



Entwicklung und Implementierung eines Routingalgorithmus für innerbetriebliche Transportsysteme mit verschiedenen Transportmitteltypen

Development and Implementation of a Routing Algorithm for Heterogeneous Inbound Transport Systems

Semesterarbeit

an der Fakultät für Maschinenwesen der Technischen Universität München

Themenstellender	Prof. Dr.-Ing. Johannes Fottner Lehrstuhl für Fördertechnik Materialfluss Logistik
Betreuer	Mestiri Slaheddine, M.Sc.
Eingereicht von:	Fabian Küster Perlacher Forstweg 18 82008, Unterhaching +49 173-3824823
Eingereicht am:	01.01.2022 in Garching
Inventarnr. fml:	2021067

Vorwort

Die vorliegende Arbeit entstand unter der wissenschaftlichen und inhaltlichen Anleitung von Mestiri Slaheddine, M. Sc., wissenschaftlicher Mitarbeiter am Lehrstuhl für Fördertechnik Materialfluss Logistik (fml) der Technischen Universität München.

Vereinbarung zum Urheberrecht

Hiermit gestatte ich dem Lehrstuhl für Fördertechnik Materialfluss Logistik diese Studienarbeit bzw. Teile davon nach eigenem Ermessen an Dritte weiterzugeben, zu veröffentlichen oder anderweitig zu nutzen. Mein persönliches Urheberrecht ist über diese Regelung hinaus nicht beeinträchtigt. Eventuelle Geheimhaltungsvereinbarungen über den Inhalt der Arbeit zwischen mir bzw. dem Lehrstuhl für Fördertechnik Materialfluss Logistik und Dritten bleiben von dieser Vereinbarung unberührt.

Ort, Datum, Unterschrift

Inhaltsverzeichnis

Inhaltsverzeichnis	I
1 Motivation	1
1.1 Problemstellung	1
1.2 Zielsetzung	1
1.3 Lösungsansatz	2
2 Optimierungsprobleme	3
2.1 Einführung in die Logistik	3
2.1.1 Schlanke Logistik	4
2.1.2 Produktionslogistik	5
2.2 Formulierung von Optimierungsproblemen	6
2.3 Komplexität	7
2.4 Graphentheorie	8
2.5 Relevante Optimierungsprobleme	9
2.5.1 Kürzeste-Wege-Problem	10
2.5.2 Das Problem des Handlungsreisenden	11
2.5.3 Tourenplanungsprobleme	13
2.5.4 Zuweisungsprobleme	14
2.6 Lösungsverfahren	16
2.6.1 Exakte Methoden	16
2.6.2 Heuristische Methoden	17
2.6.3 Metaheuristische Methoden	17
3 Algorithmus	19
3.1 Zuweisungsproblem	19
3.1.1 Ungarische Methode	20
3.2 Tourenplanungsproblem	23
3.2.1 k -opt Heuristik	23
4 Implementierung	27
4.1 Plant Simulation	27

4.2	Implementierung in das Simulationsprogramm	28
5	Evaluierung	35
5.1	Versuchsauswertung	38
5.1.1	Vermehrt Direkttransport	38
5.1.2	Vermehrter Einsatz von Routenzügen	43
5.2	Kritische Bewertung	49
6	Ausblick	51
	Literaturverzeichnis	53
	Abbildungsverzeichnis	57
	Tabellenverzeichnis	59
A	Anhang	A-1

1 Motivation

1.1 Problemstellung

Aufgrund des ständig steigenden Verlangens nach höherer Produktvielfalt und der daraus resultierenden benötigten Flexibilität in der Produktion sind eine effiziente Steuerung sowie schlanke Prozesse für eine Wettbewerbsfähigkeit unumgänglich. Zudem erschweren die Planungsunsicherheiten, aufgrund von Nachfrageschwankungen die Planung der Produktion. [Mär-2011]

All diese Faktoren spiegeln sich unter anderem in den Logistikkosten eines Unternehmens wieder. Hierbei sollte zwischen, auf den Handel spezialisierte Unternehmen, sowie produzierende Unternehmen unterschieden werden. Es leuchtet ein, dass erstere einen höheren Anteil an Logistikkosten aufweisen, zwischen 20 und 30 Prozent, da diese auf den Umschlag von Waren spezialisiert sind. Doch auch produzierende Unternehmen haben für gewöhnlich mit Logistikkosten zwischen 10 und 20 Prozent zu kämpfen. [See-2010]

1.2 Zielsetzung

Oftmals fällt eine Aufteilung der Logistikkosten nicht leicht, da sich diese nicht immer problemlos abgrenzen lassen; eine sinnvolle Einteilung dieser Kosten eines produzierenden Unternehmens ist die Folgende [See-2010]:

- System- und Steuerungskosten
- Lager- und Handlingskosten
- Bereitstellungskosten
- Transportkosten
- Bestandskosten

Im Zuge dieser studentischen Arbeit soll ein Zuweisungs- und Routing-Algorithmus, welcher möglichst optimal Logistikaufträge steuert, entwickelt und in einem generischen Simulationsprogramm implementiert werden. Ziel ist es hierbei, die Transporteinheiten aller eingehenden Aufträge auf die unterschiedlichen Transportmittel zu verteilen und diese

durch das Werk zu leiten, um dadurch die Bereitstellungskosten zu minimieren. Logistische Restriktionen für einen solchen Algorithmus sind beispielsweise die Eigenschaften der verschiedenen Transportmittel, der Zeitpunkt des Auftragseingangs oder auch das vorherrschende Layout. Die Anforderungen an das Programm an sich, welche möglicherweise einen Zielkonflikt darstellen, sind die Rechengeschwindigkeit, sowie die Abweichung von der optimalen Lösung.

1.3 Lösungsansatz

Der zu entwickelnde Algorithmus lässt sich in zwei Bereiche aufteilen. Zum einen soll dieser die eingehenden Aufträge kostenoptimal an die zu diesem Zeitpunkt verfügbaren Transportmittel verteilen. Zum anderen sollen diese Fahrzeuge sich durch eine Optimierung der Routen, möglichst effizient durch das vorherrschende Layout bewegen.

Um im Nachgang die Funktionsweise dieses Algorithmus evaluieren zu können, soll dieser in einem Simulationsprogramm implementiert werden. In einer solchen Simulationsumgebung kann durch Variation der Eingangsgrößen, die Funktionsweise, die Performance sowie die Unterschiede zu anderen Algorithmen bewertet werden. Einzustellende Variablen sind beispielsweise die Auftragsrate, die Anzahl und Typen von Transportmitteln sowie das Layout.

2 Optimierungsprobleme

Auf sogenannte Optimierungsprobleme stößt man nicht etwa nur in der Industrie auf der Suche nach dem effizientesten Produktionssystem, vielmehr findet man sich selbst im Alltag immer wieder in Situationen in denen man solche Probleme versucht zu lösen.

Beispielsweise wenn mit einem Auto verschiedenen Personen von unterschiedlichen Orten abgeholt werden müssen. Anstatt sich ohne vorherige Überlegungen in das Auto zu setzen und einen nach dem anderen abzufahren, ist es doch sinnvoller, vorab eine möglichst optimale Route zu finden. Unter optimal ist in diesem Beispiel, die schnellste oder spritsparendste Route zu verstehen.

Vor Problemen wie diesem, in größerem Ausmaß und von größerer wirtschaftlicher Bedeutung, steht man in der logistischen Planung eines Unternehmens.

2.1 Einführung in die Logistik

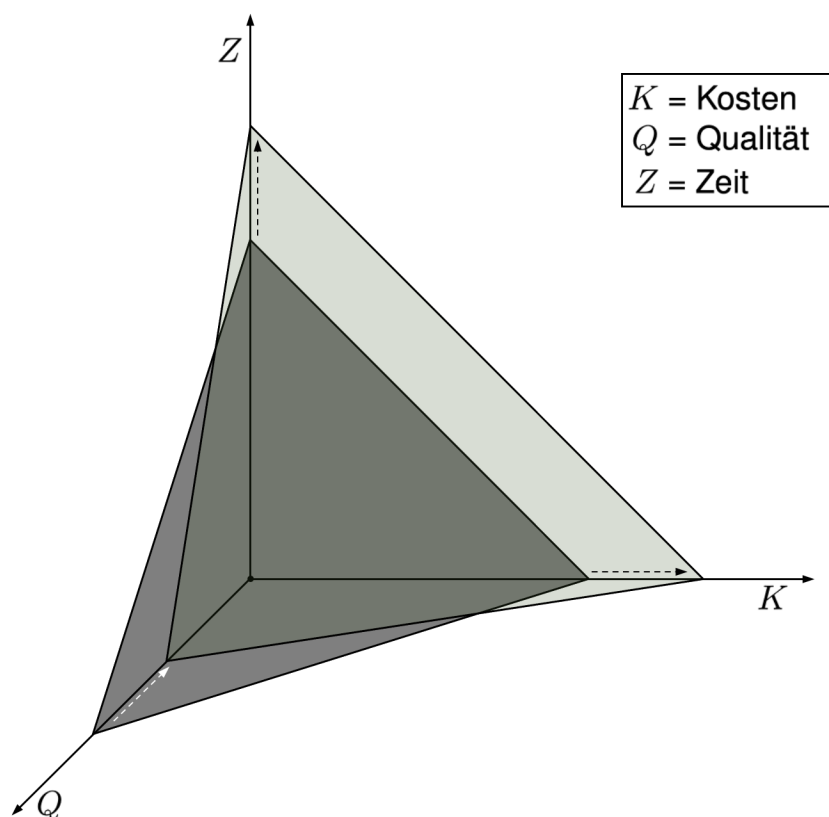


Abbildung 2-1: Magisches Dreieck der Logistik

In der Logistik, unabhängig des Bereichs, existiert ein Zielkonflikt zwischen der benötigten Zeit für die Herstellung eines Produkts, beziehungsweise der Bereitstellung einer Dienstleistung, den hierbei anfallenden Kosten sowie der Qualität. Es ist nicht möglich alle diese Ziele simultan zu optimieren. In der Literatur wird dieses Phänomen oft mittels des sogenannten Magischen Dreiecks der Logistik illustriert [Wen-2000]. In Abbildung 2-1 ist dies anhand eines simplen Beispiels dargestellt. Das Optimieren, also das Senken der Kosten und zugleich der benötigten Zeit hat meist einen Qualitätsverlust zur Folge (schwarzes, zu grünem Dreieck). Ein weiteres Beispiel wäre die Erhöhung des Bestandes beziehungsweise ein allgemein hoher Lagerbestand, wodurch die termingerechte Lieferung der Materialien an die einzelnen Arbeitsstationen gewährleistet wird und die daraus resultierende Termintreue, also die Qualität. Hohe Bestände sind jedoch mit hohen Kosten verbunden, was auch hier einen Zielkonflikt zwischen Qualität, Zeit und den Kosten darstellt.

2.1.1 Schlanke Logistik

Das Prinzip der Schlanken Logistik (engl. Lean Logistics) ist längst keine Neuheit mehr, auch wenn es trotz der vielen Vorteile, noch nicht flächendeckend umgesetzt wird [Gün-2014]. Der Grundgedanke der Schlanken Logistik ist eine kontinuierliche Prozessverbesserung und die effiziente Gestaltung der gesamten Wertschöpfungskette. Letzteres setzt voraus, dass alle nicht wertschöpfenden Prozesse minimiert oder gar vermieden werden. Das Toyota-Produktionssystem gilt als der Ursprung und Benchmark dieses Prinzips [Kit-2017]. Viele heute verwendete Konzepte entstammen diesem Produktionssystem. So etwa das Prinzip der 3-M's: *Muri*, *Mura*, *Muda*. *Muri* entspricht der Überlastung, sowohl von Mensch als auch Maschine, *Mura* der Fluktuation/Unausgeglichenheit und *Muda* der Verschwendung. Ziel ist es alle drei zu minimieren oder gar zu vermeiden. Oftmals führt die Eliminierung einer der drei M's auch zu einer Eliminierung der verbleibenden.

Ein weiteres Konzept welches abzielt die *Mudas* zu eliminieren ist die Unterteilung der Verschwendungen. Grundsätzlich gilt all das was Ressourcen verbraucht, der Kunde jedoch nicht bereit ist zu bezahlen als Verschwendung. In dem Toyota-Produktionssystem existieren folglich 7 Arten von Verschwendungen [Wil-2013]. Dargestellt in Tabelle 2-1. All diese Verschwendungen existieren in unterschiedlichem Ausmaß in den einzelnen Logistikbereichen. In Abbildung 2-2 sind die verschiedenen Logistikbereiche eines produzierenden

Tabelle 2-1: Die 7 Arten von Verschwendungen

	Art der Verschwendung	Vorkommen
1	Transport	Zwischen einzelnen Abteilungen
2	Bestände	Lagerkosten
3	Unnötige Tätigkeiten	Überflüssige Bewegungen von Arbeitern
4	Wartezeiten	Benötigtes Material an Maschine nicht verfügbar
5	Überproduktion	Es wird mehr produziert als verkauft werden kann
6	Prozessübererfüllung	Unnötig enge Toleranzen
7	Ausschuss	Fehlerhafte Teile

Unternehmens aufgezeigt [Kri-2019]. Wie zu Beginn erwähnt zielt diese Arbeit darauf ab, den innerbetrieblichen Transport zu optimieren. Der Fokus liegt daher auf der Produktionslogistik und der darin enthaltenen Steuerung der Transporte und Planung der Aufträge.

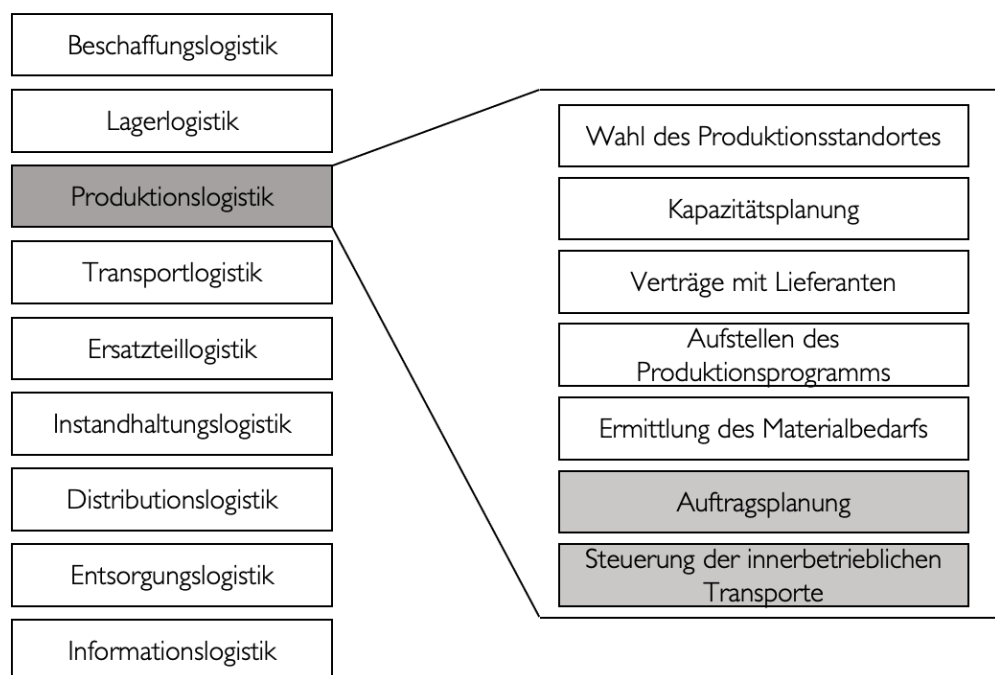


Abbildung 2-2: Einteilung der Logistikbereiche

2.1.2 Produktionslogistik

Die Produktionslogistik wirkt als Bindeglied zwischen der Beschaffungslogistik und der Distributionslogistik. Je nach Betriebsstruktur wird die Lagerlogistik von diesen mit einge-

schlossen oder extra genannt. Der Umfang der Produktionslogistik ist ebenfalls in Abbildung 2-2 dargestellt. Ziel der Produktionslogistik ist es daher die Produktion so effizient, beziehungsweise so optimal wie möglich zu gestalten.

Für diese Arbeit relevante Kenngrößen der Produktionslogistik und zur Beurteilung deren Optimalität sind vor allem die Durchlaufzeiten, Auftragserfüllung sowie die Auslastung des Produktionssystems. Die Durchlaufzeit eines Systems beschreibt die Zeitspanne für die Abarbeitung von Aufträgen bzw. Prozessen. Sie setzt sich zusammen aus der Bearbeitungszeit, der Rüstzeit, der Liegezeit, der Transportzeit und der Lagerungszeit [Hei-1979]. Offensichtlich kann durch die Optimierung der Transporte die Durchlaufzeit reduziert werden. Die Auslastung beschreibt das Verhältnis der tatsächlichen Output-Rate mit der theoretisch möglichen, also der maximalen Output-Rate. Rein betriebswirtschaftlich gesehen ist eine höhere Auslastung gleichbedeutend mit geringeren Stückkosten.

Dies ist jedoch nur eine Seite der Medaille, denn eine Auslastung nahe der maximalen Kapazität ist sehr anfällig für Störungen. Grundsätzlich ist eine hohe Auslastung natürlich besser, denn ungenutzte Wirtschaftsgüter bedeuten Kosten und sind als Verschwendung zu minimieren beziehungsweise zu vermeiden. Der optimale Auslastungsgrad jedoch ist von Unternehmen zu Unternehmen unterschiedlich und hängt von vielen Faktoren ab.

Für die Bewertung beziehungsweise die Validierung eines Assignment-, Routingalgorithmus ist die Auslastung, beispielsweise der Fahrzeuge allerdings eine sehr gute Kennzahl.

2.2 Formulierung von Optimierungsproblemen

Das Ziel der Optimierung ist das Ermitteln der optimalen Lösung eines Problems unter bestimmten Randbedingungen. Mathematisch entspricht dies der Suche eines Funktionswertes welcher die Optimierungsfunktion, je nach Formulierung des Problems maximiert beziehungsweise minimiert. Aufgrund des Dualitätsprinzips kann ein Maximierungsproblem problemlos in ein Minimierungsproblem umgewandelt werden und vis versa [Gri-2018]:

$$\max f = - \min(-f) \quad \text{bzw.} \quad \min f = - \max(-f)$$

Die allgemeine Formulierung eines Optimierungsproblem lässt sich wie folgt darstellen [Ger-2011, S. 15]:

Maximiere die Zielfunktion

$$\max z = F(x)$$

unter den Nebenbedingungen

$$g_i(x) = \begin{cases} \leq \\ = \\ \geq \end{cases} 0 \quad i = 1, \dots, m.$$

Hierbei entspricht x der Entscheidungsvariable. Je nach Problemstellung können für x bestimmte Einschränkungen, wie beispielsweise eine Nichtnegativitätsbedingung, eine Ganzzahligkeitsbedingung oder etwa eine Binärbedingung, formuliert werden.

