, liegt keine echte Stellplatzbegrenzung vor.

The screenshot shows a user interface with two rows of controls. The first row is for 'Stellplatzbegrenzung' (Job site limitation), featuring three buttons: 'Ja' (Yes), 'Nein' (No), and 'ändern' (Change). To the right of these buttons is a text label 'Anzahl der Stellplätze:' followed by an empty input field. The second row is for 'Deadline', featuring three buttons: 'Ja', 'Nein', and 'ändern'.

Abbildung 7: Stellplatzbegrenzung und Deadline wählen

Hat man sich bei der Stellplatzbegrenzung für „Ja“ entschieden, muss anschließend ausgewählt werden, ob Aufträge mit oder ohne Deadline untersucht werden sollen. Dazu werden die jetzt noch ausgegrauten Buttons in Abb. 7 aktiv. Klickt man dort auf „Ja“, so ist durch diese Wahl der weitere Verlauf des Programms der rechts dargestellte Pfad in Abb. 4. Klickt man auf „Nein“, so wird der mittlere Pfad in Abb. 4 ausgewählt. Der linke Pfad wird eingeschlagen, wenn der Button „Nein“ bei der Stellplatzbegrenzung geklickt wird.

Klickt man auf „Ja“ bei der Auswahl der Deadline und ist nicht allen Aufträgen eine

Deadline zugeordnet, so wird den betroffenen Aufträgen automatisch eine unendlich hohe Deadline zugewiesen.

Auf den Fall, dass die Anzahl der Stellplätze unbegrenzt ist, wird im Folgenden zuerst eingegangen. Für Problemstellungen ohne Stellplatzbegrenzung werden die Zielfunktionen z_1 bis z_7 genutzt (vgl. Tab. 1). Eine dieser Zielfunktionen wird in einem Pop-Up-Menü ausgewählt. Das Pop-Up-Menü ist in Abb. 8 dargestellt. Dabei sind die Zielfunktionen in sprechender Form angegeben. Standardeinstellung ist die Zielfunktion „Minimierung der summierten Fertigstellungszeitpunkte“.

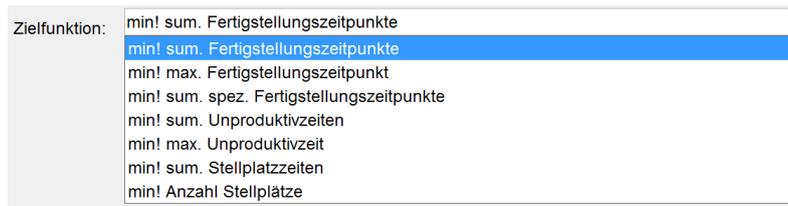


Abbildung 8: Auswahl der Zielfunktionen ohne Stellplatzbegrenzung

Nach der Wahl der Zielfunktion kann entweder „zurück“ oder „weiter“ gegangen werden. Letzteres erzeugt das Fenster aus Abb. 9, welche die Wahl der Artikelreihenfolgeverfahren zeigt. Über Checkboxen sind beliebig viele der 19 verschiedenen Artikelreihenfolgeverfahren auszuwählen. Zur Auswahl aller Artikelreihenfolgeverfahren kann der Button „alle auswählen“ genutzt werden. Mit dem Button „keins auswählen“ wird die gesamte Auswahl wieder aufgehoben. Nachdem mindestens ein Verfahren ausgewählt wurde, kann durch die Betätigung des Buttons „lösen“ die Berechnung gestartet werden.

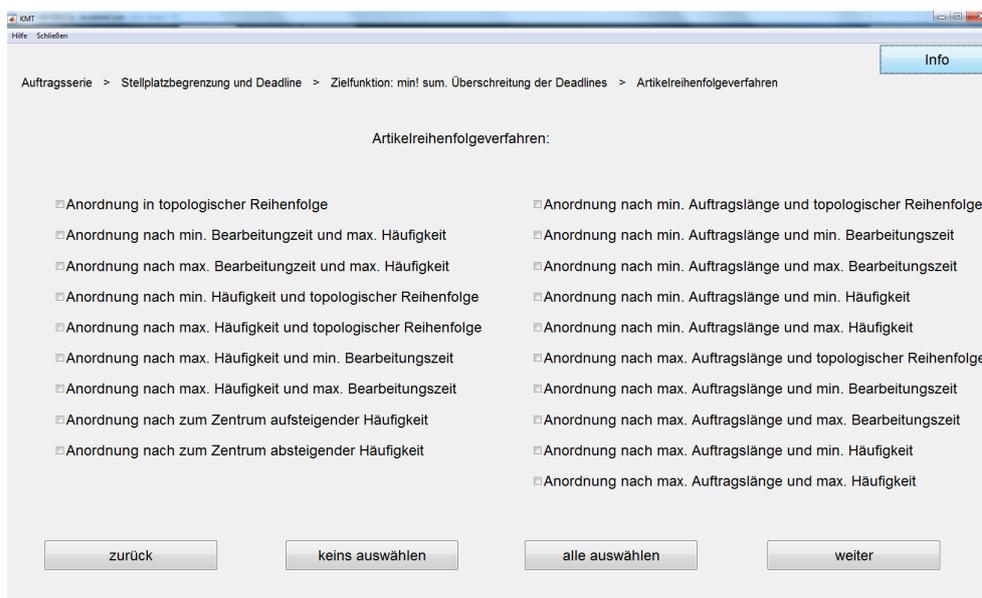


Abbildung 9: Auswahl der Artikelreihenfolgeverfahren

Daraufhin öffnet sich ein Infofeld, welches Auskunft über den Fortschritt der Berechnung gibt (siehe Abb. 10). In diesem Fenster kann die Berechnung abgebrochen werden, woraufhin die Auswahl der Artikelreihenfolgeverfahren geändert werden kann.

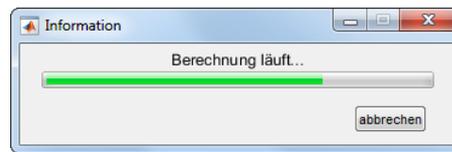


Abbildung 10: Infofeld zur Berechnung

Wenn die Berechnung der Artikelreihenfolgeverfahren vollständig abgeschlossen ist, werden die Ergebnisse in einem neuen Fenster in einer Tabelle ausgegeben. Ein Ausschnitt dieser Tabelle ist in Abb. 11 dargestellt. In der ersten Spalte der Tabelle werden die zuvor ausgewählten Artikelreihenfolgeverfahren angegeben. Der Zielfunktionswert zu der zuvor ausgewählten Zielfunktion wird in der zweiten Spalte aufgeführt. Daraufhin wird in der dritten Spalte die relative Abweichung zu der besten Lösung angegeben. Als beste Lösung wird in diesem Zusammenhang der geringste Zielfunktionswert bezeichnet. Die beste Lösung hat somit eine relative Abweichung von 0% und wird **rot** hinterlegt. In der nächsten Spalte ist die benötigte Rechenzeit in Sekunden angegeben. Die letzte Spalte gibt die ermittelte Artikelreihenfolge an. Für die Zielfunktion Minimierung der summierten spezifischen Fertigstellungszeitpunkte weist die Tabelle eine weitere Spalte auf, welche die ermittelte Auftragsreihenfolge angibt.

Verfahren	Zielfunktionswert	Abweichung (%)	Rechenzeit (s)	Artikelreihenfolge
Anordnung in topologischer Reihenfolge	251.9800	17.8964	0.0058	1 2 3 4 5
Anordnung nach min. Bearbeitungszeit und max. Häufigkeit	213.7300	0	0.0177	3 4 2 5 1
Anordnung nach max. Bearbeitungszeit und max. Häufigkeit	291.1200	36.2092	0.0070	1 5 2 4 3
Anordnung nach min. Häufigkeit und topologischer Reihenfolge	280.0900	31.0485	0.0077	1 2 5 4 3
Anordnung nach max. Häufigkeit und topologischer Reihenfolge	233.2400	9.1283	0.0045	3 4 1 2 5

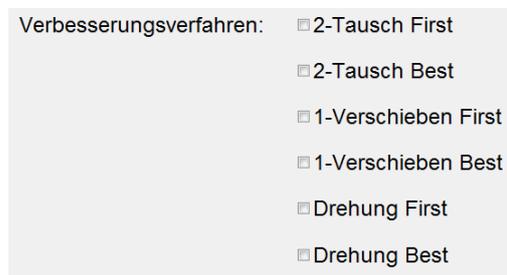
Abbildung 11: Lösung der Artikelreihenfolgeverfahren

In dem Fenster sind unter der Tabelle vier Buttons zu finden. Durch das Anklicken des Buttons „speichern“ werden die Ergebnisse sowohl in einer MAT-Datei als auch in einer XLSX-Datei gespeichert. In einem Infofeld wird ausgegeben, unter welchen Dateinamen die Ergebnisse zu finden sind. Die übrigen drei Buttons sind für den weiteren Verlauf vorgesehen. Um erneut zu der Auswahl der Artikelreihenfolgeverfahren (siehe Abb. 9) zu gelangen, nutzt man den Button „zurück“. Möchte man hingegen eine neue Auftragsreihe untersuchen, ist ein Neustart des KMT sinnvoll, welcher durch den entsprechenden Button erfolgen kann. Dabei bleiben die Einstellungen nicht erhalten. Außerdem können die Ergebnisse verbessert werden, worauf im Folgenden eingegangen wird.

Das Fenster zum Wählen der Artikelreihenfolgeverfahren, worauf die Verbesserungsverfahren angewandt werden sollen, ist Abb. 9 sehr ähnlich. Es werden nur die Artikelreihenfolgeverfahren aufgelistet, welche in Abb. 9 ausgewählt wurden. Die Artikelreihenfolgeverfahren, die einen Zielfunktionswert von null erzielt haben und somit nicht mehr verbesserbar sind, werden in diesem Fenster ausgegraut und sind daher nicht wählbar. Nachdem mindestens ein Artikelreihenfolgeverfahren zum Verbessern gewählt

wurde, gelangt man mit dem Button „weiter“ zum nächsten Schritt. In dem Fenster für die Verbesserungsverfahren sind die Verbesserungsverfahren selbst im linken Teil und die zusätzlichen Abbruchbedingungen im rechten Teil dargestellt.

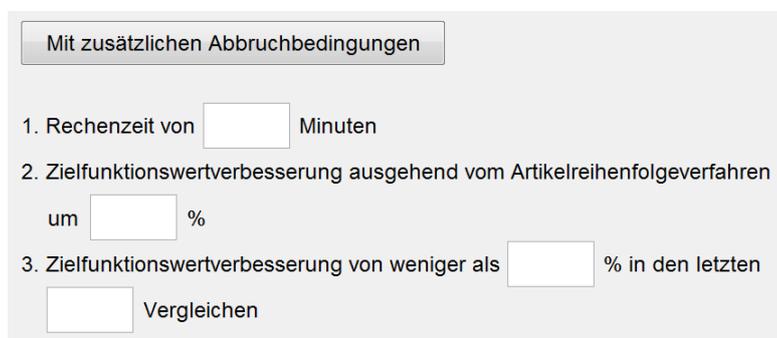
Die Abbildung 12 zeigt die Wahl der Verbesserungsverfahren. Über Checkboxen sind beliebig viele Verfahren auszuwählen. Erneut existieren die Buttons „alle auswählen“ und „keins auswählen“. Unter der Voraussetzung, dass mindestens ein Verbesserungsverfahren ausgewählt ist, wird die Berechnung durch das Anklicken des Buttons „lösen“ gestartet. Daraufhin öffnet sich erneut ein Infopfeld, welches Auskunft über den Fortschritt der Berechnung gibt (siehe Abb. 10). Mit Hilfe des Infopfeldes kann die Berechnung wieder abgebrochen und danach die Auswahl der Verbesserungsverfahren geändert werden.



Verbesserungsverfahren: 2-Tausch First
 2-Tausch Best
 1-Verschieben First
 1-Verschieben Best
 Drehung First
 Drehung Best

Abbildung 12: Auswahl der Verbesserungsverfahren

Die Berechnung der Verbesserungsverfahren kann relativ lange dauern, besonders bei einer großen Auftragsserie. Aus diesem Grund werden bei der Implementierung der Verbesserungsverfahren zusätzliche Abbruchbedingungen berücksichtigt. Bis zu drei zusätzliche Abbruchbedingungen (siehe Abb. 13) können gesetzt werden, wobei sie nicht zwingend zusammen genutzt werden. Durch das Anklicken des Buttons „Mit zusätzlichen Abbruchbedingungen“ werden diese aktiv. Daraufhin muss mindestens eine zusätzliche Abbruchbedingung angegeben werden.