2.3 Komplexität

Die Komplexität eines Algorithmus beziehungsweise der Aufwand um ein algorithmisches Problem zu lösen kann in zwei Arten eingeteilt werden. Zum einen der benötigte Speicherplatz und damit die *Space-Complexity* und zum anderen die Anzahl an Elementaraufgaben die der Algorithmus ausführen muss um das Problem zu lösen, auch *Time-Complexity* genannt. Für gewöhnlich ist jedoch die genaue Anzahl an durchzuführenden Operationen weniger interessant, als die Relation zwischen der Dimension des Inputs und der *time-Complexity*. Hierfür wird die *big-O notation* verwendet.

Ein linearer Zusammenhang würde beispielsweise durch $\mathcal{O}(n)$ dargestellt. Abhängig von dieser Komplexität werden die algorithmischen Probleme in Klassen eingeteilt. Vergleichbar mit einer Skala von *leicht* bis *schwer* existieren hierfür die Klassen *P*, *NP*, *NP-Complete* und *NP-Hard*. Zu den *P-Problemen* zählen alle, welche sich in *polynomieller* Zeit lösen lassen. *NP-Probleme* hingegen, benötigen *nicht-polynomielle* Zeit um das gewünschte Ergebnis zu liefern, jedoch eine *polynomielle* Zeit um diese Lösung zu verifizieren. Eine weitere Kategorie sind die *NP-Hard-Probleme*. Ein solches Problem zeichnet sich dadurch aus, dass es mindestens so schwer zu lösen ist, wie die schwierigsten *NP-Probleme*.

Ein Algorithmus um *NP-Hard-Probleme* zu lösen könnte reduziert werden um jedes *NP-Problem* zu lösen. Wenn ein Problem sowohl *NP* als auch *NP-Hard* ist, so nennt man

dieses *NP-Complete*. [Ner-2019]

Graphisch dargestellt ist diese Einteilung in 2-3. Die Linie stellt hierbei die Menge aller existierenden algorithmischen Probleme dar.

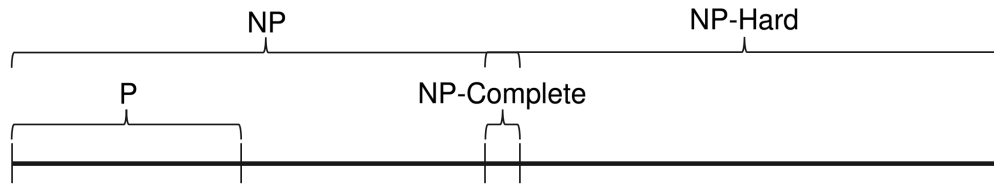


Abbildung 2-3: Komplexitätsklassen in Anlehnung an [Ner-2019, S. 426]

2.4 Graphentheorie

Da für die Optimierungsprobleme grundlegende Kenntnisse der Graphentheorie benötigt werden, soll im Folgenden kurz hierauf eingegangen werden.

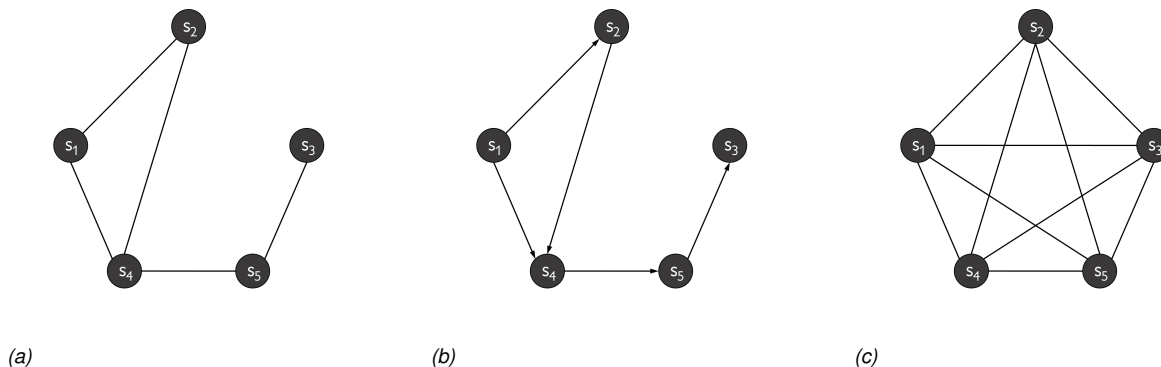


Abbildung 2-4: Beispiele für Graphen: (a) ungerichtet, (b) gerichtet, (c) vollständig

Abbildung 2-4 a) zeigt den allgemeinen Aufbau eines Graphen. Die Punkte, s_1 bis s_5 , werden als Knoten, manchmal auch Ecken bezeichnet. Die Verbindung zweier Knoten wird als Kante bezeichnet. Geben diese Kanten keine Richtung vor, wie in Abbildung 2-4 a), dann wird der Graph als *ungerichtet* bezeichnet. Werden hingegen, wie in Abbildung 2-4 b) Pfeile als Kanten verwendet und somit eine Richtung vorgegeben, dann bezeichnet man den Graph als *gerichteter Graph*. Der letzte dargestellte Graph in Abbildung 2-4 zeigt einen Spezialfall, nämlich, dass jeder Knoten mit jedem verbunden ist beziehungsweise dass zwischen allen Knoten eine Kante existiert, gerichtet oder ungerichtet. Diese Art Graph wird mit *vollständig* betitelt. [Tur-2015, S. 19 ff.]

Für die Darstellung der Optimierungsprobleme in der Logistik sind die Graphen in Abbildung 2-4 jedoch nicht detailliert genug. Je nach Art des zu lösenden Problems werden weitere Informationen benötigt.

Abbildung 2-5 zeigt einen beispielhaften Graphen zur Darstellung eines Transportproblems. In diesem Beispiel existiert ein Depot D_1 von welchem aus über drei Routen die Stationen s_1 bis s_{10} angefahren werden. Da es das Ziel ist die Transportkosten zu minimieren, müssen die Kanten gewichtet werden.

Diese Gewichtung ist in der Regel die Distanz zwischen den zwei Knoten, die benötigte Zeit um von einem Knoten zum Anderen zu gelangen oder schlichtweg die Kosten welche beim Zurücklegen dieser Strecke entstehen. Zusätzlich sind weitere Informationen denkbar, wie

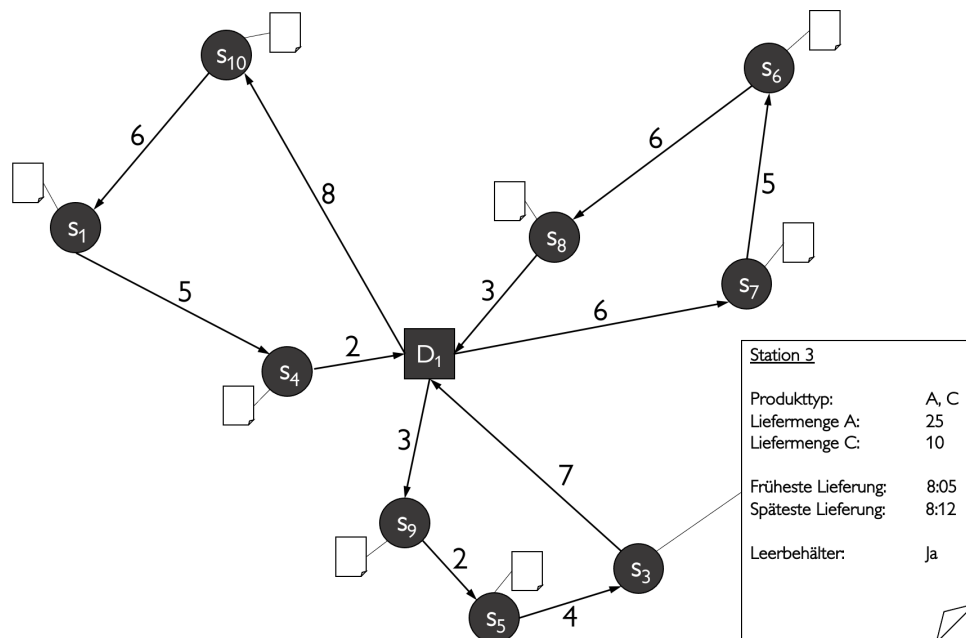


Abbildung 2-5: Graphenbeispiel eines Logistik Optimierungsproblems; in Anlehnung an [Tes-2019, S. 29]

etwa, die früheste oder späteste Ankunftszeit an den jeweiligen Knoten, die nachgefragte Menge eines bestimmten Gutes oder etwa mögliche Restriktionen der zu verwendbaren Transportmittel.

2.5 Relevante Optimierungsprobleme

Die in diesem Kapitel vorgestellten Optimierungsprobleme finden vor allem in der Logistik Anwendung, jedoch basieren viele Optimierungen in anderen Bereichen der Industrie

ebenfalls auf diesen grundlegenden Problemen. Ein Beispiel ist das im Folgenden vorgestellte *Problem des Handlungsreisenden*, welches das Ziel verfolgt, die Transportkosten eines reisenden Vertreters zu minimieren. Dieses Problem tritt ebenfalls in der effizienten Verkabelung von Computer Interfaces auf, in der möglichst optimalen Maschinenzuweisung bei nicht vorhandener Zwischenlagerung, in der Clusterung von Data-Arrays und in vielen weiteren Bereichen. [Len-1974]

2.5.1 Kürzeste-Wege-Problem

Die vermeintlich simpelste Problemstellung ist das Ermitteln des kürzesten beziehungsweise günstigsten Weges von einem Ort zum Anderen (engl. Shortest-Route-Problem). Dieses Problem ist als ein NP-Complete Problem klassifiziert.

In Abbildung 2-6 ist dieses Problem beispielhaft dargestellt. Ziel ist es die günstigste Route vom Depot D_1 zur Station s_9 zu ermitteln. Diese ist in rot eingezeichnet. Möchte man

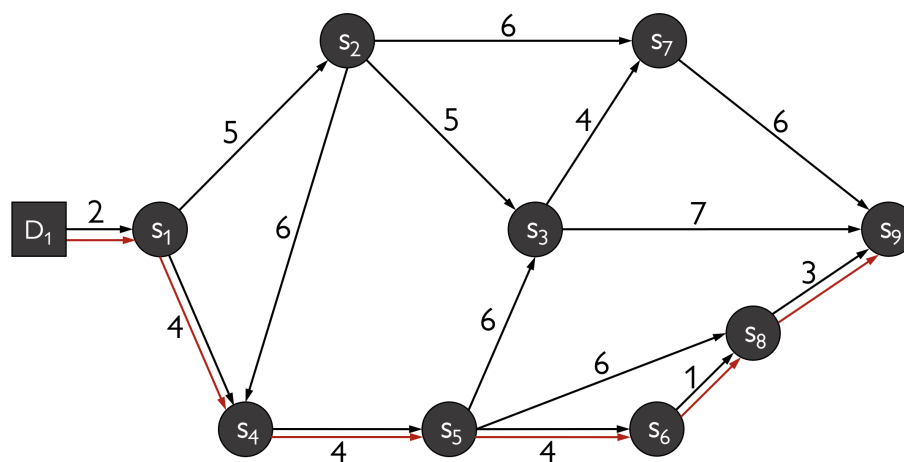


Abbildung 2-6: Kürzeste-Wege-Problem - Graph-Darstellung

dieses Problem ohne Optimierungsalgorithmus lösen, so müsste man um die günstigste Route zu ermitteln, alle möglichen Wege berechnen und miteinander vergleichen. Dies ist in Abbildung 2-7 dargestellt. Es ist leicht ersichtlich, dass die Komplexität mit der Anzahl der möglichen Wege immens steigt. In Abbildung 2-2 sind die wichtigsten Algorithmen um dieses Optimierungsproblem zu lösen sowie deren Komplexität, abhängig von der Anzahl der Knoten V (engl. vertex) und Kanten E (engl. edge) aufgezählt [Ede-2011].

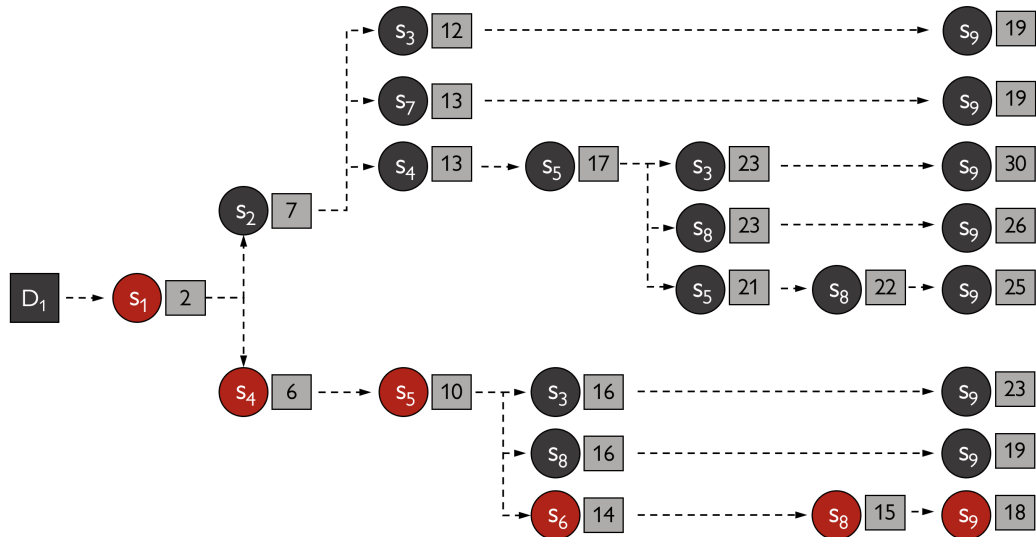


Abbildung 2-7: Kürzeste-Wege-Problem - Ereignisbaum

Tabelle 2-2: Algorithmen zur Lösung von Kürzeste-Wege-Problemen

Algorithmus	Komplexität
Breadth-first, & depth-first search	$O(E + V)$
Dijkstra's algorithm	$O(E + V \log V)$
Bellman-Ford algorithm	$O(VE)$
Floyd-Warshall algorithm	$O(V^3)$

2.5.2 Das Problem des Handlungsreisenden

Ein komplexeres und ebenfalls weit verbreitetes Optimierungsproblem ist das Problem des Handlungsreisenden (engl. Traveling-Salesman-Problem, oder TSP). Es fällt unter die NP-Hard Probleme.

Wie der Name bereits vermuten lässt, zielt es ebenfalls darauf hinaus Transportkosten beziehungsweise Transportzeiten zu minimieren. Üblicherweise wird für die Problembeschreibung ein Vertreter verwendet, welcher alle naheliegenden Städte genau einmal bereisen möchte, und dies möglichst kostengünstig. Offensichtlich hat dieses Problem in der Logistik hohe Relevanz. Ein abgeleitetes Beispiel wäre die Planung einer optimalen Route, ausgehend von dem Depot, um alle Kunden genau einmal zu beliefern und schlussendlich wieder im Depot anzukommen. In Abbildung 2-8 ist dies veranschaulicht.

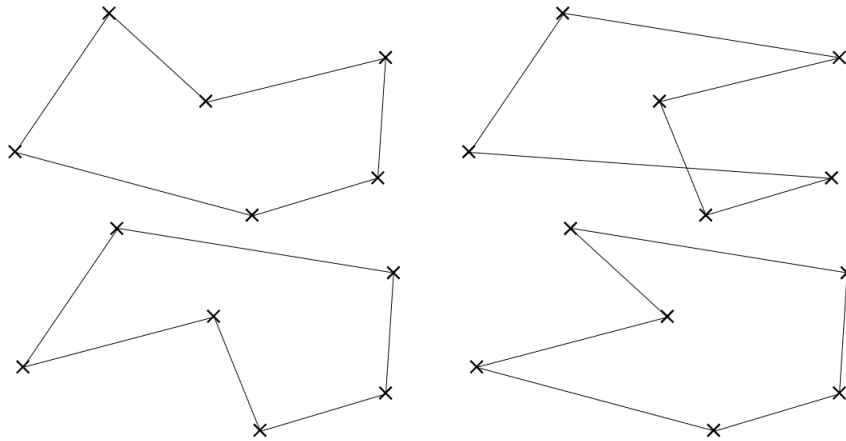


Abbildung 2-8: Beispiel Problem des Handlungsreisenden

Wie zuvor im Kürzeste-Wege-Problem, müssen hierfür die Kanten gewichtet werden und entsprechen gewöhnlicherweise den Distanzen, Zeiten, oder Transportkosten zwischen zwei Knoten. Grundsätzlich gilt die Annahme, dass Kanten zwischen allen Knoten und in beide Richtungen bestehen. Sind diese richtungsunabhängig, sprich die Gewichtung von s_i nach s_{i+1} ist identisch zu der Gewichtung von s_{i+1} nach s_i , dann spricht man von einem symmetrischen TSP und die Anzahl der möglichen Routen würde sich halbieren. Ist dies nicht der Fall, dann spricht man von einem asymmetrischen TSP.

Abhängig von der Ausgangssituation des TSP und den Ansprüchen an Genauigkeit und Geschwindigkeit, existieren diverse heuristische und meta-heuristische Algorithmen. Es wird unterschieden zwischen Tour-Konstruierenden und Tour-Verbessernden Algorithmen.

Bekannte Tour-Konstruierende Algorithmen mit deren Komplexität sind in Tabelle 2-3 aufgeführt. Der Rechenaufwand hängt offensichtlich von der Anzahl der Knoten n ab [Nil-2003]. Für gewöhnlich entspricht die, durch die Tour-Konstruierenden Algorithmen gefundene Lö-

Tabelle 2-3: Algorithmen zum Kreieren von Touren nach

Algorithmus	Komplexität
Nearest Neighbor	$O(n^2)$
Greedy	$O(n^2 \log_2(n))$
Insertion Heuristics	$O(n^2)$
Christofides	$O(n^2 \log_2(n))$

sung nur etwa 10 - 15% der optimalen Lösung. Grund hierfür ist die Tatsache, dass diese

Algorithmen meist stoppen sobald eine Lösung gefunden wurde.

Um eine solche Tour zu optimieren können beispielsweise die in Tabelle 2-4 genannten Algorithmen verwendet werden. [Nil-2003]

Tabelle 2-4: Algorithmen zur Optimierung von Touren nach

Algorithmus	Komplexität
k-opt	$O(n^k)$
Lin-Kernighan	$O(n^{2.2})$
Tabu-Search	$O(n^3)$
Simulated Annealing	$O(n^2)$

2.5.3 Tourenplanungsprobleme

Ein weiteres, ebenfalls in den NP-Hard Problemen angesiedelte Problem ist das Tourenplanungsproblem (engl. Vehicle-Routing-Problem). Hierbei handelt es sich beispielsweise um die Lieferungen eines Produktes, von n Depots $D_1 \dots D_n$ zu m Stationen $S_1 \dots S_m$ mit dem Ziel die Gesamt-Transportkosten zu minimieren.

- Jedes Depot D_i hat ein Angebot von a_i Einheiten des Produktes, mit $i = 1, \dots, n$
- Jede Station S_j hat eine Nachfrage von b_j Einheiten des Produktes, mit $j = 1, \dots, m$
- Die Transportkosten je Einheit des Produktes von dem Depot D_i zu der Station S_j wird mit c_{ij} beschrieben.

Ziel ist es nun Werte für die Entscheidungsvariable x_{ij} zu finden, sodass Angebot und Nachfrage gedeckt sind, und dies zu minimalen Gesamt-Transportkosten. Das zugehörige zu minimierende mathematische Model ist das Folgende:

$$\min z = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

Hierbei sind bestimmte Restriktionen zu erfüllen.

Zum einen darf die maximale Liefermenge eines Depots nicht die maximale angebotene Menge dieses Depots überschreiten:

$$\sum_{j=1}^m x_{ij} \leq a_i, \quad \forall i$$

Gleichzeitig dürfen nicht mehr Einheiten an eine Station geliefert werden, als dort nachgefragt wird:

$$\sum_{i=1}^n x_{ij} \geq b_j, \quad \forall j$$

Und zu guter Letzt darf die Liefermenge nicht negativ sein:

$$x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

Das Tourenplanungsproblem wurde hier in dessen grundlegendster Form vorgestellt. Durch Hinzufügen weiterer Restriktionen, wie etwa zusätzliche Kapazitäten oder exakte Liefertermine entstehen diverse Varianten dieses Problems. Hierfür sei auf die Literatur verwiesen.

2.5.4 Zuweisungsprobleme

Ein weiteres in der Produktionsplanung häufig vorkommendes NP-Hard Problem stellt das Zuweisungsproblem (engl. Assignment Problem) dar [DAR-2012]. Meist wird dieses Problem mittels der Zuordnung von einer bestimmten Anzahl zu verrichtender Arbeiten an eine bestimmte Anzahl an Arbeitern, mit dem Ziel die Kosten zu minimieren vorgestellt. Ein für diese Arbeit passenderes Beispiel wäre folgendes:

Die Zuordnung von n Transportgütern $TG \in \{1, n\}$ auf m Transportmittel/Ladeflächen $TM \in \{1, m\}$, mit den jeweiligen Transportkosten c_{ij} für $TG_i \rightarrow TM_j$.

Die mathematische Entscheidungsvariable x_{ij} kann hierbei zwei Werte annehmen.

$$x_{ij} = \begin{cases} 1, & \text{Transportgut wird dem Transportmittel zugeordnet} \\ 0, & \text{Transportgut wird dem Transportmittel **nicht** zugeordnet} \end{cases}$$

Das Optimierungsproblem lässt sich nun wie folgt darstellen:

$$\min z = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

Typische Nebenbedingungen für ein solches Problem sind die Folgenden:

Jedes Transportgut kann nur einem Transportmittel zugeordnet werden:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n$$

Jedes Transportmittel kann nur ein Transportgut befördern:

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, m$$

$$x_{ij} = 0, 1$$

Falls die Anzahl der Transportmittel der Anzahl der Transportgütern entspricht ($n = m$), dann spricht man von einem ausbalancierten Zuweisungsproblem (engl. Balanced-Assignment-Problem) [Gah-2015]. Ein nicht ausbalanciertes Zuweisungsproblem ($n \neq m$), kann mittels *Dummy-Variablen* zu einem ausbalancierten Zuweisungsproblem umgeformt werden. Die Kosten dieser *Dummy-Variablen* entsprechen 0. Weiterhin, falls ein bestimmtes Transportgut von einem Transportmittel nicht befördert werden kann, so wird dies durch die Zuweisung von unverhältnismäßig großen Transportkosten berücksichtigt.

Da das Ziel die Kostenminimierung ist, wird ein Algorithmus diese Kombination automatisch vermeiden. Aufgrund der oben erwähnten Nebenbedingungen existieren n Möglichkeiten für die erste Zuordnung, $(n - 1)$ Möglichkeiten für die zweite und dementsprechend insgesamt $n!$ unterschiedliche Zuordnungen. Tabelle 2-5 verdeutlicht die Komplexität eines solchen Problems.

Tabelle 2-5: Komplexität des Zuordnungsproblems

Umfang des Zuweisungsproblems	Notwendige Iterationen
3x3	6
6x6	720
9x9	362.880
12x12	479.001.600

Häufig verwendete Algorithmen und Heuristiken zur Reduzierung der Komplexität beziehungsweise zur Lösung solcher Zuweisungsprobleme sind in Tabelle 2-6 aufgeführt. [Özn-2019]

Tabelle 2-6: Algorithmen und Heuristiken zur Lösung von Zuweisungsproblemen

Algorithmus	Komplexität
Brute Force Algorithm	$O(n!)$
OACE Algorithm	$O(n^4)$
Hungarian Algorithm	$O(n^3)$
Greedy Algorithm	$O(n^2 \log n)$

2.6 Lösungsverfahren

Für die algorithmische Lösung eines Optimierungsproblems existieren drei grundlegende Methoden. Theoretisch kann ein jedes Optimierungsproblem exakt gelöst werden, da dies

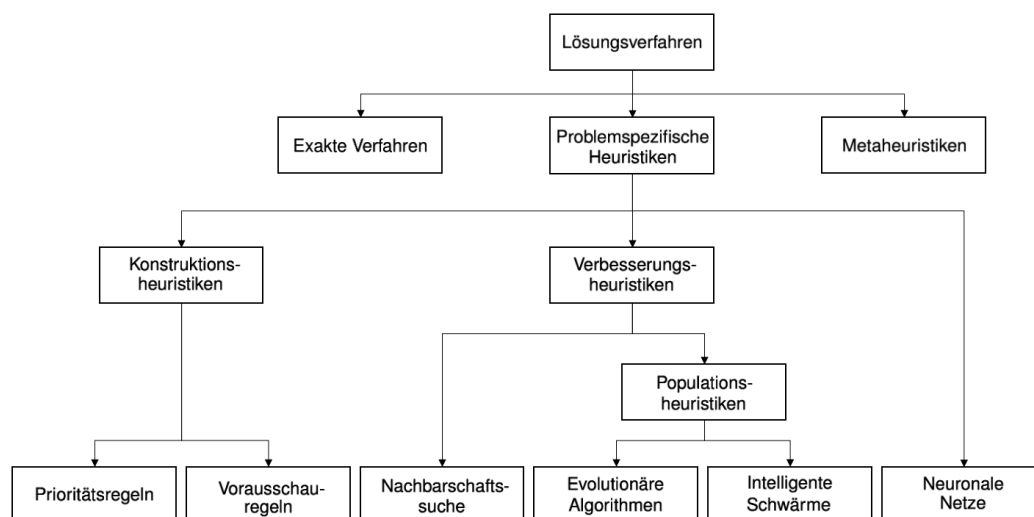


Abbildung 2-9: Einteilung von algorithmischen Lösungsverfahren; in Anlehnung an [Hüf-2006]

jedoch in vielen Fällen mit einem unverhältnismäßigen Aufwand verbunden ist, werden in der Praxis meist Heuristiken beziehungsweise Metaheuristiken angewendet.

Abbildung 2-9 zeigt eine grobe Einteilung von algorithmischen Lösungsverfahren.

2.6.1 Exakte Methoden

An dem zuvor erwähnten Problem des Handlungsreisenden lassen sich die Ausmaße eines exakten Lösungsversuchs mittels eines *Brute Force Algorithmus*, also der Berechnung aller

möglichen Routen, verdeutlichen.

Die Anzahl der möglichen Routen eines symmetrischen TSP entspricht $\frac{1}{2}(n-1)!$. Bei 15 Knoten existieren daher bereits über 43 Milliarden unterschiedliche Routen, bei 18 Knoten übersteigt die Anzahl der Routen bereits 177 Billionen. Es ist leicht ersichtlich, dass eine exakte Berechnung dieses Problems praktisch nicht sinnvoll umsetzbar ist. Dies ist nicht nur bei dem TSP der Fall und daher werden in der Praxis heuristische und metaheuristische Methoden benötigt um die Optimierungsprobleme wirtschaftlich lösen zu können.

2.6.2 Heuristische Methoden

Wenn die Suche der optimalen Lösung nicht sinnvoll, oder schlichtweg nicht möglich ist, werden Heuristische Methoden angewendet. Mittels dieser werden effizient, möglichst gute Lösungen gefunden. Für gewöhnlich ist die mittels heuristischen Methoden ermittelte Lösung nicht optimal und meist kann keine Aussage getroffen werden wie weit die Lösung von dem tatsächlichen Optimum abweicht. Heuristische Methoden zeichnen sich durch folgende Eigenschaften aus [Zim-2008, S. 272 ff.].

Ausschluss potentieller Lösungen Zur Effizienzsteigerung des Algorithmus wird nicht der gesamte Lösungsraum abgesucht.

Nicht-willkürliche Suchprozesse Der verbleibende Lösungsraum wird anhand von definierten Regeln untersucht.

Fehlende Lösungsgarantie Aufgrund der Eingrenzung des zu untersuchenden Lösungsraums kann nicht garantiert werden, dass eine optimale oder zulässige Lösung gefunden wird.

Im Gegensatz zu den sogenannten Näherungsverfahren, welche meist erst im Unendlichen konvergieren, kann bei heuristischen Methoden keine ϵ -Umgebung, in welcher sich die gesuchte Lösung befinden muss, angegeben werden [Zim-2008, S. 274].

2.6.3 Metaheuristische Methoden

Im Gegensatz zu den problemspezifischen Heuristiken, welche wie der Name bereits verrät, lediglich für ein bestimmtes Optimierungsproblem angewendet werden können, existieren

diverse Metaheuristiken. Diese zeichnen sich vor allem dadurch aus, dass sie während der Laufzeit auch Verschlechterungen zulassen und dadurch oftmals anstatt zu einem lokalen, zu einem globalen Optimum führen. [Suh-2009, S. 13 f.]

3 Algorithmus

Der für das im weiteren Verlauf vorgestellte Problem, entwickelte Optimierungsalgorithmus ist eine Kombination aus der Ungarischen Methode und der 2-opt Heuristik, er optimiert daher die Zuweisung der Aufträge zu den Transportmitteln sowie die zu fahrende Route. Diese beiden Optimierungsprobleme werden unabhängig voneinander betrachtet und gelöst. Im ersten Schritt werden vorhandene Aufträge möglichst kostengünstig auf verfügbare Fahrzeuge verteilt und im Nachhinein die zu fahrende Route optimiert.

3.1 Zuweisungsproblem

Die Zuordnung von n Aufträgen $TO \in \{1, n\}$ auf m Transportmittel $TM \in \{1, m\}$, mit den jeweiligen Transportkosten c_{ij} für $TO_i \rightarrow TM_j$.

$$\min z_1 = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

Die mathematische Entscheidungsvariable x_{ij} ist eine binärische Variable und kann daher entweder den Wert 0 oder 1 annehmen. Sobald sie den Wert 1 annimmt, wird der entsprechende Auftrag i dem entsprechenden Transportmittel j zugeordnet und die Kosten c_{ij} entstehen.

Für den Fall dass ein Transportmittel einen Auftrag nicht bearbeiten kann, muss die zugehörige Kostenmatrix C entsprechend angepasst werden. Falls das Transportmittel TM_j den Auftrag TO_i nicht bearbeiten kann, wird den zugehörigen Transportkosten c_{ij} ein Wert M , größer als jeder in der Kostenmatrix C vorkommende, Wert zugeordnet.

Dadurch wird garantiert, dass der Algorithmus diese Auftrag-Transportmittel Kombination ignoriert.

Die Restriktionen sind die des allgemeinen Zuweisungsproblems:

Jeder Auftrag darf nur einem Transportmittel zugeordnet werden und jedem Transportmittel kann nur ein Auftrag zugeordnet werden. Mathematisch lässt sich das folgendermaßen formulieren:

$$\sum_{i=1}^m x_{ij} = 1, \quad \forall n$$
$$\sum_{i=1}^n x_{ij} = 1, \quad \forall m$$
$$x_{ij} = 0, 1$$

Als Kosten c_{ij} werden in diesem Fall die Distanz von dem aktuellen Ort des Fahrzeugs, über den Quellen bis zu den Senken, oder die benötigte Durchlaufzeiten der einzelnen Aufträge gewählt. Die Durchlaufzeit eines Auftrags ist definiert als die Zeitspanne vom Erstellen des Auftrags bis zur vollständigen Bearbeitung desselben. Mit Hilfe dieses Algorithmus wird demnach die Durchlaufzeit des innerbetrieblichen Transportes minimiert.

3.1.1 Ungarische Methode

Die am häufigsten verwendete exakte Methode zur Lösung eines Zuweisungsproblems ist die sogenannte Ungarische Methode, vorgestellt 1955 von Harold W. Kuhn [Sax-2015]. Diese Methode formt eine gegebene Kostenmatrix in eine Matrix mit den entsprechenden Opportunitätskosten um, also Kosten die entstehen, aufgrund der Wahl einer anderen Alternative.

Die zur Lösung eines solchen Problems ebenfalls häufig verwendete Greedy Methode hat gegenüber der Ungarischen Methode zwar einen Vorteil in Sachen Performance, jedoch fällt diese unter die Heuristiken und liefert daher keine Garantie dafür, dass eine optimale Lösung gefunden wurde. [Özn-2019]

Wie zuvor erläutert ist die Komplexität eines solchen Zuweisungsproblems gleich $O(n!)$. Mittels der Ungarischen Methode lässt sich die exakte Lösung mit einem Rechenaufwand von $O(n^3)$ ermitteln.

Die Methode kann grundsätzlich in 5 Schritte unterteilt werden.

Schritt 0 Symmetrische Matrix aufstellen und wenn nötig Dummy Variablen einführen

Schritt 1 Das Minimum jeder Zeile von der jeweiligen Zeile subtrahieren

Schritt 2 Das Minimum jeder Spalte von der jeweiligen Spalte subtrahieren

Schritt 3 Überdecken aller Nullen mit der minimalen Anzahl an horizontalen und vertikalen Linien

Schritt 4 Test auf Optimalität

Schritt 5 Minimum der nicht überdeckten Elemente von eben diesen Elementen subtrahieren und zu den von gleichzeitig horizontaler und vertikaler Linie überdeckten Elementen addieren.

Im Schritt 4 werden die Anzahl der Linien mit dem Rang der Matrix verglichen. Entspricht der Rang der Matrix der Anzahl der Linien so wurde eine optimale Lösung gefunden und die Methode ist beendet. Ist dies nicht der Fall wird Schritt 5 benötigt.

Die Schritte 3 bis 5 werden solange wiederholt bis die Anzahl der mindestens benötigten Linien um alle Nullen zu bedecken, dem Rang der Matrix entspricht, also eine optimale Lösung gefunden wurde. Sobald alle Schritte abgearbeitet wurden und eine optimale Lösung gefunden wurde, kann die finale Zuweisung mittels der erzeugten Nullen vorgenommen werden. Diese Zuweisung muss nicht zwangsläufig eindeutig sein, denn je nach Ausgangsmatrix kann es vorkommen, dass im Laufe der Iterationen mehrere Nullen pro Zeile beziehungsweise Spalte erzeugt werden. [Iye-2008, S. 135]

Bei der Zuweisung, also der Auswahl der Nullen, müssen lediglich die anfangs festgelegten Nebenbedingungen beachtet werden. Das bedeutet, je Zeilen-Spaltenkombination darf nur eine Null ausgewählt werden. Alle unter dieser Voraussetzung möglichen Kombination führen zu dem gleichen Optimum.

In Abbildung 1 bis 5 ist diese Methode anhand eines simplen Beispiels illustriert.

Angenommen es sollen 4 Transportgüter $n_1, \dots, n_4$ mittels 4 Transportmittel $m_1, \dots, m_4$ befördert werden. Als Transportkosten $c_{11}, \dots, c_{44}$ werden die jeweils benötigten Wege gewählt.

87	0	0	73
0	51	29	77
0	56	81	65
20	49	23	0

87	0	0	73
0	51	29	77
0	56	81	65
20	49	23	0

116	0	0	73
0	22	0	48
0	27	52	36
49	49	23	0

Abbildung 3-2: Schritt 2-5: Optimalitätstest und Erzeugung weiterer Nullen

		Transportgüter												
Transportmittel		91	14	4	77	87	10	0	73	(-4)	87	0	0	73
		10	71	39	87	0	61	29	77	(-10)	0	51	29	77
		1	67	82	66	0	66	81	65	(-1)	0	56	81	65
		35	74	38	15	20	59	23	0	(-15)	20	49	23	0
													(-0)	(-10)

Abbildung 3-1: Schritt 1 und 2: Subtraktion der Zeilen- und Spaltenminima

116	0	0	73	116	0	0	73	91	14	4	77
0	22	0	48	0	22	0	48	10	71	39	87
0	27	52	36	0	27	52	36	1	67	82	66
49	49	23	0	49	49	23	0	35	74	38	15

Abbildung 3-3: Finale Zuordnung

Die optimale Zuordnung der Aufträge, welche zu den minimalen Kosten von 69 führt, ist in Tabelle 3-1 dargestellt.