Mit zusätzlichen Abbruchbedingungen

1. Rechenzeit von Minuten

2. Zielfunktionswertverbesserung ausgehend vom Artikelreihenfolgeverfahren um %

3. Zielfunktionswertverbesserung von weniger als % in den letzten Vergleichen

Abbildung 13: Zusätzliche Abbruchbedingungen der Verbesserungsverfahren

Unter der Voraussetzung, dass das Verbesserungsverfahren seinen Suchprozess noch nicht beendet hat, wird bei der ersten zusätzlichen Abbruchbedingung das Verfahren nach einer zeitlichen Vorgabe abgebrochen und die momentan beste Lösung ausgegeben. Dazu wird

zuvor die Rechenzeit in Minuten eingegeben, die maximal für die Berechnung benötigt werden darf. Dieser Wert muss positiv sein. Die zweite zusätzliche Abbruchbedingung nutzt die relative Verbesserung des Zielfunktionswerts ausgehend von der Startlösung. Dabei bricht das Verfahren ab, wenn sich der Zielfunktionswert um einen bestimmten Wert verbessert hat. Als relative Verbesserung können Werte aus $(0; 100)$ eingegeben werden. Dabei ist sowohl der Intervallanfang als auch das Intervallende ausgeschlossen. Bei einer Verbesserung des Zielfunktionswertes um 0% würde das Verfahren sofort abbrechen, wohingegen bei einem Wert von 100% keine zusätzliche Abbruchbedingung benötigt wird. Die Bedingung für den Eingabewert gilt auch bei der dritten zusätzlichen Abbruchbedingung. Außerdem wird als zweiter Eingabewert die Anzahl der Vergleiche benötigt. Dieser Wert muss positiv und ganzzahlig sein. Das Verbesserungsverfahren bricht ab, wenn der Zielfunktionswert in der vorgegebenen Anzahl an Vergleichen um weniger als den prozentualen Wert verbessert wird.

Die zusätzlichen Abbruchbedingungen sind sinnvoll, da besonders am Anfang der Verbesserungsverfahren große Verbesserungen erzielt werden. Zur Demonstration (siehe Abb. 14) wird eine Testdatei mit einer Auftragsserie, die aus 20 Artikeln besteht, benutzt. Als Startlösung wird die topologische Reihenfolge der Artikel genutzt. Daraufhin wird das Verfahren 2-Tausch Best angewandt. In Abb. 14 ist die relative Verbesserung zu jedem Vergleich im Verfahren angegeben. Es ist zu erkennen, dass mit steigender Anzahl der Vergleiche der Zuwachs der relativen Verbesserung abnimmt. Wenn auf ein paar Prozent der relativen Verbesserung verzichtet wird, können fast die Hälfte der Vergleiche, und somit Zeit, gespart werden. Für andere Verfahren ist der Verlauf ähnlich.

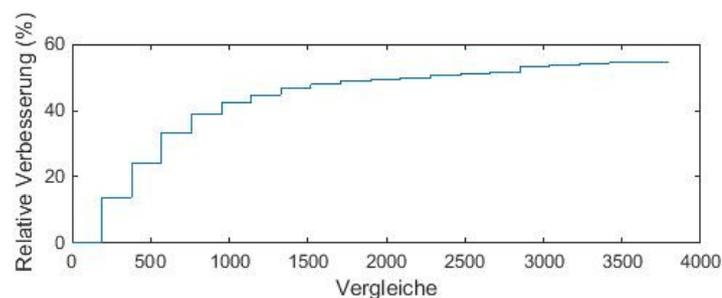


Abbildung 14: 2-Tausch-Best-Verfahren mit $m = 20$

Die Ergebnisse der Verbesserungsverfahren werden in einem neuen Fenster in einer Tabelle ausgegeben, sobald die Berechnung abgeschlossen ist. Dieses Fenster ist mit einem Teil der Tabelle in Abb. 15 dargestellt.

[Hilfe](#) [Schließen](#)

[Auftragsserie](#) > [Stellplatzbegrenzung und Deadline](#) > [Zielfunktion: min! sum. Fertigstellungszeitpunkte](#) > [Artikelreihenfolgeverfahren](#) > [Lösung](#) > [Verbesserungsverfahren](#) > [Lösung](#)

Verfahren	Zielfunktionswert	Abweichung (%)	Absolute Verbesserung	Relative Verbesserung (%)	Anz. Vergleiche bis Best.	Anz. Verbesserungen	Anz. Nachbarn	Rechenzeit (s)	Artikelreihenfolge
Anordnung in topologischer Reihenfolge	251.9800	17.8964						0.0058	1 2 3 4 5
---2-Tausch First	213.7300	0	38.2500	17.8964	26	4	10	0.0573	3 4 2 5 1
---2-Tausch Best	213.7300	0	38.2500	17.8964	30	3	10	0.0639	3 2 4 5 1
---1-Verschieben First	213.7300	0	38.2500	17.8964	25	5	10	0.0233	3 4 2 5 1
---1-Verschieben Best	213.7300	0	38.2500	17.8964	20	2	10	0.0230	3 2 4 5 1
---Drehung First	213.7300	0	38.2500	17.8964	26	4	10	0.0285	3 4 2 5 1
---Drehung Best	213.7300	0	38.2500	17.8964	30	3	10	0.0226	3 4 2 5 1
Anordnung nach min. Bearbeitungszeit und max. Häufigkeit	213.7300	0						0.0177	3 4 2 5 1
---2-Tausch First	213.7300	0	0	0	0	0	10	0.0022	3 4 2 5 1
---2-Tausch Best	213.7300	0	0	0	0	0	10	0.0026	3 4 2 5 1
---1-Verschieben First	213.7300	0	0	0	0	0	10	0.0022	3 4 2 5 1
---1-Verschieben Best	213.7300	0	0	0	0	0	10	0.0024	3 4 2 5 1
---Drehung First	213.7300	0	0	0	0	0	10	0.0021	3 4 2 5 1
---Drehung Best	213.7300	0	0	0	0	0	10	0.0025	3 4 2 5 1
Anordnung nach max. Bearbeitungszeit und max. Häufigkeit	291.1200	36.2092						0.0070	1 5 2 4 3
---2-Tausch First	213.7300	0	77.3900	36.2092	39	6	10	0.0054	3 4 2 5 1
---2-Tausch Best	213.7300	0	77.3900	36.2092	20	2	10	0.0034	3 4 2 5 1
---1-Verschieben First	213.7300	0	77.3900	36.2092	30	7	10	0.0044	3 4 2 5 1
---1-Verschieben Best	213.7300	0	77.3900	36.2092	40	4	10	0.0063	3 4 2 5 1
---Drehung First	213.7300	0	77.3900	36.2092	33	5	10	0.0051	3 4 2 5 1
---Drehung Best	213.7300	0	77.3900	36.2092	10	1	10	0.0027	3 4 2 5 1
Anordnung nach min. Häufigkeit und topologischer Reihen...	280.0900	31.0485						0.0077	1 2 5 4 3