Tabelle 3-1: Optimale Zuordnung des Zuweisungsproblems

Transportmittel	Zugeordneter Auftrag
TM m_1	TG n_2
TM m_2	TG n_3
TM m_3	TG n_1
TM m_4	TG n_4

Um diesen Algorithmus für das vorliegende Problem verwenden zu können, müssen die Kosten c_{ij} in eine Matrix zusammengeführt werden. Wie in dem Beispiel zuvor, werden die Transportaufträge/Transportgüter horizontal und die verfügbaren Transportmittel vertikal aufgetragen. Je nach Konstellation müssen Dummy Variablen hinzugefügt werden um eine quadratische Matrix zu erhalten [Iye-2008, S. 141]. Falls eine bestimmte Transportauftrag-Transportmittel-Kombination nicht möglich ist, werden hierfür wie erwähnt die fiktiven Ko-

sten M eingetragen. Die benötigte Kostenmatrix C zu einem bestimmten Zeitpunkt t , bei mehr offenen Transportaufträgen als verfügbaren Transportmittel, könnte wie folgt aussehen:

$$C_t = \begin{bmatrix} 21461 & 4563 & 654 & M & 454 \\ 435 & M & 4527 & 987 & 9223 \\ M & 5637 & M & 444 & 328 \\ 222 & 546 & 7387 & 6643 & M \\ 436 & 1111 & 827 & 8008 & M \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3.2 Tourenplanungsproblem

Sobald die Zuweisung der Aufträge erfolgt ist und ein Transportmittel mit einer Kapazität größer 1 mehrere Aufträge zugeordnet bekommen hat, soll die zu fahrende Route optimiert werden. Wie bereits in Kapitel 2.5.3 beschrieben, ist dies ein sehr komplexes mathematisches Problem, welches sich für gewöhnlich wirtschaftlich nicht exakt lösen lässt. Aus diesem Grunde soll im Folgenden auf die mathematische Formulierung des Problems verzichtet werden und stattdessen direkt auf den Lösungsvorschlag mittels der k -opt Heuristik eingegangen werden. Für die mathematische Formulierung eines solchen Problems sei auf die Arbeit von K. S. Ruland verwiesen [Rul-1994].

3.2.1 k -opt Heuristik

Der k -Opt Algorithmus ist eine Touren-Optimierende Heuristik, sie ist also nicht in der Lage eine Tour zu kreieren, sondern wird genutzt um eine bereits vorhandene Tour zu optimieren. Der Algorithmus versucht die jeweilige Tour iterativ zu verbessern, in dem er k Kanten trennt und diese neu verbindet, solange bis keine Verbesserung mehr möglich ist. Die resultierende Tour wird dann mit k -opt Tour bezeichnet. Im schlimmsten Fall ist hierfür eine

exponentielle Anzahl an Iterationen nötig [Cha-1999]. Dies ist jedoch keineswegs der Standardfall, denn für praktische Anwendungen wird meist eine zufriedenstellende Lösung in polynomialer Zeit erreicht [Eng-1999].

Die am häufigsten verwendeten k -opt Varianten sind die 2-opt und die 3-opt Heuristiken, jedoch sind auch andere Werte oder gar ein mit der Laufzeit veränderbarer k -Wert denkbar. Da der Fokus dieser Arbeit auf dem Lösen des Zuweisungsproblems liegt und aus Gründen der Performance, wurde zur Lösung des vorliegenden Tourenplanungsproblems die 2-opt Heuristik gewählt. Eine durch diese Heuristik erhaltene Tour, also eine 2-opt Tour, mit n Knoten ist nicht schlechter als $4\sqrt{n}$ mal das Optimum [Cha-1999].

In jeder Iteration des 2-opt Algorithmus müssen die neu erstellten Touren bewertet werden und mit der aktuellen Route verglichen werden. Die Anzahl der möglichen Touren für jede Iteration entspricht $n \cdot \frac{n-3}{2}$. Damit eine dieser neuen Touren ausgewählt wird, muss diese strikt besser sein als die bisherige.

Zur Bewertung der neuen Touren können entweder alle möglichen neuen Touren berechnet und mit der aktuellen Tour verglichen werden oder um die Performance des Algorithmus zu steigern, direkt die erste bessere Route gewählt werden.

Der Algorithmus endet entweder sobald keine Verbesserungen mehr möglich sind oder nach einer festgelegten Laufzeit beziehungsweise nach einer festgelegten Anzahl an Iterationen. Wie bereits erwähnt muss vor Verwendung der Heuristik bereits eine Tour vorhanden sein. Diese kann entweder zufällig bestimmt oder mittels eines Tour-Kreierenden Algorithmus erstellt werden.

Für den in Abbildung 3-4 dargestellten Graphen mit 5 Knoten ($s_1 - s_5$) muss daher primär eine initiale Route gewählt werden. Für dieses Beispiel wurde die in Abbildung 3-5 (a) gezeigte Route zufällig gewählt, welche Kosten von 33 verursacht .

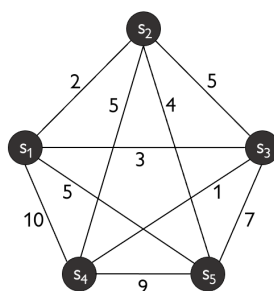


Abbildung 3-4: Beispielgraph für 2-opt Illustration

Ein beispielhafter Iterationsschritt des 2-opt Algorithmus ist in Abbildung 3-5 dargestellt. Je nach Gestaltung des Algorithmus wählt dieser entweder die erste Route aus welche eine Verbesserung zur Folge hat, in diesem Fall wäre dies die Route (b) mit Transportkosten von 26, oder er vergleicht alle möglichen Routen und wählt daher die Route (e) als Referenz für den nächsten Iterationsschritt, da diese Kosten von 22 verursacht.

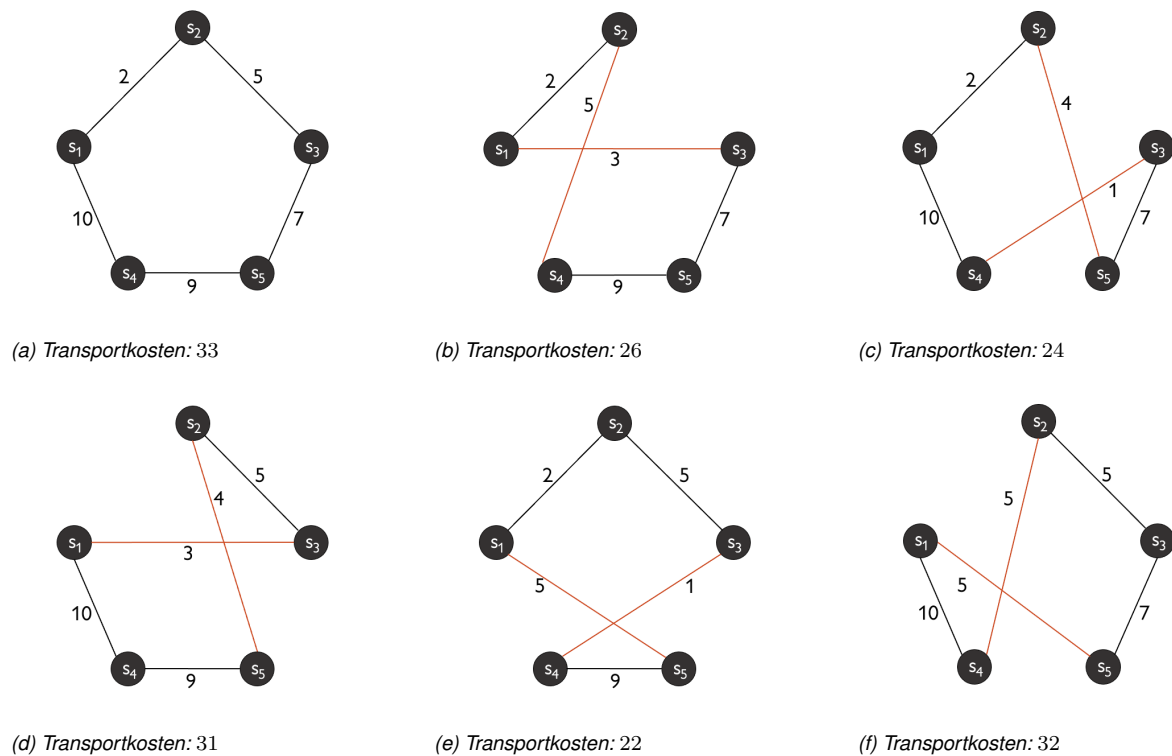


Abbildung 3-5: Beispielhafter 2-opt Iterationsschritt für die initiale Route in (a)

4 Implementierung

Um den im vorherigen Kapitel vorgestellten Algorithmus evaluieren zu können und ihn in annähernd realistischen Bedingungen zu testen, soll dieser in einem Simulationsprogramm implementiert werden. Die Vorteile einer Simulation liegen auf der Hand.

Es lassen sich eine Vielzahl an Was-wäre-wenn Szenarien untersuchen, ohne hierbei das bestehende System verändern zu müssen, oder gar vor der eigentlichen Einführung eines neuen Systems [Sei-2008]. Neben dem Simulationsprogramm und selbstverständlich der Zeit für die Implementierung und Simulation werden keinerlei Ressourcen benötigt um diese Szenarien zu testen und zu evaluieren. Fehler, welche erst nach einer sehr langen Laufzeit auftreten würden, können am Computer innerhalb weniger Sekunden oder Minuten simuliert werden, was zu einer enormen Risikominimierung führt. Je nach Simulationsprogramm können die Simulationen zusätzlich in 2D oder gar 3D visualisiert werden, was eine noch realistischere Evaluierung ermöglicht.

Der Implementierungsaufwand sowie die Lizenzkosten eines solchen Simulationsprogramms sind jedoch nicht zu unterschätzen. Eine vorherige Prüfung über die Sinnhaftigkeit des Einsatzes sollte durchgeführt werden. Weiterhin ist eine Simulation eine Abbildung der Realität in ein für die Evaluierung vereinfachtes Modell. Daher hängen die Resultate der Simulation immens von der Genauigkeit der Abbildung ab. Um also falsche Schlüsse aufgrund der Abweichungen zum realen System zu vermeiden, sollte vor Beginn der Testung, primär das erstellte Modell auf Exaktheit geprüft werden [Sei-2008].

4.1 Plant Simulation

Das für die Implementierung gewählte Simulationsprogramm ist das von Siemens entwickelte Plant Simulation. Dieses ist speziell ausgelegt für produzierende Unternehmen sowie reine Logistikunternehmen und bietet daher viele bereits ab Werk implementierte Funktionen einen solchen Algorithmus zu testen und auszuwerten. Für die Simulation benötigte Objekte wie beispielsweise Ladeeinheiten, Transportmittel, Arbeitsstationen, Lager sind bereits in Plant Simulation implementiert und können je nach Bedarf angepasst werden. Um das Simulationsmodell so nah wie möglich an die Realität anzugleichen sind be-

nutzerdefinierte Funktionen, sogenannte Methoden, notwendig. Diese werden in der eigens entwickelten Programmierumgebung SimTalk erstellt. Zusätzlich existieren Schnittstellen zu anderen Programmierumgebungen wie etwa C.

4.2 Implementierung in das Simulationsprogramm

Das schematische Layout des zu optimierenden Systems ist in Abbildung 4-1 dargestellt. Sobald ein neuer Auftrag erstellt wurde, wird die entsprechende Transporteinheit in den auf der linken Seite angeordneten Quellen erzeugt. Jeder Auftrag besteht aus genau einer Transporteinheit. Diese muss nun abgeholt und an die zugehörige Senke, welche rechts angeordnet sind, geliefert werden.

In diesem System werden drei verschiedene Transportmittel verwendet. Gabelstapler (ST), Fahrerlose Transportsystem (FTS oder AGV) und Routenzüge (RZ). Die Ladekapazitäten der Gabelstapler sowie der FTS entsprechen der einer Ladeeinheit und die der Routenzüge, die der fünf Ladeeinheiten. Zu Beginn der Simulation befinden sich die Fahrzeuge in den jeweiligen Fahrzeugpools dargestellt in Abbildung 4-1.

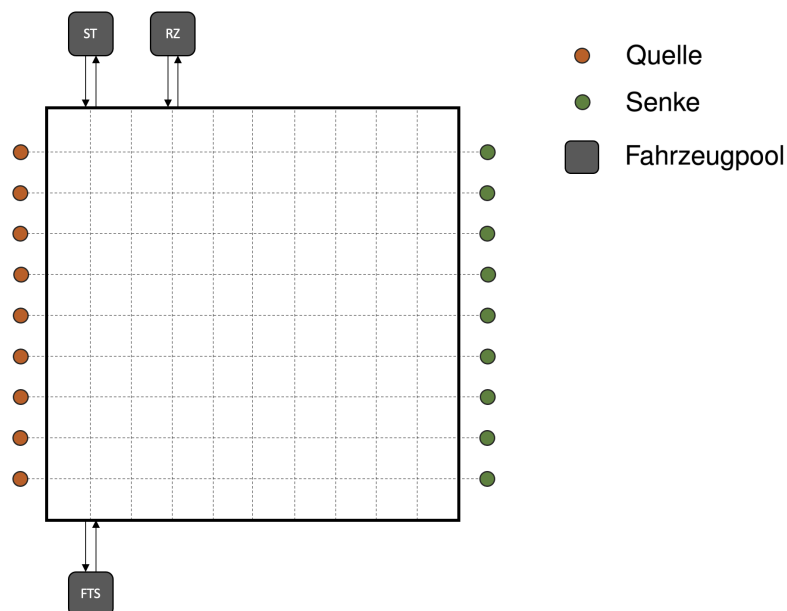


Abbildung 4-1: Schematisches Layout

Abbildung 4-2 zeigt den vereinfachten Ablauf des Zuweisungsalgorithmus bei der fahrzeugorientierten Auftragszuweisung. Dieser wird hierbei von den jeweiligen Fahrzeugen aufgerufen, sobald diese einen Auftrag bearbeitet haben und wieder freie Kapazitäten besitzen.

Für den Fall, dass zu diesem Zeitpunkt keine Aufträge verfügbar sind, wird der Algorithmus in einer Sekunde erneut aufgerufen. Falls das Fahrzeug mehrere Kapazitäten besitzt und zugleich mehrere Aufträge zugeordnet hat, so wird automatisch die 2-opt Methode aufgerufen um die Tour zu optimieren.

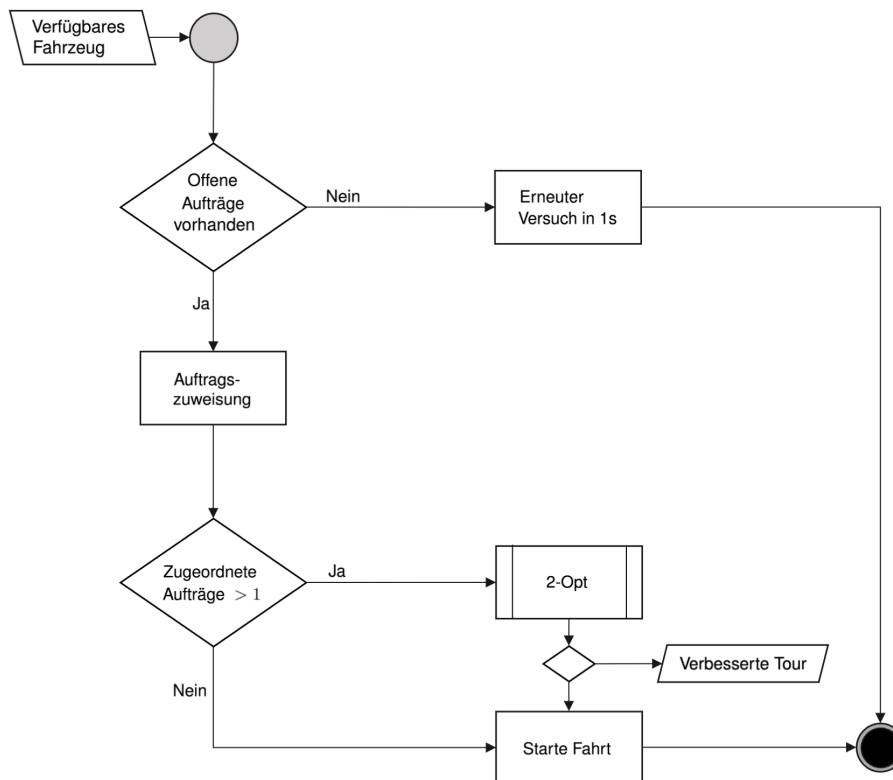


Abbildung 4-2: Flussdiagramm - Fahrzeugorientierte Auftragszuweisung

Der grobe Funktionsablauf bei einer auftragsorientierten Auftragszuweisung ohne Optimierung der Zuweisung ist in Abbildung 4-3 illustriert. In diesem Fall wird der Algorithmus ausgelöst durch die Erstellung eines neuen Auftrags. Für diesen wird nun ein passendes Transportmittel gesucht. Um die Funktionsweisen der einzelnen Algorithmen beziehungsweise die aus deren Verwendung resultierenden Systemleistungen im Folgenden besser evaluieren zu können, soll die Zuweisung in diesem Fall ohne Optimierung geschehen. Aus diesem Grund wird aus der Liste der momentan in Frage kommenden Fahrzeuge ein zufälliges gewählt. Dadurch wird sichergestellt, dass die Reihenfolge in der die Fahrzeuge im System hinterlegt sind, keinen Einfluss auf die Ergebnisse hat.

Falls es sich bei dem Fahrzeug um einen Routenzug handelt und dieser nach der erfolg-

ten Auftragszuweisung mehrere Aufträge zugeordnet hat, wird die Route wie auch bei der fahrzeugorientierten Auftragszuweisung mittels der 2-opt Heuristik optimiert.

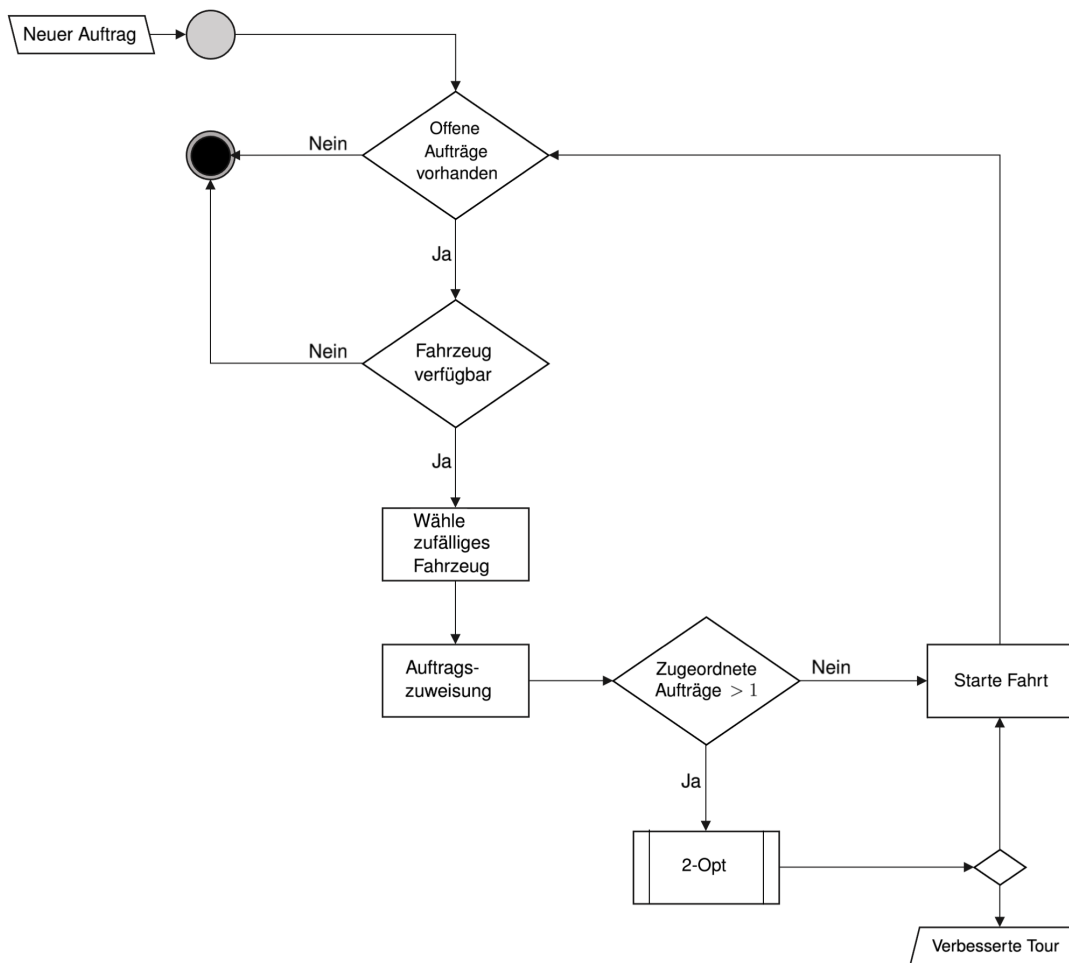


Abbildung 4-3: Flussdiagramm - Auftragsorientierte Auftragszuweisung ohne Optimierung

Bei Verwendung der Auftragszuweisung mittels des Ungarischen Algorithmus wird der in Abbildung 4-4 vereinfacht dargestellte Ablauf durchgeführt. Auslöser ist auch hier zunächst die Erstellung eines neuen Auftrags. Allerdings werden hier nicht nur der am längsten wartende Auftrag betrachtet, sondern alle derzeit offenen. Zusammen mit den derzeit verfügbaren Transportmitteln wird zunächst die für den Algorithmus benötigte Kostenmatrix erstellt. Dies geschieht wie bereits in Kapitel 3.1.1 beschrieben.

Mit Hilfe des Ungarischen Algorithmus werden dann den Aufträgen die verfügbaren Fahrzeugen zugeordnet. Sobald alle Fahrzeuge einen Auftrag zugeordnet bekommen haben oder alle Aufträge verteilt wurden, wird innerhalb einer Schleife über alle Zuordnungen nach dem selben Prinzip wie bei den vorherigen Zuweisungsalgorithmen, die Route für

jedes Fahrzeug bestimmt und die Fahrt gestartet.

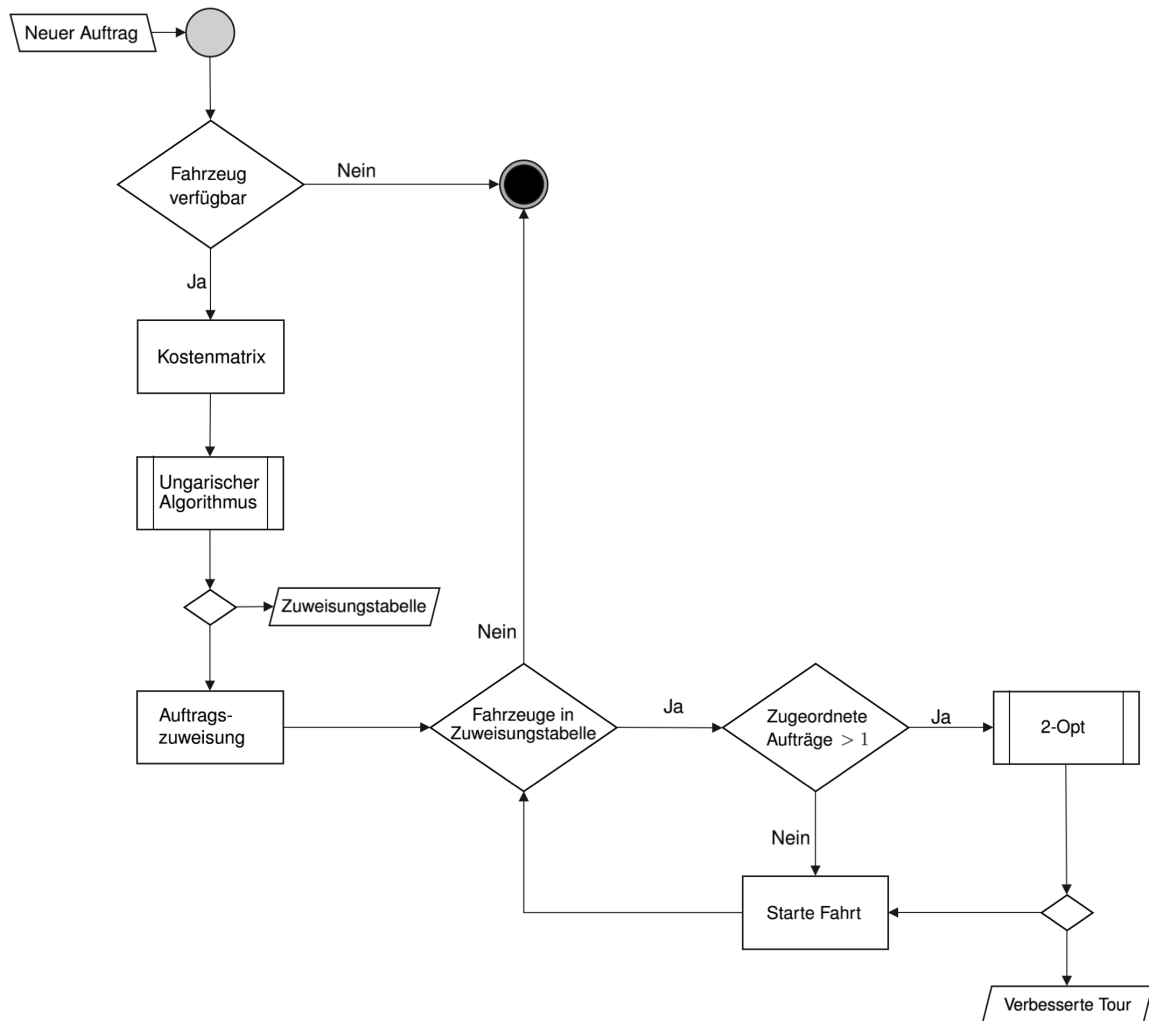


Abbildung 4-4: Flussdiagramm - Fahrzeugorientierte Auftragszuweisung mit Ungarischem Algorithmus

In Abbildung 4-5 sind die Schritte der zuvor vorgestellten Ungarischen Methode grob dargestellt. Diese wird innerhalb der Auftragszuweisung aufgerufen und ihr wird als Input die zuvor erstellte Kostenmatrix übergeben. Diese Matrix wurde bereits so bearbeitet, dass die Methode angewendet werden kann, denn wie bereits erwähnt muss die Kostenmatrix eine quadratische Form besitzen. Falls also die Anzahl der verfügbaren Transportmittel nicht mit der Anzahl der offenen Aufträge übereinstimmt, so wird die kleinere Dimension mit Nullen aufgefüllt. Die Funktion *Markiere Zeilen und Spalten* ist das Äquivalent zu dem Schritt des Linienziehens. Hierfür wird die Kostenmatrix in eine boolesche Matrix ($0 = True$) umgeformt und die entsprechenden Zeilen und Spalten markiert. Falls die Anzahl der markierten Zeilen und Spalten nicht dem Rang der Matrix entspricht wird die Kostenmatrix wie in Ka-

pitel 3.1.1 Schritt 5 bearbeitet. Nach der erfolgreichen Zuweisung wird eine entsprechende Tabelle erstellt und an den aufrufenden Algorithmus zurückgegeben.

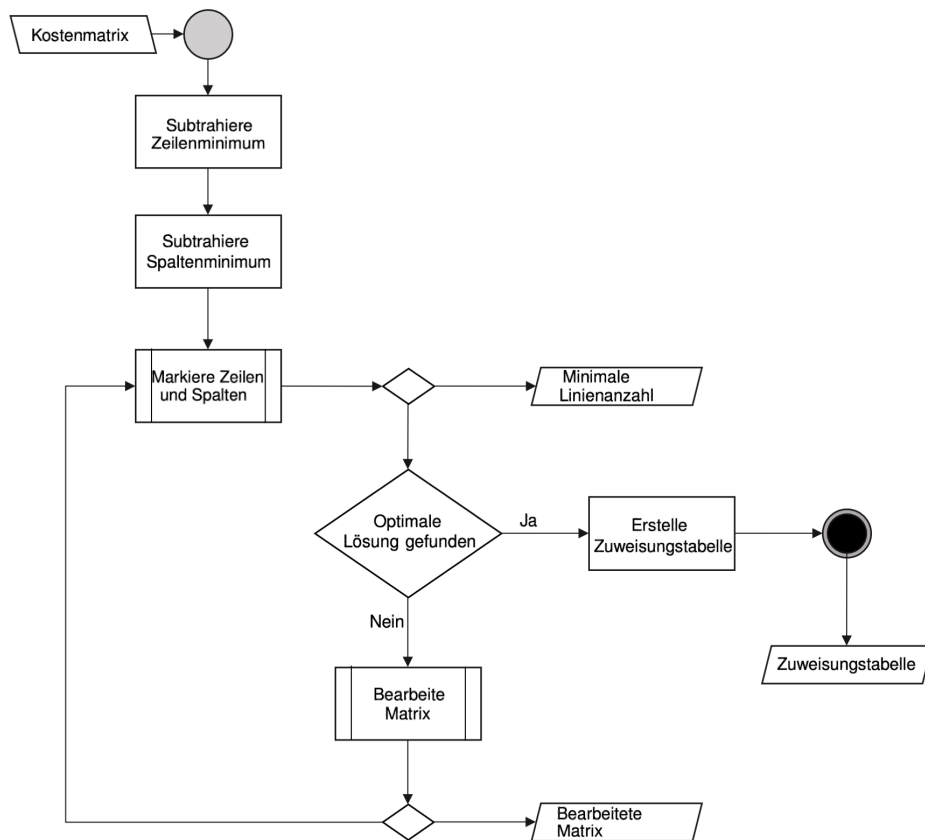


Abbildung 4-5: Flussdiagramm - Ungarische Methode

Alle Fahrzeuge deren Kapazität größer Eins ist und die mehrere Aufträge zugeordnet haben, rufen die in Abbildung 4-6 dargestellte 2-opt Methode auf.

Im Zuge der Auftragszuweisung werden die entsprechende Quelle und Senke eines neu zugeordneten Auftrags zunächst der bestehenden Route des Fahrzeugs angehängt. Diese Route ist die Referenzroute welche mittels der 2-opt Methode verbessert werden soll und für die zunächst die Kosten berechnet werden. Abhängig von den Einstellungen im User Interface werden entweder die zurückzulegende Wegstrecke oder die durchschnittliche Durchlaufzeit der zu bearbeitenden Aufträge als Transportkosten verwendet. Die 2-opt Methode vertauscht nun innerhalb einer Schleife die einzelnen Knotenpunkte, also die Quellen und Senken der Aufträge und vergleicht die dadurch entstehenden Kosten mit den Referenzkosten. Sind diese geringer, sprich es wurde eine günstigere Route gefunden, muss im nächsten Schritt überprüft werden, ob diese zulässig ist. Hierfür müssen die Quellen ei-

nes jeden Auftrags vor der zugehörigen Senke angefahren werden, da die Transporteinheit schließlich erst abgeholt werden muss, bevor diese geliefert werden kann.

Falls eine Route niedrigere Kosten aufweist und zugleich zulässig ist, so ist dies die neue Referenzroute und es wird versucht diese Route im folgenden Iterationsschritt weiter zu verbessern. Dies wird solange wiederholt bis keine Verbesserung mehr möglich ist.

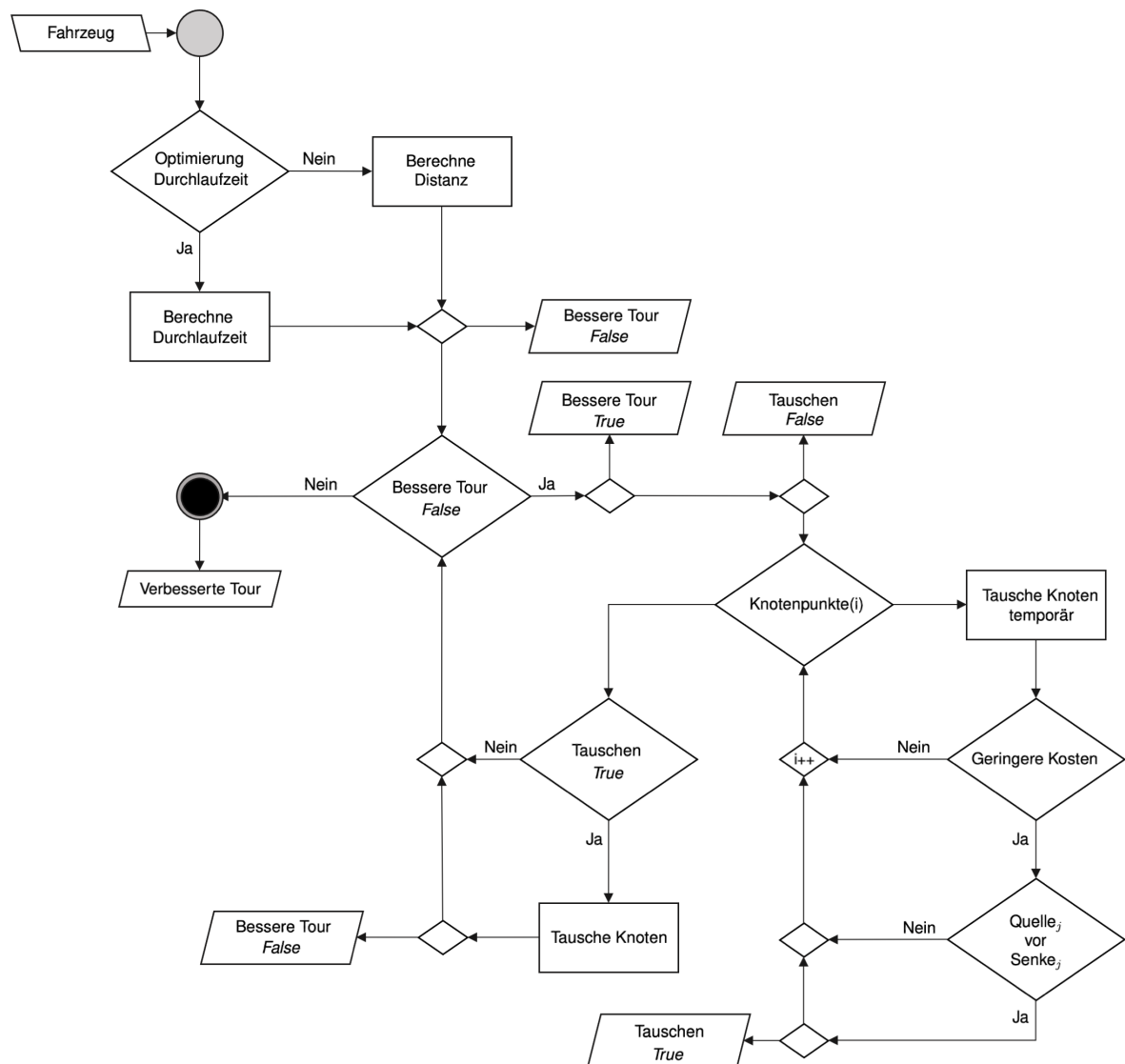


Abbildung 4-6: Flussdiagramm - 2-opt Methode

5 Evaluierung

Wie zuvor erwähnt, dadurch dass der Algorithmus in einem Simulationsprogramm implementiert wurde, lässt sich dieser sehr effizient unter realitätsnahen Bedingungen evaluieren. Hierfür werden verschiedene Szenarien und Algorithmus-Kombinationen untersucht. Zum einen soll der Einfluss der Transportmittel-Zusammensetzung, also das Verhältnis von Direkttransport zu Routenzügen, untersucht werden. Zum anderen sollen die Unterschiede zwischen einer auftragsorientierten und einer fahrzeugorientierten Auftragszuweisung aufgezeigt werden.

Weiterhin werden die Simulationen für unterschiedliche Systemauslastungen durchgeführt. Hierfür wird zwischen einem nicht ausgelastetem, einem deutlich überlastetem sowie einem gut ausgelastetem System unterschieden. Die Referenz für das gut ausgelastete System ist die erreichbare Leistung bei der Verwendung des Ungarischen Algorithmus sowie der 2-opt Heuristik.

Die erste Unterscheidung ist damit welche Art von Transport überwiegend verwendet wird, also ob vermehrt Direkttransport oder Routenzüge zum Einsatz kommen. Als nächstes wird zwischen den drei zuvor erwähnten Systemauslastungen unterschieden. Für all diese Szenarien werden jeweils neun verschiedene Algorithmus-Kombinationen untersucht, was zu den in Tabelle 5-1 und 5-2 beschriebenen Versuchen führt.

Um vergleichbare Ergebnisse zu erhalten, müssen alle Simulationen ausreichend lange durchgeführt werden. Hierfür wurde für alle Versuche eine Simulationszeit von 5 Tagen gewählt, da bei dieser, die natürlich vorkommenden anfänglichen Schwankungen aller Algorithmus-Kombinationen, keinen merkbaren Einfluss mehr spielen und das System sich eingependelt hat.

Weiterhin werden jeweils 28 Transportmittel verwendet. Für den ersten Fall, also bei vermehrtem Direkttransport entspricht dies 12 Gabelstaplern, 12 Fahrerlose Transportsysteme sowie 4 Routenzügen.

Für die Versuche bei vermehrter Verwendung von Routenzügen werden 4 Gabelstapler, 4 Fahrerlose Transportsysteme und 20 Routenzüge verwendet.