zurück
Neustart
speichern
schließen

Abbildung 15: Ergebnisse der Verbesserungsverfahren

In der ersten Spalte der Tabelle werden die Verfahren angegeben. Dabei werden die Artikelreihenfolgeverfahren mit den darauf angewandten Verbesserungsverfahren in Blöcken zusammengefasst. In einem solchen Block wird das Artikelreihenfolgeverfahren zuerst angegeben. Darunter werden die Verbesserungsverfahren aufgelistet. Die nächste Spalte gibt den Zielfunktionswert für alle Verfahren an. Die dritte Spalte beinhaltet die relative Abweichung zu der besten Lösung. In diesem Zusammenhang wird der geringste Zielfunktionswert von der Menge aller Verfahren als beste Lösung bezeichnet. Erneut ist die beste Lösung **rot** hinterlegt. In den nächsten beiden Spalten werden die absolute und relative Abweichung des Zielfunktionswertes von dem Ergebnis des Verbesserungsverfahrens zu der Startlösung, also dem Ergebnis des Artikelreihenfolgeverfahrens, angegeben. Die sechste Spalte beinhaltet die Anzahl der Vergleiche, bis die beste Lösung gefunden wird. Das Verbesserungsverfahren benötigt N weitere Vergleiche zur Prüfung, ob dies wirklich die beste Lösung in der Nachbarschaft ist. Die Anzahl der Nachbarn ist in der achten Spalte angegeben. Der siebten Spalte ist die Anzahl der Verbesserungen zu entnehmen. In den letzten zwei, gegebenenfalls drei Spalten ist erneut die benötigte Rechenzeit in Sekunden, die ermittelte Artikelreihenfolge und die ermittelte Auftragsreihenfolge angegeben.

Werden bei den Verbesserungsverfahren zusätzliche Abbruchbedingungen berücksichtigt, enthält die Tabelle zwischen der fünften und sechsten Spalte zwei zusätzliche Spalten. In der ersten zusätzlichen Spalte wird angegeben, welche Abbruchbedingung zum Abbruch des Verbesserungsverfahrens geführt hat. Dabei steht der Wert 0 für einen kompletten Durchlauf des Verbesserungsverfahrens. Die Werte 1, 2 und 3 geben die eingetretene zusätzliche Abbruchbedingung mit der Bedeutung aus Abb. 13 an. Die zweite zusätzliche Spalte beinhaltet die Anzahl der insgesamt durchgeführten Vergleiche im Verbesserungsverfahren.

Im Folgenden wird der in der Mitte von Abb. 4 dargestellte Pfad erläutert, also der Pfad mit Stellplatzbegrenzung und ohne Berücksichtigung einer Deadline.

Für Problemstellungen mit Stellplatzbegrenzung werden die Zielfunktionen z_8 bis z_{10} genutzt (vgl. Tab. 2). Die gewünschte Zielfunktion wird aus dem in Abb. 16 dargestellten Pop-Up-Menü ausgewählt.

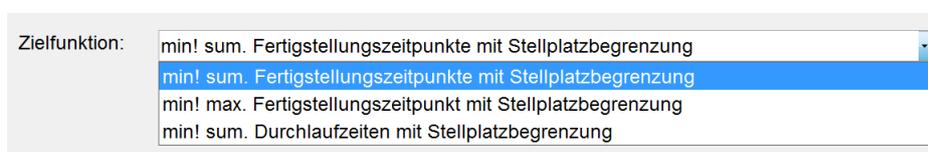


Abbildung 16: Auswahl der Zielfunktionen mit Stellplatzbegrenzung

Um zur Auswahl der Artikelreihenfolgeverfahren zu gelangen, wird nach der Wahl der Zielfunktion der Button „weiter“ betätigt. Daraufhin erscheint ein neues Fenster (siehe Abb. 17), welches dem aus Abb. 9 sehr ähnelt. Der einzige Unterschied ist, dass nun für jedes der 19 Artikelverfahren drei separate Checkboxes existieren, wodurch das entsprechende Verfahren mit einer hierarchischen, rollierenden bzw. dynamischen Holstrategie angewendet werden kann (vgl. Abschn. 3.1.3). Außerdem ist der „lösen“-Button durch einen „weiter“-Button ersetzt worden (vgl. Abb. 9).

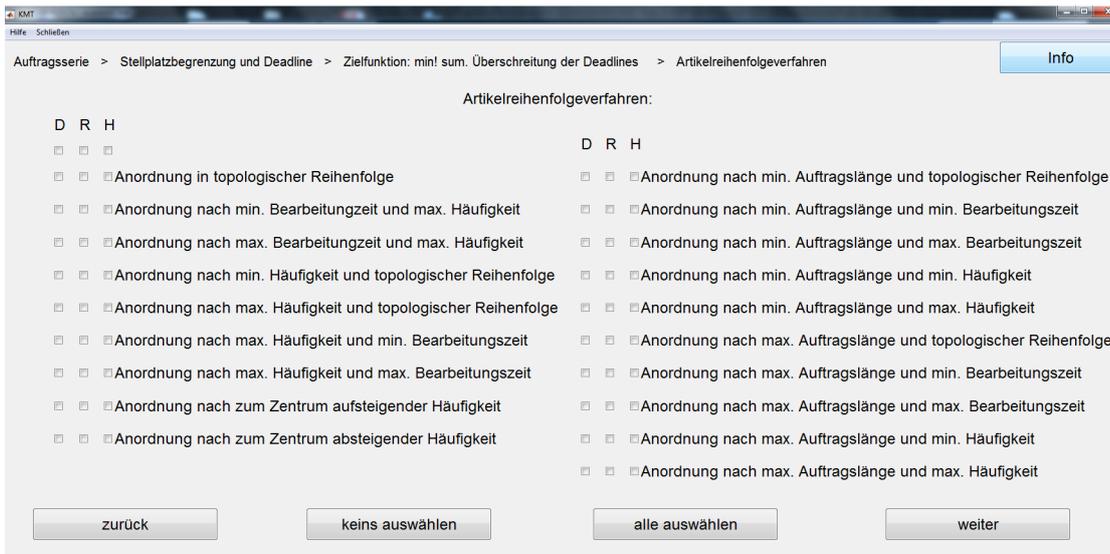


Abbildung 17: Auswahl der Artikelreihenfolgeverfahren mit Stellplatzbegrenzung

Ist ein Verfahren mit dynamischer Holstrategie ausgewählt, so wird durch das Anklicken des „weiter“-Buttons ein Dialogfenster geöffnet, in dem eingegeben werden kann, nach wie vielen fertigen Aufträgen die Auftragsreihenfolge neu berechnet werden soll. Für kleine Zuweisungsmatrizen W kann hier eine „1“ eingetragen werden. Da die Rechenzeit dynamischer Verfahren bei größeren Auftragslagen allerdings stark ansteigt, kann durch die Eingabe einer größeren Zahl $x \in \mathbb{N}$ festgelegt werden, dass nicht nach jedem neuen Auftrag, der in den Stellplatzbereich gestellt wird, die Artikelreihenfolge aktualisiert wird, sondern eben nur nach jedem x -ten, wodurch die Rechenzeit zu Lasten des Zielfunktionswertes erheblich verringert werden kann.