Tabelle 5-1: Versuche zur Evaluierung des Optimierungsalgorithmus - Vermehrt Direkttransport

Nr.	Auslastung	Auslöser	2-Opt	Optimierungs- variable 2-Opt	Ungarischer Algorithmus
1	Nicht ausgelastet	Fahrzeugorientiert	Nein	–	Nein
2	Nicht ausgelastet	Fahrzeugorientiert	Ja	Distanz	Nein
3	Nicht ausgelastet	Fahrzeugorientiert	Ja	Durchlaufzeit	Nein
4	Nicht ausgelastet	Auftragsorientiert	Nein	–	Nein
5	Nicht ausgelastet	Auftragsorientiert	Ja	Distanz	Nein
6	Nicht ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Nein
7	Nicht ausgelastet	Auftragsorientiert	Nein	–	Ja
8	Nicht ausgelastet	Auftragsorientiert	Ja	Distanz	Ja
9	Nicht ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Ja
10	Überlastet	Fahrzeugorientiert	Nein	–	Nein
11	Überlastet	Fahrzeugorientiert	Ja	Distanz	Nein
12	Überlastet	Fahrzeugorientiert	Ja	Durchlaufzeit	Nein
13	Überlastet	Auftragsorientiert	Nein	–	Nein
14	Überlastet	Auftragsorientiert	Ja	Distanz	Nein
15	Überlastet	Auftragsorientiert	Ja	Durchlaufzeit	Nein
16	Überlastet	Auftragsorientiert	Nein	–	Ja
17	Überlastet	Auftragsorientiert	Ja	Distanz	Ja
18	Überlastet	Auftragsorientiert	Ja	Durchlaufzeit	Ja
19	Gut ausgelastet	Fahrzeugorientiert	Nein	–	Nein
20	Gut ausgelastet	Fahrzeugorientiert	Ja	Distanz	Nein
21	Gut ausgelastet	Fahrzeugorientiert	Ja	Durchlaufzeit	Nein
22	Gut ausgelastet	Auftragsorientiert	Nein	–	Nein
23	Gut ausgelastet	Auftragsorientiert	Ja	Distanz	Nein
24	Gut ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Nein
25	Gut ausgelastet	Auftragsorientiert	Nein	–	Ja
26	Gut ausgelastet	Auftragsorientiert	Ja	Distanz	Ja
27	Gut ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Ja

Tabelle 5-2: Versuche zur Evaluierung des Optimierungsalgorithmus - Vermehrt Routenzüge

Nr.	Auslastung	Auslöser	2-Opt	Optimierungs- variable 2-Opt	Ungarischer Algorithmus
28	Nicht ausgelastet	Fahrzeugorientiert	Nein	–	Nein
29	Nicht ausgelastet	Fahrzeugorientiert	Ja	Distanz	Nein
30	Nicht ausgelastet	Fahrzeugorientiert	Ja	Durchlaufzeit	Nein
31	Nicht ausgelastet	Auftragsorientiert	Nein	–	Nein
32	Nicht ausgelastet	Auftragsorientiert	Ja	Distanz	Nein
33	Nicht ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Nein
34	Nicht ausgelastet	Auftragsorientiert	Nein	–	Ja
35	Nicht ausgelastet	Auftragsorientiert	Ja	Distanz	Ja
36	Nicht ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Ja
37	Überlastet	Fahrzeugorientiert	Nein	–	Nein
38	Überlastet	Fahrzeugorientiert	Ja	Distanz	Nein
39	Überlastet	Fahrzeugorientiert	Ja	Durchlaufzeit	Nein
40	Überlastet	Auftragsorientiert	Nein	–	Nein
41	Überlastet	Auftragsorientiert	Ja	Distanz	Nein
42	Überlastet	Auftragsorientiert	Ja	Durchlaufzeit	Nein
43	Überlastet	Auftragsorientiert	Nein	–	Ja
44	Überlastet	Auftragsorientiert	Ja	Distanz	Ja
45	Überlastet	Auftragsorientiert	Ja	Durchlaufzeit	Ja
46	Gut ausgelastet	Fahrzeugorientiert	Nein	–	Nein
47	Gut ausgelastet	Fahrzeugorientiert	Ja	Distanz	Nein
48	Gut ausgelastet	Fahrzeugorientiert	Ja	Durchlaufzeit	Nein
49	Gut ausgelastet	Auftragsorientiert	Nein	–	Nein
50	Gut ausgelastet	Auftragsorientiert	Ja	Distanz	Nein
51	Gut ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Nein
52	Gut ausgelastet	Auftragsorientiert	Nein	–	Ja
53	Gut ausgelastet	Auftragsorientiert	Ja	Distanz	Ja
54	Gut ausgelastet	Auftragsorientiert	Ja	Durchlaufzeit	Ja

5.1 Versuchsauswertung

Die in Tabelle 5-1 und 5-2 aufgelisteten Versuche sollen im Folgenden überwiegend graphisch ausgewertet werden. Der Fokus liegt hierbei vor allem auf der Auftragserfüllung, der Fahrzeugauslastung sowie der Durchlaufzeit. Die zugehörigen Tabellen mit den Versuchsergebnissen sind im Anhang dieser Arbeit zu finden.

5.1.1 Vermehrt Direkttransport

Die Abbildungen 5-1 bis 5-3 zeigen die Versuchsergebnisse der ersten 9 Versuche. Das System ist für keine der drei Algorithmus-Varianten ausgelastet und alle Aufträge können problemlos abgearbeitet werden. Jedoch unterscheiden sich die Varianten teils immens in der durchschnittlichen Fahrzeugauslastung sowie der durchschnittlichen Durchlaufzeit.

Vergleicht man die Fahrzeugauslastung bei einer fahrzeugorientierten Zuweisung (Versuche 1-3) mit der einer auftragsorientierten Zuweisung bei Verwendung des Ungarischen Algorithmus (Versuche 7-9) fällt auf, dass um die gleiche Leistung zu erzielen die Transportmittel bei der fahrzeugorientierten Zuweisung mehr als doppelt so ausgelastet sind wie bei der Verwendung des Ungarischen Algorithmus. Weiterhin fällt auf, dass die Nutzung der

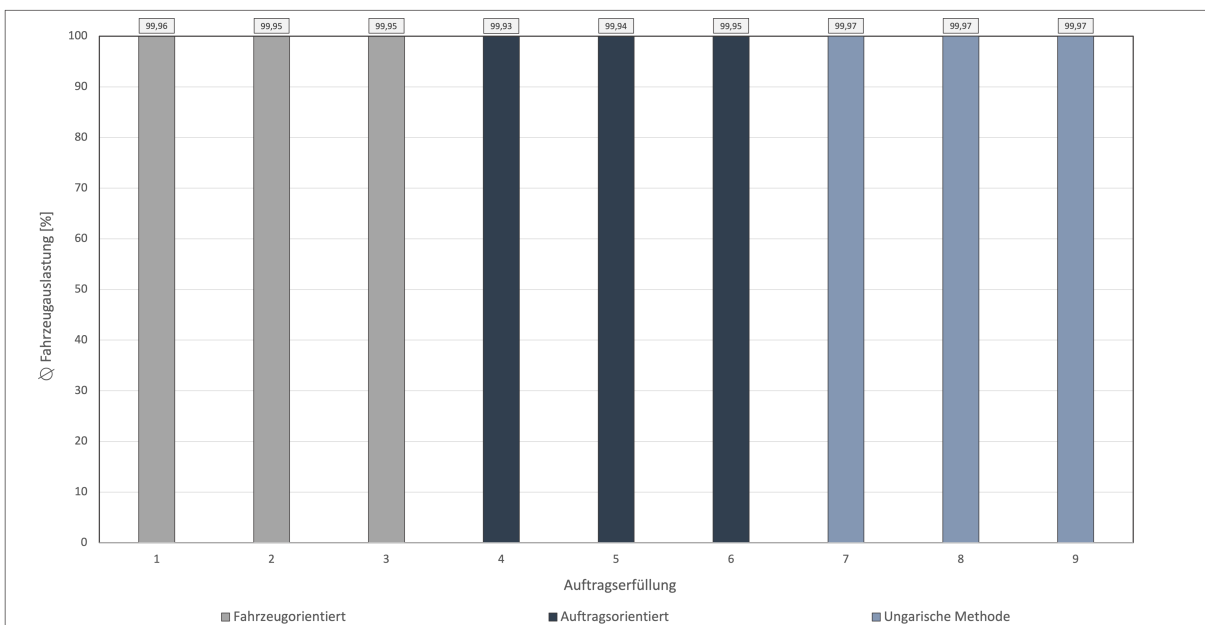


Abbildung 5-1: Durchschnittlicher Grad der Auftragserfüllung - Versuche 1 bis 9

2-opt Heuristik bei der Verwendung des Ungarischen Algorithmus keinerlei Auswirkungen auf die Fahrzeugauslastung und die Durchlaufzeit hat. Das liegt daran, dass durch die

Verwendung der Ungarischen Methode die Aufträge effizienter zugewiesen werden und die erforderliche Leistung auch ohne die Nutzung der Routenzüge erreicht wird. Die Gabelstapler sowie die FTS fahren auch ohne Nutzung der 2-opt Heuristik die optimale Route, da diese in Tabellenform im System hinterlegt sind. Aufgrund des vermehrten

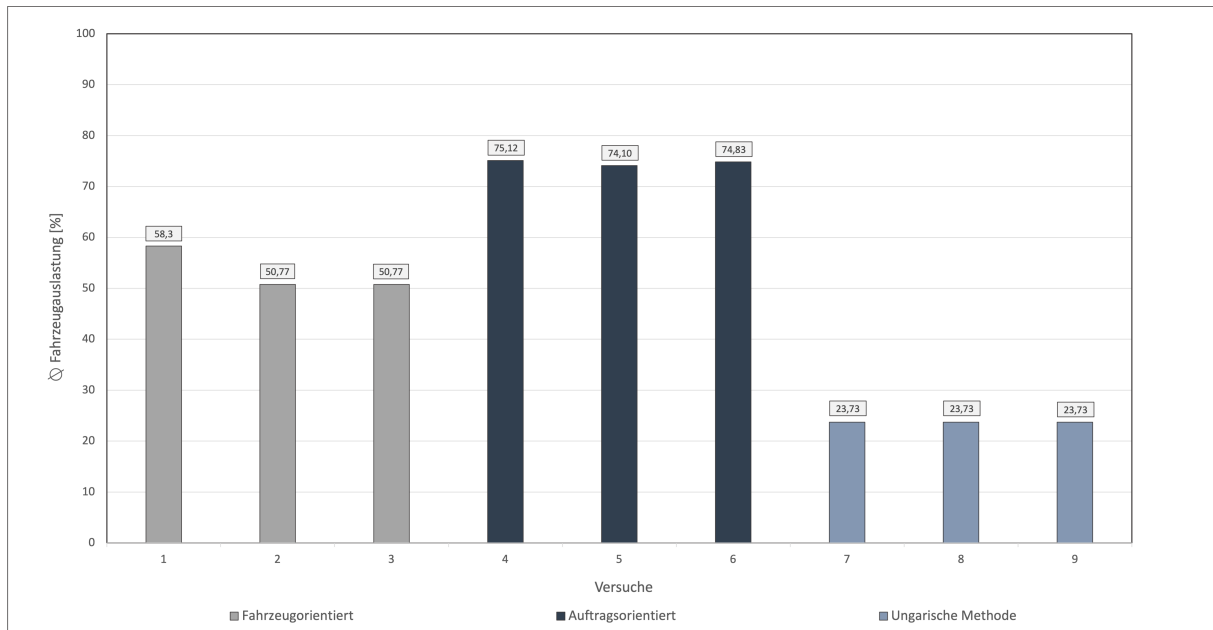


Abbildung 5-2: Durchschnittliche Fahrzeugauslastung - Versuche 1 bis 9

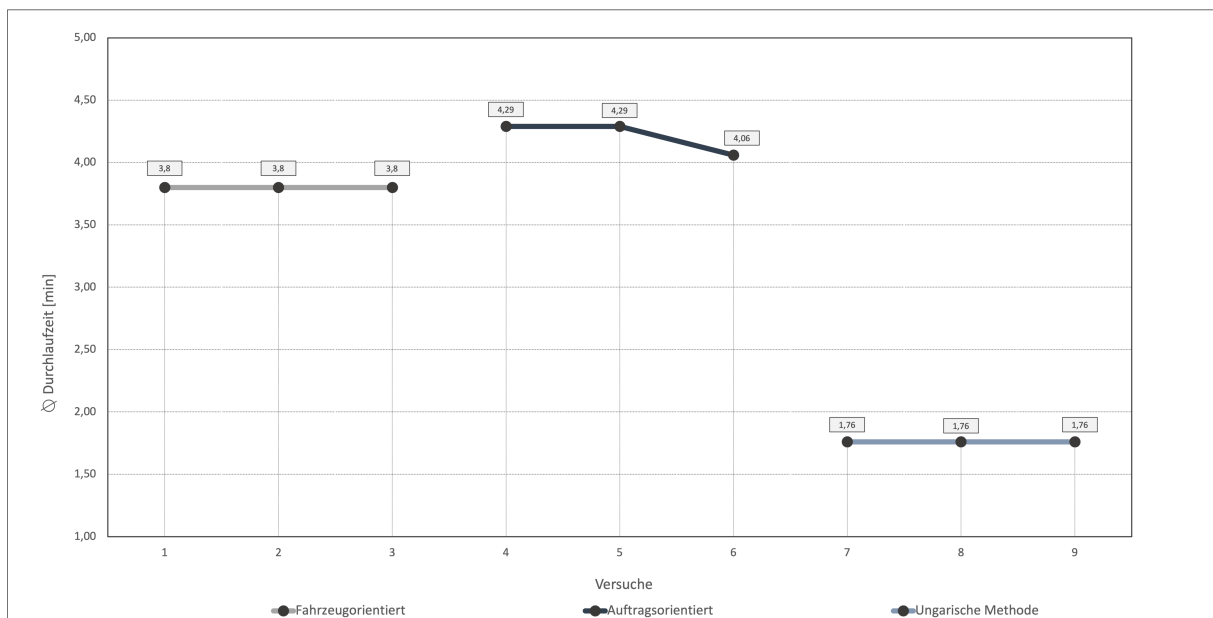


Abbildung 5-3: Durchschnittliche Durchlaufzeit - Versuche 1 bis 9

Direkttransports verbleiben die Transporteinheiten kürzer auf den Fahrzeugen wodurch die durchschnittliche Durchlaufzeit der Aufträge deutlich geringer ist als bei den anderen

Algorithmus-Varianten. Die Verwendung der reinen auftragsorientierten Zuweisung hat sowohl höhere Durchlaufzeiten, als auch eine höhere Fahrzeugauslastung zur Folge.

In Abbildung 5-3 ist der unterschiedliche Einfluss der 2-opt Heuristik zu sehen. Im Zuge des Versuchs Nummer 5 wurde versucht die zurückzulegende Strecke zu optimieren was, aufgrund der vergleichsweise geringen Nutzung der Routenzüge, keinen Einfluss auf die Durchlaufzeit hat. Bei Versuch Nummer 6 hingegen wurde diese gezielt optimiert wodurch diese auch verringert werden konnte.

Im Falle eines überlasteten Systems, also einer zu hohen Auftragsrate, erreicht die Ungarische Methode eine minimal höhere Leistung als die reine fahrzeugorientierte Auftragszuweisung. Die Abbildungen 5-4 bis 5-6 zeigen die entsprechenden Versuchsergebnisse.

Bei allen Versuchen ist eine Leistungsverbesserung bei Verwendung der 2-opt Heuristik zu

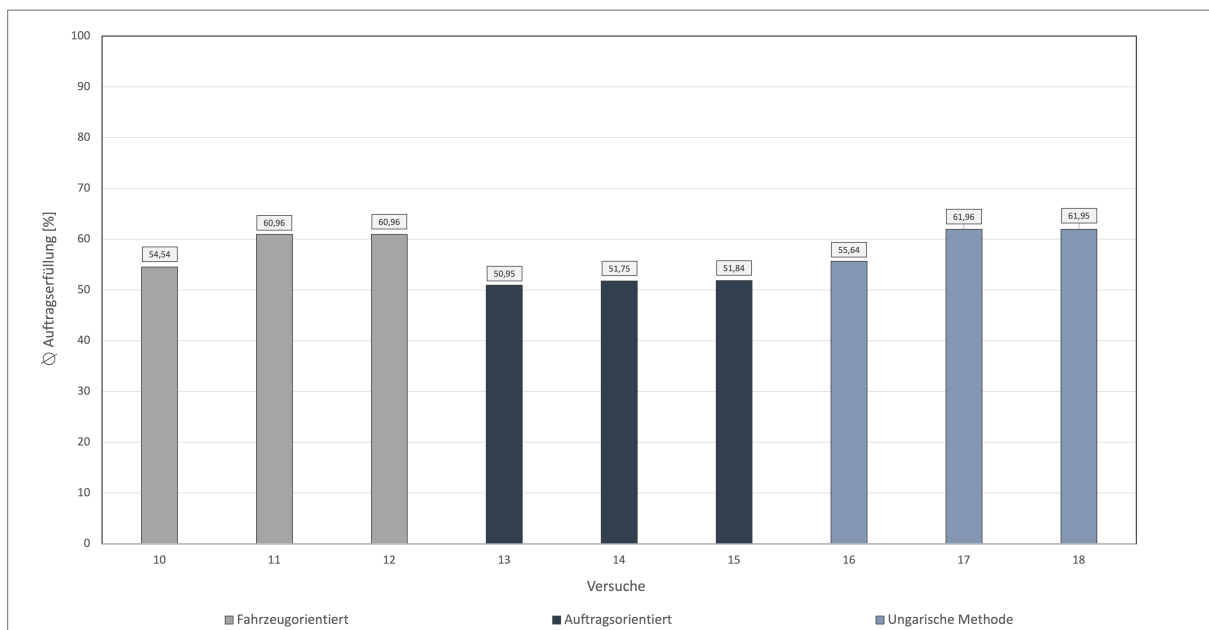


Abbildung 5-4: Durchschnittlicher Grad der Auftragserfüllung - Versuche 10 bis 18

erkennen. Diese Verbesserungen sind jedoch, durch die geringe Anzahl an verwendeten Routingzügen, bei der Ungarischen Methode auf 6,31% und bei der fahrzeugorientierten Zuweisung auf 6,42% begrenzt.

Aufgrund der zufälligen Auftragsverteilung bei der reinen fahrzeugorientierten Zuweisung, werden die Kapazitäten der Routingzüge nur selten komplett ausgelastet wodurch die 2-opt Heuristik die Leistung um weniger als 1% verbessern konnte.

Bei Versuch 18 wurde die maximale Auftragserfüllung (61,95%), welche bei der ent-

sprechenden Fahrzeugaufteilung möglich ist, erreicht. Die Verwendung einer reinen auftragsorientierten Zuweisung führt zwar zu den geringsten Durchlaufzeiten, jedoch auf Kosten einer deutlich geringeren Leistung, verglichen mit den anderen Algorithmus-Varianten.

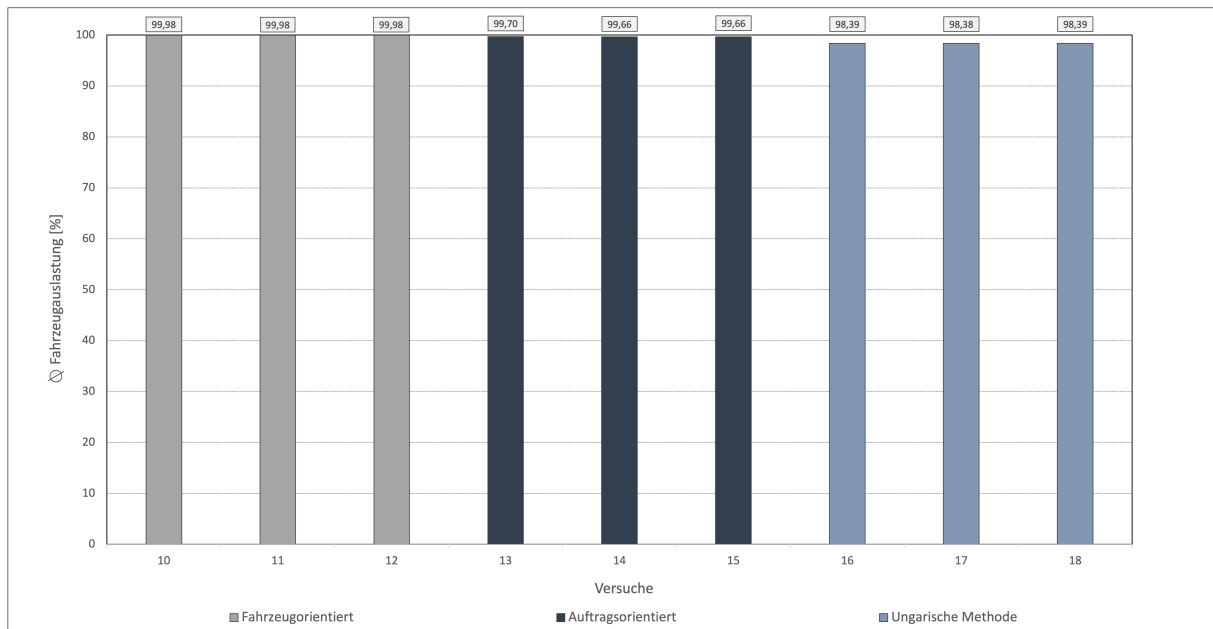


Abbildung 5-5: Durchschnittliche Fahrzeugauslastung - Versuche 10 bis 18

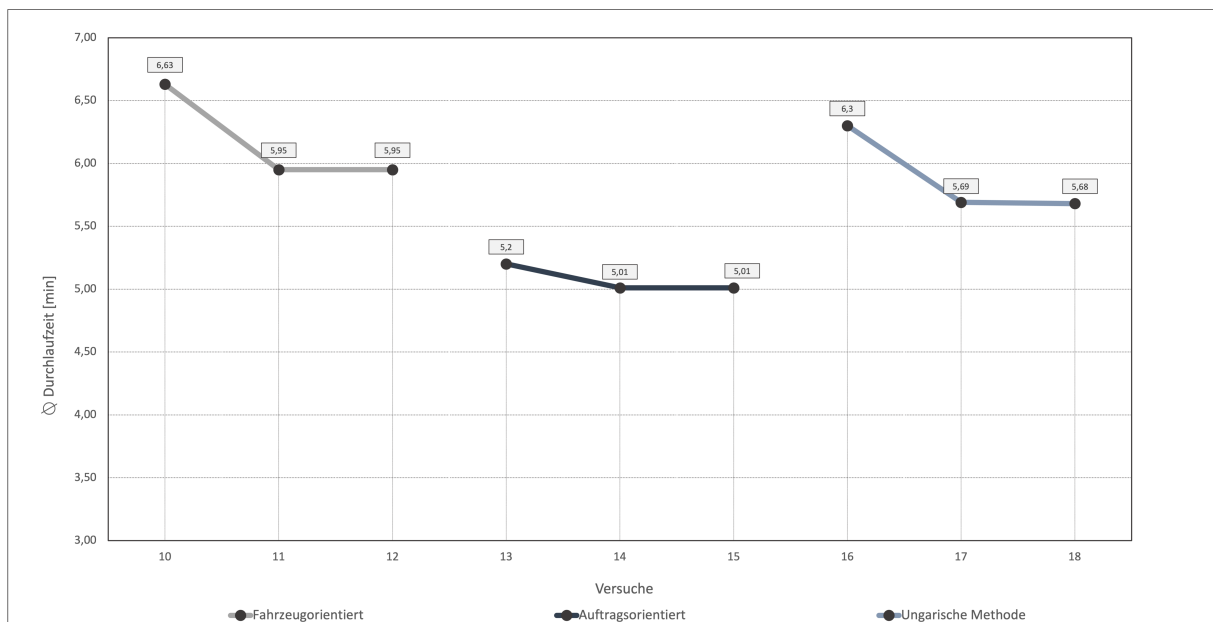


Abbildung 5-6: Durchschnittliche Durchlaufzeit - Versuche 10 bis 18

Für die Versuche 19 bis 27 wurde das System nun so ausgelegt, dass bei Verwendung des

Ungarischen Algorithmus in Kombination der 2-opt Heuristik und einer Fahrzeugauslastung von etwa 96% gerade die Auftragsrate erreicht wird. Die entsprechenden Versuchsergebnisse sind in den Abbildungen 5-7 bis 5-9 dargestellt. Bei Verwendung der fahrzeugorien-

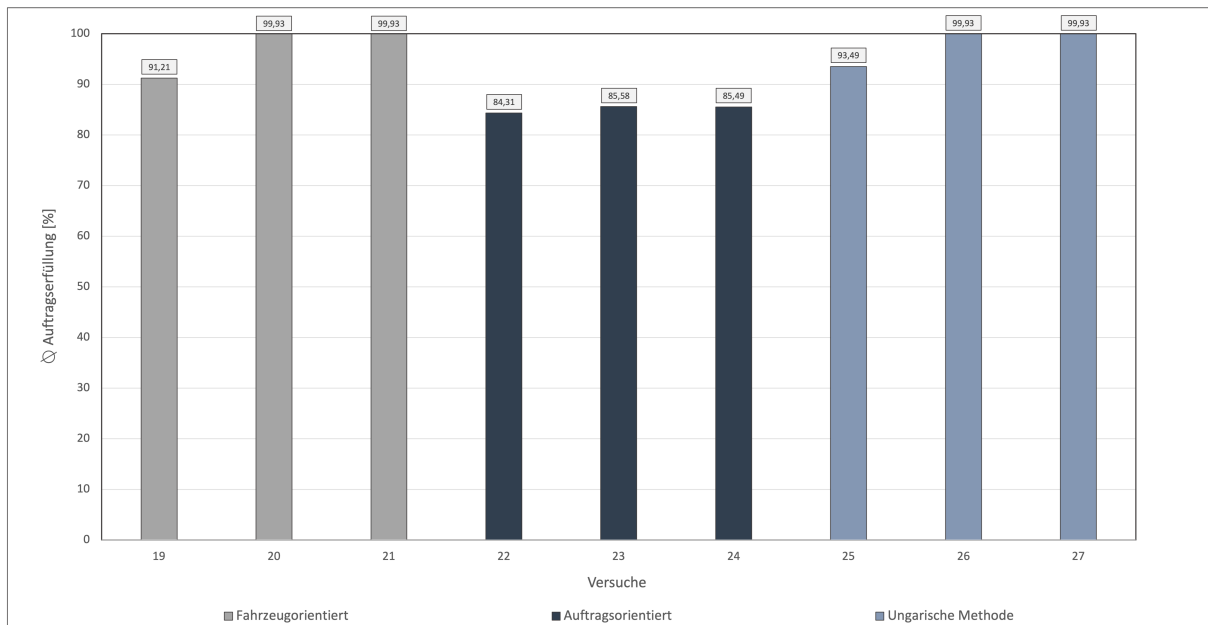


Abbildung 5-7: Durchschnittlicher Grad der Auftragserfüllung - Versuche 19 bis 27

tierten Zuweisung zusammen mit der 2-opt Heuristik wird die Leistung ebenfalls erreicht, der entscheidende Unterschied ist jedoch die daraus resultierende Fahrzeugauslastung, welche bei einer Auftragszuweisung durch die Ungarische Methode geringer ist.

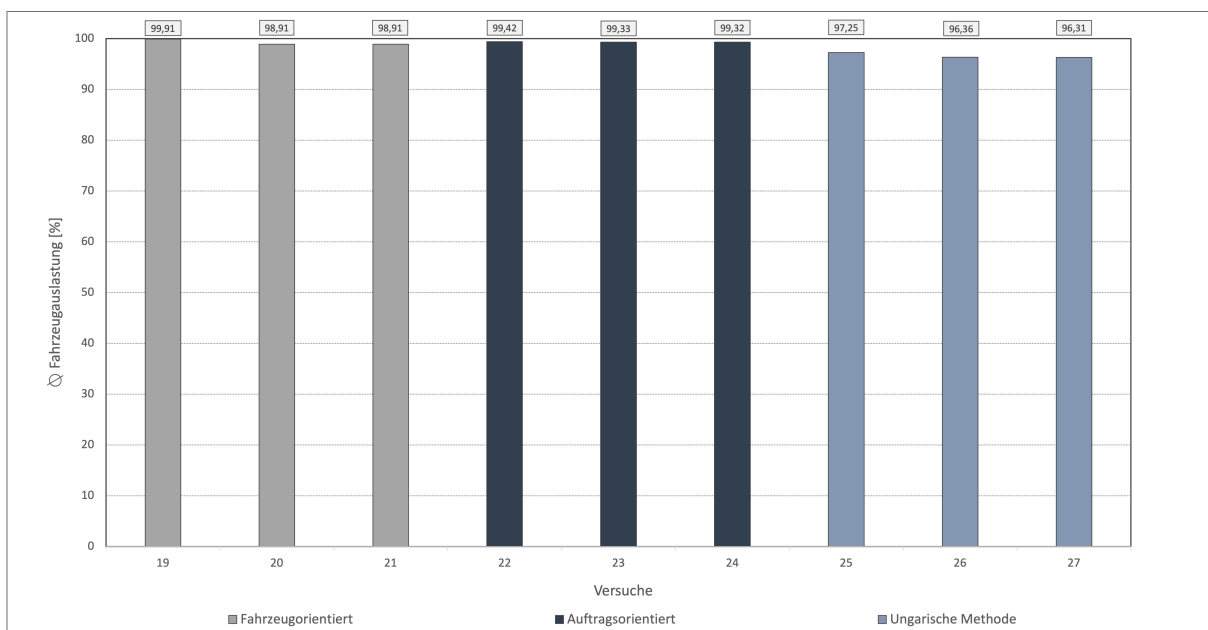


Abbildung 5-8: Durchschnittliche Fahrzeugauslastung - Versuche 19 bis 27

Eine Fahrzeugauslastung nahe den 100 Prozent lässt keinerlei Spielraum für mögliche Störungen und sollte daher möglichst vermieden werden. Betrachtet man den Einfluss der

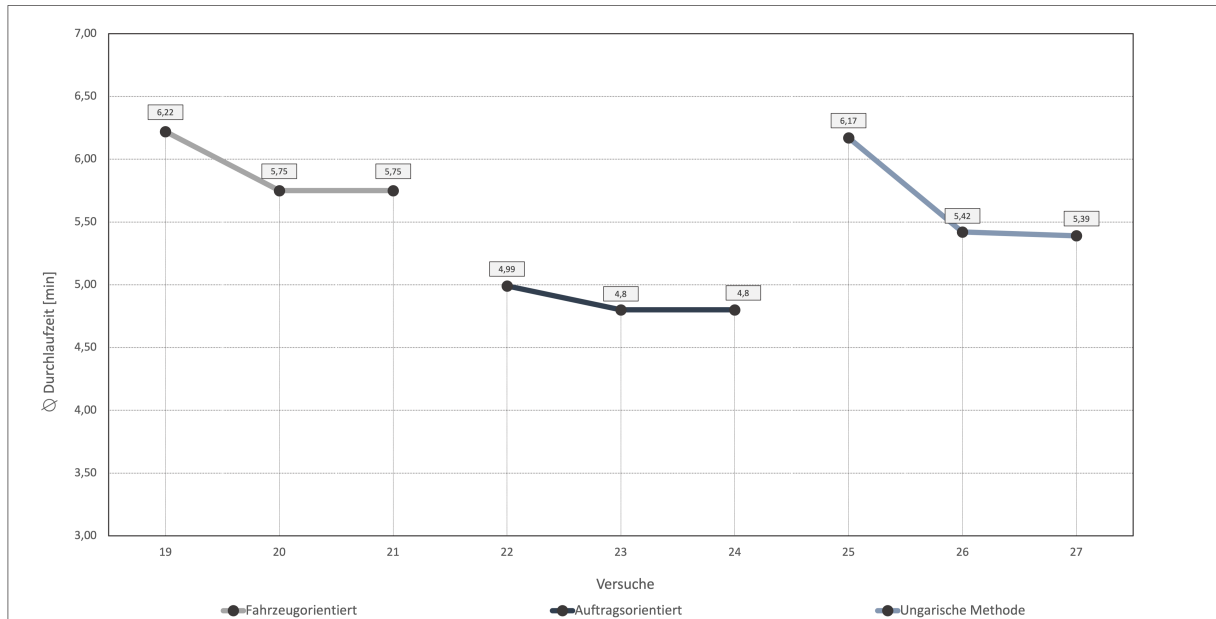


Abbildung 5-9: Durchschnittliche Durchlaufzeit - Versuche 19 bis 27

2-opt Heuristik auf die Auftragserfüllung bei der Zuweisung durch die Ungarische Methode, erkennt man nicht nur eine deutliche Leistungssteigerung, sondern ebenfalls eine nicht unerheblichen Reduzierung der Durchlaufzeit. Die durchschnittliche Durchlaufzeit bei der auftragsorientierten Zuweisung ist zwar geringer, aber wie zuvor nur auf Kosten von hohen Leistungseinbußen sowie einer Fahrzeugauslastung von 100%.

5.1.2 Vermehrter Einsatz von Routenzügen

Für die Versuche 28 - 54 wurden nun die Auswirkungen eines vermehrten Einsatzes von Routenzügen untersucht. Es ist zu erwarten, dass die erreichbare Leistung deutlich höher ist als zuvor und dass die 2-ppt Heuristik einen noch größeren Einfluss hierauf hat.

In den Abbildungen 5-10 bis 5-12 sind die Versuchsergebnisse bei einem nicht ausgelastetem System aufgeführt. Abgesehen von Versuch 31, auftragsorientierte Zuweisung ohne jegliche Optimierung, erreichen alle Algorithmus Kombinationen die geforderte Leistung. Da die Versuche immer direkt nach Ablauf von 5 Tagen stoppen, gibt es immer eine bestimmte Anzahl an Aufträgen welche sich zu diesem Zeitpunkt in der Auslieferung befinden.

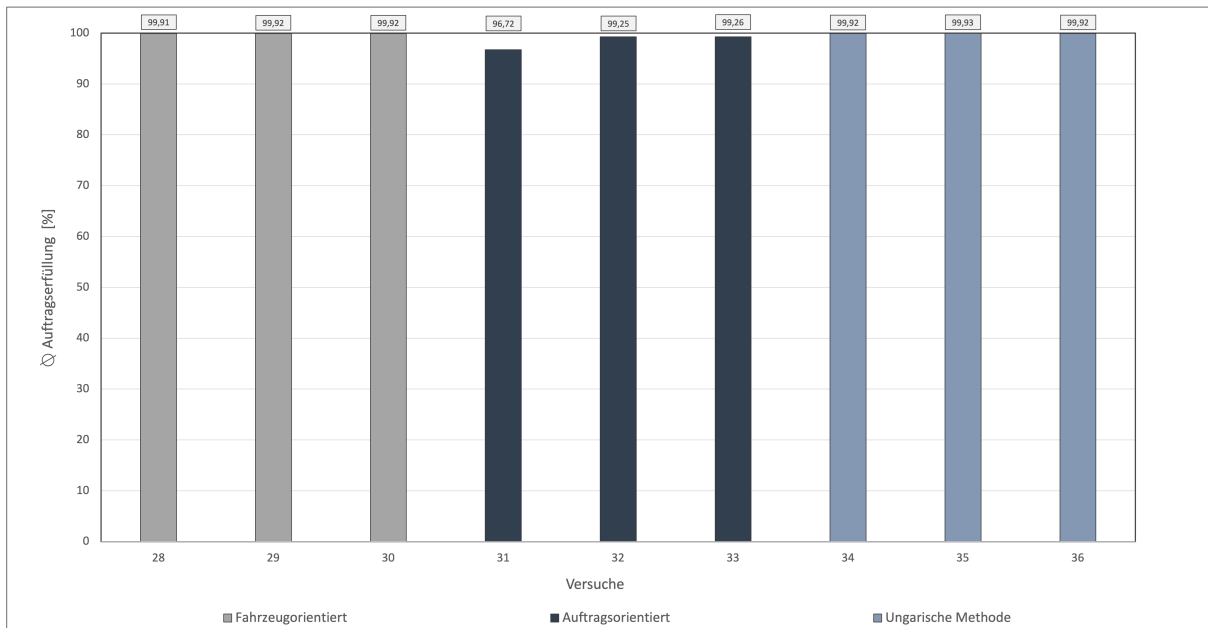


Abbildung 5-10: Durchschnittlicher Grad der Auftragserfüllung - Versuche 28 bis 36

Vergleicht man die Fahrzeugauslastungen welche bei der Verwendungen der einzelnen Algorithmus-Varianten resultieren, erkennt man, dass die fahrzeugorientierte Zuweisung am schlechtesten abschneidet. Grund hierfür ist wie zuvor die zufällige Zuweisung der Fahrzeuge wodurch deren Kapazitäten nicht optimal genutzt werden. Die fahrzeugorientierte Zuweisung schneidet hierbei knapp vor der Ungarischen Methode am besten ab. Das Gleiche gilt für die durchschnittliche Durchlaufzeit. Bei der gewählten Auftragsrate und