Danach kann gewählt werden, ob ein Offline- oder Online-Verfahren angewendet werden soll (vgl. Abb. 4). Beim Offline-Verfahren ist dem Programm zum Zeitpunkt $t = 0$ bereits die komplette Zuweisungsmatrix W bekannt. Beim Online-Verfahren hingegen wird dem Programm zu Anfang nicht die gesamte Zuweisungsmatrix enthüllt. Vielmehr wird diese dem Programm in zufälligen Zeitintervallen immer weiter enthüllt, sodass eine Online-Situation simuliert wird, bei der „im Laufe des Tages“ zusätzliche Aufträge eingehen.

Anschließend erscheint das Fenster zur Auswahl der Auftragsreihenfolgeverfahren. Abb. 18 zeigt die Wahl der Auftragsreihenfolgeverfahren. Über Checkboxen wird mindestens ein Auftragsreihenfolgeverfahren ausgewählt.

Auftragsreihenfolgeverfahren: Anordnung in topologischer Reihenfolge
 Anordnung nach aufsteigenden spez. Fertigstellungszeitpunkten
 Anordnung nach absteigenden spez. Fertigstellungszeitpunkten

Abbildung 18: Auswahl der Auftragsreihenfolgeverfahren

Nachdem die Berechnung der Artikel- und Auftragsreihenfolgeverfahren abgeschlossen ist, werden die Ergebnisse in einem neuen Fenster tabellarisch ausgegeben. Dieses Fenster ist mit einem Teil der Tabelle in Abb. 19 dargestellt. Der Unterschied zu Abb. 15 ist, dass in der letzten Spalte die ermittelte Auftragsreihenfolge dargestellt ist. Außerdem sind die einzelnen Zeilen zur besseren Übersicht eingefärbt worden. Ist eine Zeile **gelb** eingefärbt, sind die Ergebnisse in dieser Zeile durch eine hierarchische Holstrategie erzielt worden; **blau** steht für eine rollierende und **grün** für eine dynamische Holstrategie.

Verfahren	Zielfunktionswert/Abweichung (%)	Rechenzeit (s)	Artikelreihenfolge	Auftragsreihenfolge
Anordnung in topologischer Reihenfolge				
---Anordnung in topologischer Reihenfolge (Hierarchisch)	363.59 41.2878	0.0005	1 3 1 3 4 1 2 3 4 5 2 3 4 5	1 2 3 4 5 6 7
---Anordnung in topologischer Reihenfolge (Rollierend)	308.86 20.0202	0.0008	1 3 4 5 1 2 3 4 5 2 3 4	1 2 3 4 5 6 7
---Anordnung in topologischer Reihenfolge (Dynamisch)	363.59 41.2878	0.0018	1 3 1 3 4 1 2 3 4 5 2 3 4 5	1 2 3 4 5 6 7
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Hierarchisch)	313.71 21.9049	0.0006	3 3 4 5 1 3 4 2 5 2 5 3 4 2 1	1 2 3 4 5 6 7
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Rollierend)	315.29 22.5188	0.0008	3 4 5 1 3 4 2 5 1 3 4 2 5 3	1 2 3 4 5 6 7
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Dynamisch)	313.71 21.9049	0.0021	3 3 4 5 1 3 4 2 5 2 5 3 4 2 1	1 2 3 4 5 6 7
---Anordnung nach max. Bearbeitungszeit und max. Häufigkeit (Hierarchisch)	324.75 26.1949	0.0005	1 4 3 1 5 2 5 4 3 5 2 4 3	1 2 3 4 5 6 7
---Anordnung nach max. Bearbeitungszeit und max. Häufigkeit (Rollierend)	313.98 22.0098	0.0007	1 4 3 1 5 2 4 3 5 2 4 3	1 2 3 4 5 6 7
---Anordnung nach max. Bearbeitungszeit und max. Häufigkeit (Dynamisch)	305.24 18.6135	0.0016	1 4 3 1 2 5 4 3 5 2 4 3	1 2 3 4 5 6 7
---Anordnung nach min. Häufigkeit und topologischer Reihenfolge (Hierarchisch)	305.24 18.6135	0.0010	1 4 3 1 2 5 4 3 2 5 4 3	1 2 3 4 5 6 7
---Anordnung nach min. Häufigkeit und topologischer Reihenfolge (Rollierend)	305.24 18.6135	0.0006	1 4 3 1 2 5 4 3 2 5 4 3	1 2 3 4 5 6 7
---Anordnung nach min. Häufigkeit und topologischer Reihenfolge (Dynamisch)	341.16 32.5717	0.0013	1 4 3 2 3 4 5 1 2 3 4 5 2 3 4	1 2 3 4 5 6 7
Anordnung nach aufsteigenden spez. Fertigstellungszeitpunkten				
---Anordnung in topologischer Reihenfolge (Hierarchisch)	342.82 33.2168	0.0008	1 2 1 3 2 3 4 1 3 4 5 2 5	4 2 1 7 3 5 6
---Anordnung in topologischer Reihenfolge (Rollierend)	329.07 27.8736	0.0010	1 2 3 4 1 2 3 4 5 1 2 3 5	4 2 1 7 3 5 6
---Anordnung in topologischer Reihenfolge (Dynamisch)	342.82 33.2168	0.0016	1 2 1 3 2 3 4 1 3 4 5 2 5	4 2 1 7 3 5 6
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Hierarchisch)	257.34	0.0007	3 3 4 2 2 5 3 4 5 1 2 1	2 7 5 6 1 3 4
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Rollierend)	258.41 0.4158	0.0009	3 4 2 5 3 4 2 5 1 3 4 2 1	2 7 5 6 1 3 4
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Dynamisch)	257.34	0.0020	3 3 4 2 2 5 3 4 5 1 2 1	2 7 5 6 1 3 4
---Anordnung nach max. Bearbeitungszeit und max. Häufigkeit (Hierarchisch)	328.92 27.8153	0.0007	1 5 2 1 4 3 1 5 4 3 2 4 3	4 6 1 2 3 5 7
---Anordnung nach max. Bearbeitungszeit und max. Häufigkeit (Rollierend)	297.70 15.6835	0.0009	1 5 2 4 3 1 5 4 3 2 4 3	4 6 1 2 3 5 7

Abbildung 19: Ergebnisse der Artikel- und Auftragsreihenfolgeverfahren

Abschließend wird der rechte Pfad von Abb. 4 erläutert, also der Pfad mit Stellplatzbegrenzung und mit Berücksichtigung von Deadlines.

Für diese Problemstellungen werden die Zielfunktionen z_{11} bis z_{15} genutzt (vgl. Tab. 3). Eine dieser Zielfunktionen wird in einem Pop-Up-Menü ausgewählt. Das Pop-Up-Menü ist in Abb. 20 dargestellt.

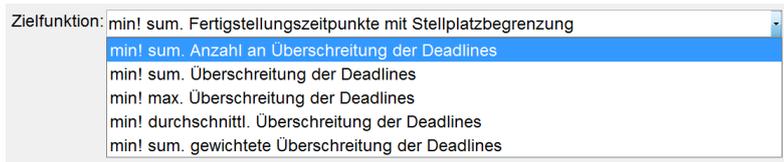


Abbildung 20: Auswahl der dealine-orientierten Zielfunktionen

Anschließend werden entsprechend Abb. 17 wieder beliebig viele der 19 Artikelreihenfolgeverfahren ausgewählt. Auch hier verfügt jedes Verfahren über drei Checkboxes, die unabhängig voneinander ausgewählt werden können und für eine hierarchische, rollierende bzw. dynamische Artikelholstrategie stehen. Wurde eine dynamische Strategie ausgewählt, muss anschließend wieder festgelegt werden, nach wie vielen fertigen Aufträgen die Auftragsreihenfolge neu berechnet werden soll.

Dann muss das Auftragsreihenfolgeverfahren ausgewählt werden. Das Auswahlfenster ist in Abb. 21 zu sehen. Über Checkboxes wird mindestens ein Auftragsreihenfolgeverfahren ausgewählt. Die Auswahl muss durch drücken auf „OK“ bestätigt werden. Dadurch wird der untere Teil des Auswahlfensters aktiv. Die hier aufgelisteten Verfahren geben an, wie mit Aufträgen verfahren wird, denen keine Deadline zugeordnet ist, die also eine unendlich große Deadline aufweisen. Auch hier muss mindestens ein Verfahren gewählt werden.

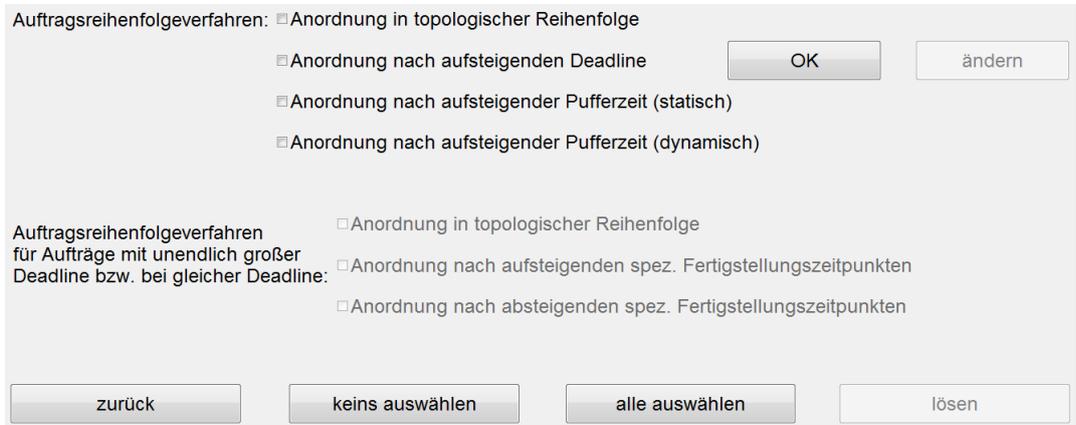


Abbildung 21: Auswahl von deadline-orientierten Auftragsreihenfolgestrategien

Eine Mehrfachauswahl im unteren Teil des Fensters führt dazu, dass jedes der im oberen Teil ausgewählten Auftragsreihenfolgeverfahren je einmal mit jedem im unteren Teil ausgewählten Verfahren kombiniert wird. Ausgenommen ist die „Anordnung in topologischer Reihenfolge“. Anschließend kann durch einen Klick auf „lösen“ die Berechnung gestartet werden.

Die Ergebnisse werden in einer Tabelle ähnlich Abbildung 19 dargestellt. Durch einen Klick auf den Button „verbessern“ können die gefundenen Zielfunktionswerte weiter

verbessert werden. Dazu müssen in dem folgenden Fenster zuerst die Auftragsreihenfolgeverfahren ausgewählt werden, dessen Ergebnisse verbessert werden sollen. Es stehen alle in Abb. 21 ausgewählten Kombinationen zur Auswahl. Anschließend muss die Artikelreihenfolge gewählt werden, die verbessert werden soll. Auch in diesem Fenster stehen nur die Artikelreihenfolgeverfahren (mit ihrer gewählten Holstrategie: hierarchisch, rollierend oder dynamisch) zur Auswahl, die zuvor berechnet wurden. Abschließend muss das Verbesserungsverfahren ausgewählt werden (siehe Abb. 22). Auch können wieder zusätzliche Abbruchbedingungen festgelegt werden.

Verbesserungsverfahren:

Insert

2-Tausch Lite

Mit zusätzlichen Abbruchbedingungen:

1. Rechenzeit von Minuten

2. Zielfunktionswertverbesserung ausgehend vom Artikelreihenfolgeverfahren um %

3. Zielfunktionswertverbesserung von weniger als % in den letzten Vergleichen

Abbildung 22: Auswahl der Verbesserungsverfahren mit Berücksichtigung von Deadlines

Durch einen Klick auf den „lösen“-Button wird die Berechnung gestartet. Die Ergebnisdarstellung entspricht Abb. 15, allerdings erweitert um die eingeführten Einfärbungen der einzelnen Zeilen. Außerdem muss nun nicht nur zwischen den einzelnen zu verbessernden Artikel-, sondern auch zwischen den Auftragsreihenfolgeverfahren unterschieden werden. Hierzu werden für jedes zu verbessernde Auftragsreihenfolgeverfahren (dargestellt durch eine weiße Zeile) die Ergebnisse aller ausgewählten zu verbessernden Artikelreihenfolgeverfahren aufgeführt. Die Ergebnisdarstellung ist in Abb. 23 zu erkennen.

Info

Auftragsserie > Stellplatzbegrenzung und Deadline > Zielfunktion: min! sum. Überschreitung der Deadlines > Artikelreihenfolgeverfahren > Lösung > Verbesserung
 Verbesserungsverfahren > Lösung

Verfahren	Zielfunktionswert	Abweichung (%)	Absolute Verbesserung	Relative Verbesserung (%)	Anz. Vergleiche bis Best.	Anz. Verbes.	Rechenzeit (s)	Artikelreihenfolge	Auftragsreihenfolge
Anordnung in topologischer Reihenfolge									
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Hierarch...	29.09	-					0.0013	3 3 4 5 1...	1 2 3 4 5 6 7
-----Insert	0.00	0	29.0900	100	4	3	0.0024	3 3 4 2 3...	1 2 7 3 4 5 6
-----2-Tausch Lite	0.00	0	29.0900	100	2	2	0.0015	3 4 2 3 4...	1 7 3 4 2 6 5
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Rolliere...	33.86	-					0.0017	3 4 5 1 3...	1 2 3 4 5 6 7
-----Insert	0.18	-	33.6800	99.4684	5	3	0.0039	3 4 5 1 3...	1 2 3 7 4 5 6
-----2-Tausch Lite	0.00	0	33.8600	100	2	2	0.0021	3 4 2 5 1...	1 7 3 4 2 6 5
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Dynam...	29.09	-					0.0040	3 3 4 5 1...	1 2 3 4 5 6 7
-----Insert	0.00	0	29.0900	100	4	3	0.0057	3 3 4 2 3...	1 2 7 3 4 5 6
-----2-Tausch Lite	0.00	0	29.0900	100	2	2	0.0039	3 4 2 3 4...	1 7 3 4 2 6 5
---Anordnung nach max. Bearbeitungszeit und max. Häufigkeit (Hierarc...	37.34	-					9.1795e-04	1 4 3 1 5...	1 2 3 4 5 6 7
-----Insert	0.00	0	37.3400	100	4	3	0.0023	1 5 4 3 2...	1 3 2 7 4 5 6
-----2-Tausch Lite	0.00	0	37.3400	100	3	3	0.0019	1 5 2 4 3...	3 7 1 4 2 6 5
Anordnung nach aufsteigenden Deadline und in topologischer Reihen...									
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Hierarch...	0.00	0					8.7445e-04	3 4 2 2 5...	7 3 4 2 1 6 5
-----Insert	0.00	0	0.0000	0	0	0	5.3460e-04	3 4 2 2 5...	7 3 4 2 1 6 5
-----2-Tausch Lite	0.00	0	0.0000	0	0	0	7.0014e-04	3 4 2 2 5...	7 3 4 2 1 6 5
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Rolliere...	0.00	0					0.0012	3 4 2 5 1...	7 3 4 2 1 6 5
-----Insert	0.00	0	0.0000	0	0	0	8.0525e-04	3 4 2 5 1...	7 3 4 2 1 6 5
-----2-Tausch Lite	0.00	0	0.0000	0	0	0	7.7576e-04	3 4 2 5 1...	7 3 4 2 1 6 5
---Anordnung nach min. Bearbeitungszeit und max. Häufigkeit (Dynam...	0.00	0					0.0030	3 4 5 2 2...	7 3 4 2 1 6 5
-----Insert	0.00	0	0.0000	0	0	0	0.0013	3 4 5 2 2...	7 3 4 2 1 6 5

zurück neustart speichern speichern klein schließen

Abbildung 23: Ergebnisse der Verbesserungsverfahren mit Berücksichtigung von Deadlines

5 Fazit und Ausblick

Die Lösungsgüte der einzelnen Artikel- bzw. Auftragsreihenfolgestrategien ist auf kanonische Weise von der jeweils betrachteten Problemausprägung sowie von der konkreten Datenkonstellation abhängig. Daher ist es schwierig vorherzusagen, welche Reihenfolgestrategien im vorliegenden Einzelfall zu bevorzugen sind. Vielmehr ist das vorgestellte KMT-Programm ja genau zum Austesten bei individuellen Problemstellungen entwickelt worden. Somit werden an dieser Stelle auch keine ausführlichen Analyseergebnisse aus einzelnen Untersuchungen vorgestellt, sondern lediglich einige allgemeine Aussagen über bisherige Erfahrungen zusammengefasst, ohne letztendlich damit unumstößliche Allgemeingültigkeiten vorzulegen.

- Es gibt nicht „das beste“ Verfahren. Allerdings lassen sich tendenziell effiziente Verfahren finden; z.B. sind MinA*-Strategien bei Problemstellungen mit und ohne Stellplatzbegrenzungen tendenziell besser als andere Strategien, und zwar in Bezug auf nahezu alle betrachteten Kenngrößen.
- Die gewählte Artikel-Holstrategie hat großen Einfluss auf den Zielfunktionswert und die Rechenzeit. Eine rollierende Holstrategie führt in der Regel zu den besseren Ergebnissen, wenn Aufträge mit maximalen Fertigstellungszeitpunkten berücksichtigt werden.
- Die untersuchten Verbesserungsverfahren beanspruchen schon bei mittleren Problemgrößen enorme Rechenzeiten, bewirken in vielen Fällen dabei aber nur geringe Verbesserungen der Lösungsgüte (relativ geringe Effizienz). Daher sollte insbesondere auf die weitere Entwicklung von effizienten Konstruktionsverfahren geachtet werden. Hierzu könnten verschiedene bionische Verfahren geeignet sein (z.B. genetische Algorithmen, Ameisen-, Bienen-, Fledermäuse-Algorithmen u.v.a.m.).

Um die Weiterentwicklung des Programms KMT zu fördern und auch Interessierten außerhalb des Entwicklerkreises die Nutzung des Programms zu ermöglichen, wird der komplette Source-Code von KMT samt Dokumentationen via Internet-Link öffentlich zur Verfügung gestellt: <https://github.com/tlask/KMT>.

An dieser Stelle sei nochmals allen Absolvent*inn*en und Kommiliton*inn*en gedankt, die im Rahmen ihrer Qualifikationsarbeiten am KMT-Projekt mitgewirkt haben.

Literaturverzeichnis

- [1] Bator M: *Entwicklung und Implementierung von heuristischen Verfahren zur Bestimmung von effizienten Artikelreihenfolgen bei einem Ware-zum-Mann-Kommissionierproblem*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2010
- [2] Bichler K, Krohn R, Riedel G, Schöppach F: *Beschaffungs- und Lagerwirtschaft. Praxisorientierte Darstellung der Grundlagen, Technologien und Verfahren*. 9. Auflage, Gabler Verlag, 2010
- [3] Cilgin S, Ala R, Gola R: *Master-Projekt Kommissionierungstool*. Projektarbeit im Master-Studiengang Optimierung und Simulation, Fachhochschule Bielefeld, 2017
- [4] Gudehus T: *Logistik. Grundlagen – Strategien – Anwendungen*. 4. Auflage, Springer-Verlag, 2010
- [5] Kleine-Döpke T: *KMT Anwender-Handbuch*. Projektarbeit im Master-Studiengang Optimierung und Simulation, Fachhochschule Bielefeld, 2014
- [6] Kroos M: *Entwicklung von effizienten Reihenfolgestrategien zur Kostenminimierung bei einem Ware-zum-Mann-Kommissionierproblem*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2013
- [7] Large R: *Logistikfunktionen. Betriebswirtschaftliche Logistik*. Band 1, Oldenbourg Verlag, 2012
- [8] Littau I: *Entwicklung von effizienten Artikelreihenfolgestrategien zur Minimierung der maximalen unproduktiven Auftragsbearbeitungszeit bei einem Ware-zum-Mann-Kommissionierproblem*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2014
- [9] Lye TF, Lask T: *KMT-Anwenderhandbuch mit Beispielrechnungen*. Fachbereich Ingenieurwissenschaften und Mathematik der Fachhochschule Bielefeld, 2016
- [10] Lye TF: *Entwicklung eines Optimierungstools für eine Klasse von Ware-zum-Mensch-Kommissionierungsproblemen*. Masterarbeit im Studiengang Optimierung und Simulation, Fachhochschule Bielefeld, 2015

- [11] Lye TF, Kruse H-J, Lask T: *Heuristische Lösungsmethoden für eine Klasse von Ware-zum-Mensch-Kommissionierungsproblemen*. Fachhochschule Bielefeld, AMMO-Berichte aus Forschung und Technologietransfer, Heft Nr. 7, März 2016.
- [12] Martin H: *Transport- und Lagerlogistik. Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik*. 9. Auflage, Springer Vieweg, 2014
- [13] Matis V: *Entwicklung von Reihenfolgestrategien zur Bestimmung einer effizienten Anzahl von Stellplätzen bei einem Ware-zum-Mann-Kommissionierprozess*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2012
- [14] Nehab D: *Weiterentwicklung und Implementierung eines Optimierungstools für WzP-Kommissionierungsprobleme unter Berücksichtigung von Fälligkeitsterminen*. Masterarbeit im Studiengang Optimierung und Simulation, Fachhochschule Bielefeld, 2017
- [15] Pulverich M, Schietinger J: *Handbuch Kommissionierung. Effizient picken und packen*. 1. Auflage, Verlag Heinrich Vogel, 2009
- [16] Schreiber A: *Entwicklung von effizienten Artikelreihenfolgestrategien zur Minimierung von unproduktiven Zeiten bei einem Ware-zum-Mann-Kommissionierproblem*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2013
- [17] ten Hompel M, Sadowsky V, Beck M: *Kommissionierung. Materialflusssysteme 2 – Planung und Berechnung der Kommissionierung in der Logistik*. Reihe: Intralogistik, Springer-Verlag, 2011
- [18] Verein Deutscher Ingenieure (VDI): *VDI-Richtlinie 3590 Blatt 1 – Kommissioniersysteme – Grundlagen*. VDI-Handbuch Technische Logistik, Band 8, Beuth-Verlag, 1994
- [19] Weiz K: *Entwicklung von heuristischen Verfahren zur Bestimmung von effizienten Auftragsreihenfolgen bei einem Ware-zum-Mann-Kommissionierproblem mit begrenzter Anzahl von Kommissionierplätzen*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2011
- [20] Wolkow S: *Effizienzvergleich von Lösungsstrategien für Ware-zum-Mensch-Kommissionierungsprobleme mit Stellplatzbegrenzung*. Bachelorarbeit im Studiengang Angewandte Mathematik, Fachhochschule Bielefeld, 2017

Kontakt Daten

Autoren

David Nehab, M. Sc.

dnehab@gmx.de

Prof. Dr. rer. pol. Hermann-Josef Kruse

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

Interaktion 1

33619 Bielefeld

Telefon +49.521.106-7411

Telefax +49.521.106-7176

hermann-josef.kruse@fh-bielefeld.de

Raum D 229

Timo Lask, M. Sc.

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

Interaktion 1

33619 Bielefeld

Telefon +49.521.106-7403

timo.lask@fh-bielefeld.de

Raum A 504

FSP AMMO

Sprecherin

Prof. Dr. rer. nat. Svetozara Petrova

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

FSP Angewandte Mathematische Modellierung und Optimierung

Interaktion 1

33619 Bielefeld

Telefon +49.521.106-7410

Telefax +49.521.106-7190

svetozara.petrova@fh-bielefeld.de

Raum D 226

Stellv. Sprecherin

Dr. rer. nat. Sabrina Proß

Fachhochschule Bielefeld

Fachbereich Ingenieurwissenschaften und Mathematik

FSP Angewandte Mathematische Modellierung und Optimierung

Schulstraße 10

33330 Gütersloh

Telefon +49.5241.21143-21

sabrina.pross@fh-bielefeld.de

Raum 303

Veröffentlichungsreihe:

AMMO – Berichte aus Forschung und Technologietransfer

- Heft 8:** J. Silberberg, T. Lask, B. Bachmann: *Formalismen für gefärbte Petri-Netze und Verfahren zur effizienten Bestimmung von aktiven Modus-Mengen*. Juli 2016.
- Heft 7:** T.F. Lye, H.-J. Kruse, T. Lask: *Heuristische Lösungsmethoden für eine Klasse von Ware-zum-Mensch-Kommissionierungsprobleme*. März 2016.
- Heft 6:** T. Kleine-Döpke, H.-J. Kruse: *Lösungsansätze für Konfliktsituationen bei Feuerprozessen in kapazitierten Petri-Netzen*. Juni 2015.
- Heft 5:** H.-J. Kruse: *Optimumgraphen*. Oktober 2014.
- Heft 4:** S. Proß: *Diskrete Modellierung und Optimierung praxisrelevanter Prozesse mit Petri-Netzen*. September 2014.
- Heft 3:** R. Ueckerdt, H.-W. Schmidt, M. Weber, E. Mindlina: *Entwicklung einer Dispatcherfunktion zur Überprüfung von Nominierungsmengen in der Betriebsführung von Erdgasspeichern*. Juli 2014.
- Heft 2:** R. Walden und V.-M. Roemer: *Methoden der quantitativen rechnergestützten CTG-Analyse*. April 2014.
- Heft 1:** AMMO-Team: *Informationen über den Forschungs- und Entwicklungsschwerpunkt Angewandte Mathematische Modellierung und Optimierung*. Dezember 2013.

ISSN 2198-4824

Herausgeber:

Vorstand von FSP AMMO

Fachhochschule Bielefeld