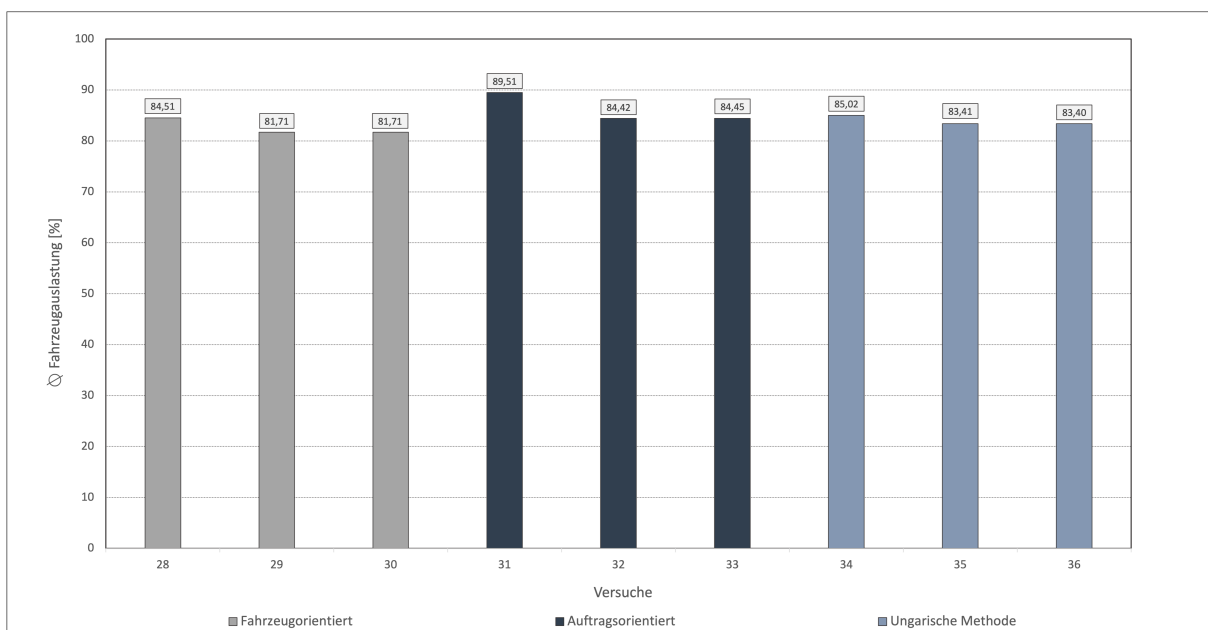


Abbildung 5-11: Durchschnittliche Fahrzeugauslastung - Versuche 28 bis 36

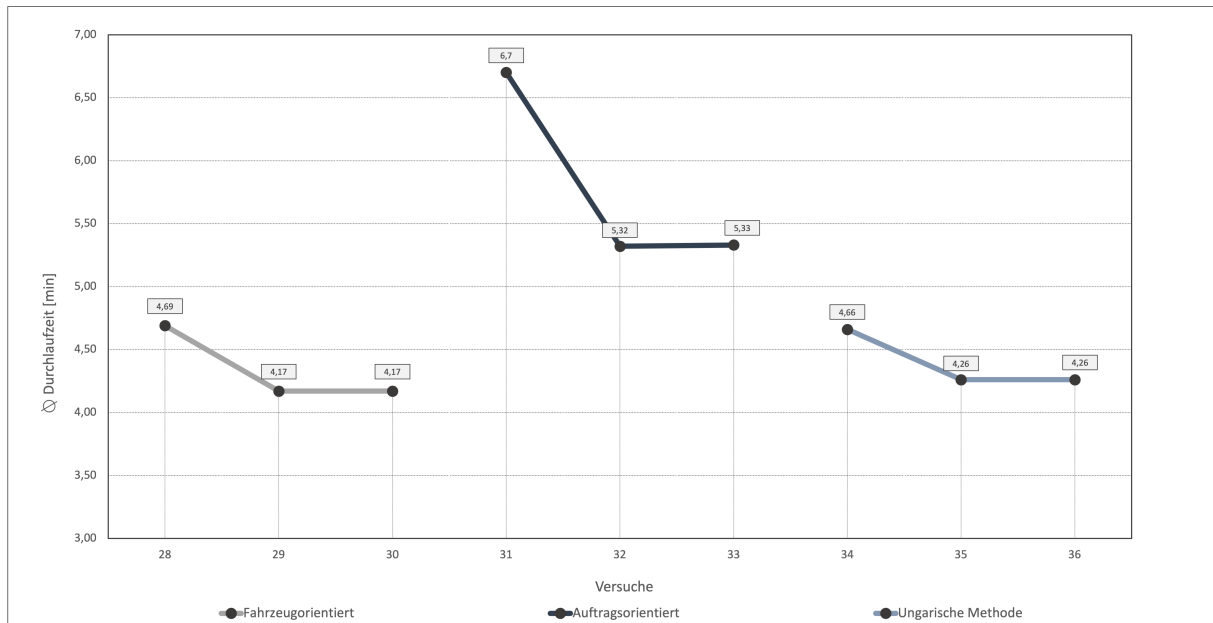


Abbildung 5-12: Durchschnittliche Durchlaufzeit - Versuche 28 bis 36

Fahrzeugtypenaufteilung sind die Versuchsergebnisse der fahrzeugorientierten Zuweisung und die bei Verwendung der Ungarischen Methode näherungsweise identisch.

Abbildungen 5-13 bis 5-15 zeigen nun die Versuchsergebnisse bei einem deutlich überlastetem System. Es wird deutlich, dass bei Verwendung der Ungarischen Methode eine höhere Leistung erzielt werden kann. Die Leistungssteigerungen aufgrund des Einsatzes

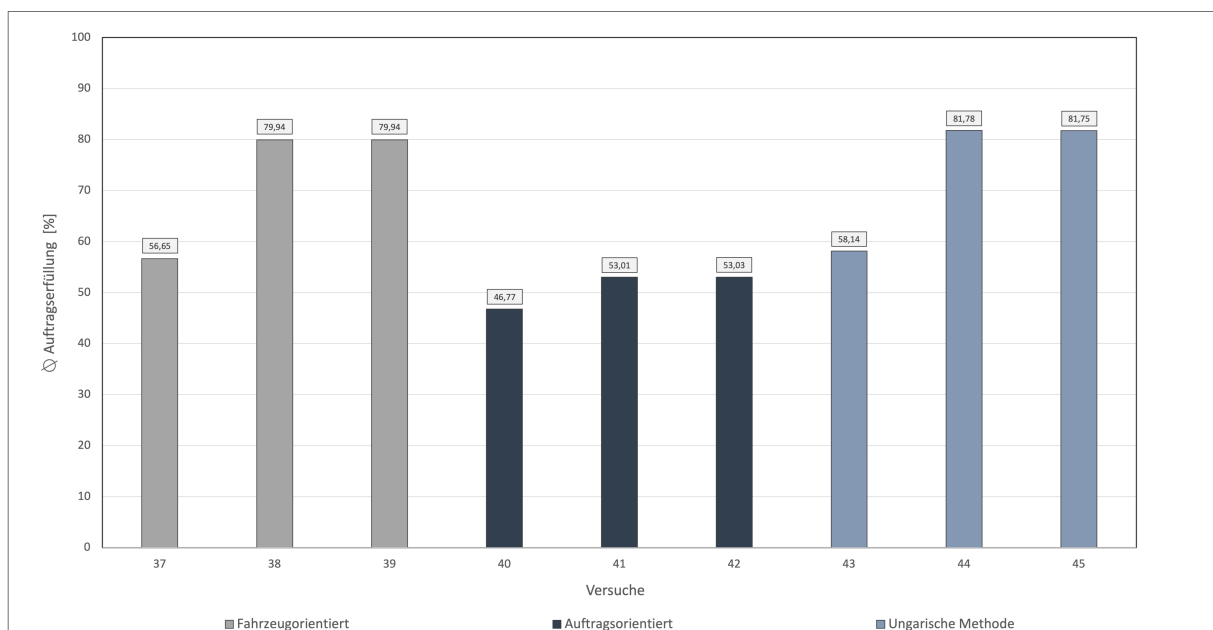


Abbildung 5-13: Durchschnittlicher Grad der Auftragserfüllung - Versuche 37 bis 45

der 2-opt Heuristik sind jedoch deutlich höher. Dies war aufgrund der hohen Anzahl an Routenzügen zu erwarten. Die niedrige Durchlaufzeit bei Verwendung einer auftragsorien-

tierten Zuweisung ohne Optimierung ist zwar deutlich geringer als bei den anderen beiden Algorithmus-Varianten, sollte jedoch nicht zu hoch bewertet werden. da die erreichbare Leistung dieser Alternative deutlich geringer ist. Die Kombination der Ungarischen Methode zusammen mit der 2-opt Heuristik führt neben der höheren Leistung, auch gleichzeitig zu niedrigeren Durchlaufzeiten verglichen mit der fahrzeugorientierten Zuweisung.

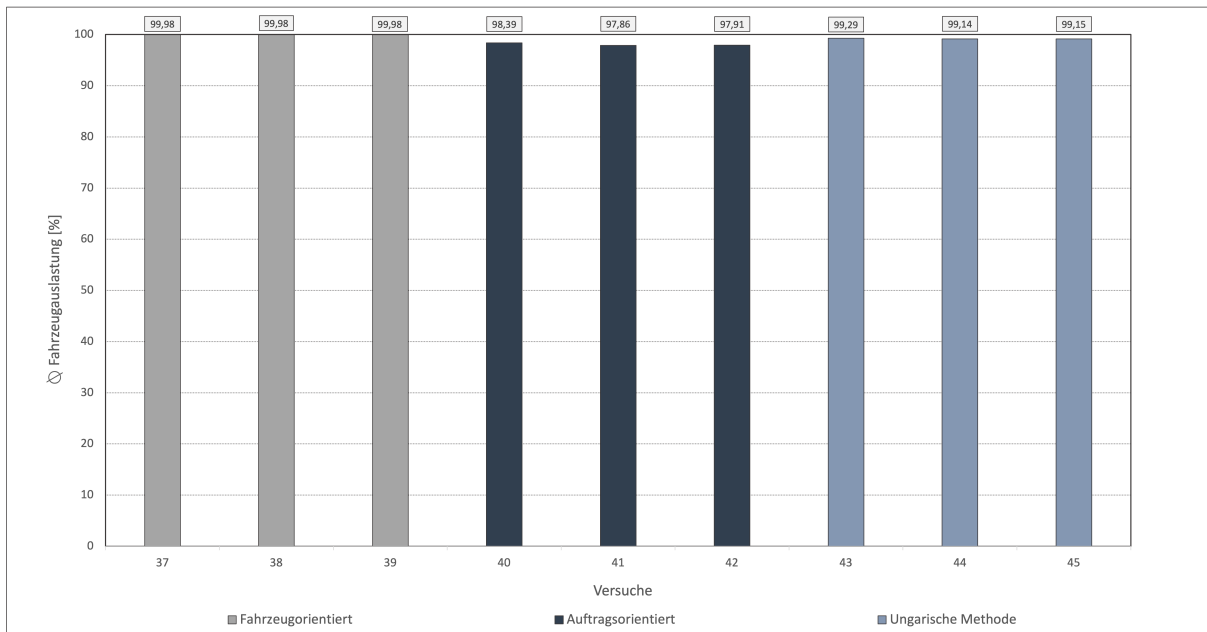


Abbildung 5-14: Durchschnittliche Fahrzeugauslastung - Versuche 37 bis 45

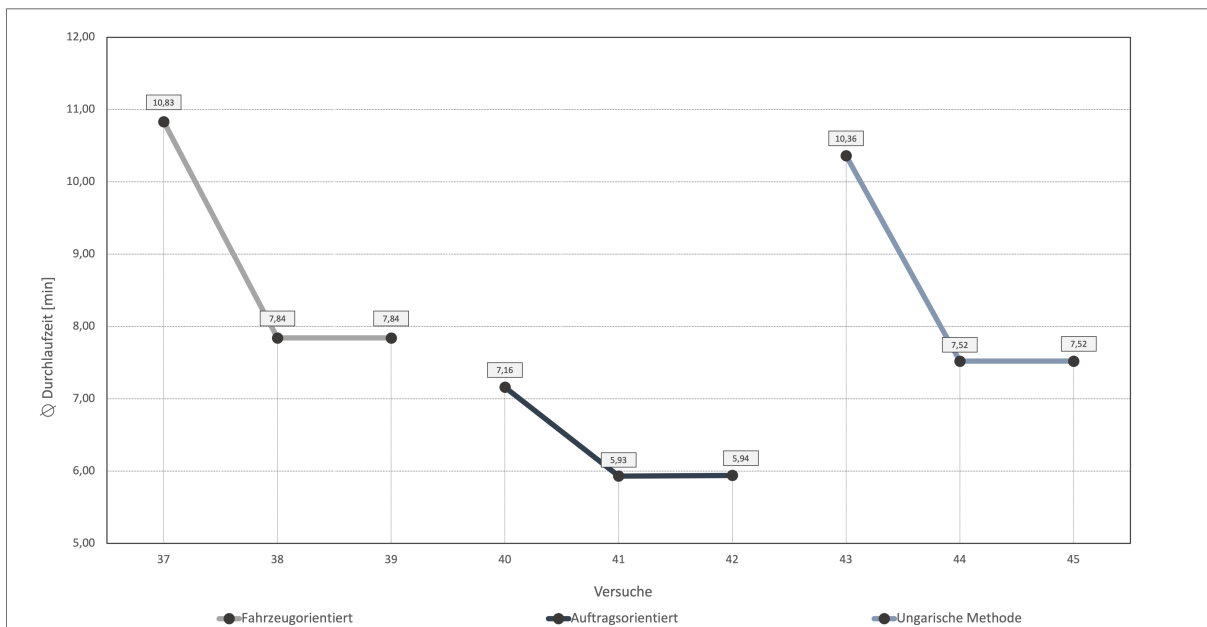


Abbildung 5-15: Durchschnittliche Durchlaufzeit - Versuche 37 bis 45

Zuletzt wurde nun erneut das System so ausgelegt, dass bei Verwendung der Ungarischen

Methode sowie der Optimierung der Durchlaufzeit mittels der 2-opt Heuristik gerade die Leistung erreicht wird.

Dargestellt ist dies in den Abbildungen 5-16 bis 5-18. Bei Verwendung der fahrzeugorien-

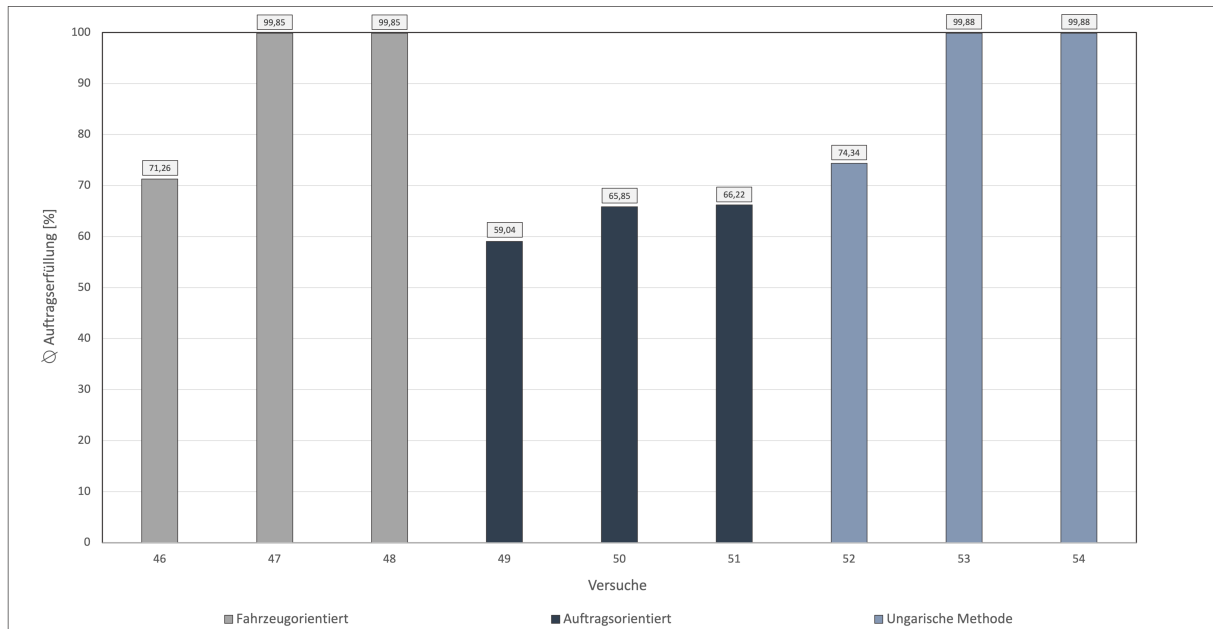


Abbildung 5-16: Durchschnittlicher Grad der Auftragserfüllung - Versuche 46 bis 54

tierten Auftragszuweisung zusammen mit der 2-opt Heuristik wird die geforderte Leistung, bei vergleichbarer Fahrzeugauslastung, ebenfalls erreicht, jedoch mit einer höheren Durchlaufzeit. In Abbildung 5-16 ist die enorme Leistungsverbesserung durch die 2-opt Optimierung zu sehen. Bei der fahrzeugorientierten Zuweisung konnte durch die 2-opt Routenoptimierung die Leistung um fast 30% verbessert werden. Bei Verwendung der Ungarischen Methode konnte die zuvor bereits höhere Leistung um weitere 25% verbessert werden. Diese Verbesserungen der Systemleistungen spiegeln sich ebenfalls in der durchschnittlich benötigten Durchlaufzeit der beiden Varianten wieder. Dargestellt ist dies in Abbildung 5-18. Wie bei den vorherigen Versuchen kann die reine fahrzeugorientierte Auftragszuweisung, mit und ohne Routingoptimierung, die Leistung, trotz einer Fahrzeugauslastung nahe den 100 Prozent, nicht erreichen. Lediglich im Hinblick auf die Durchlaufzeiten hat diese Algorithmus-Variante einen Vorteil gegenüber den Anderen. Ebenfalls auffällig ist die nur geringe Leistungsverbesserung bei Verwendung der 2-opt Heuristik. Dies hängt erneut mit der ungünstigen Zuweisung der Aufträge zusammen wodurch die vorhandenen Kapazitäten der Routingzüge nicht ausgelastet werden. Der Graph in Abbildung 5-17 zeigt zwar eine Auslastung nahe den 100 Prozent, diese beschreibt allerdings nur das Verhältnis zwi-

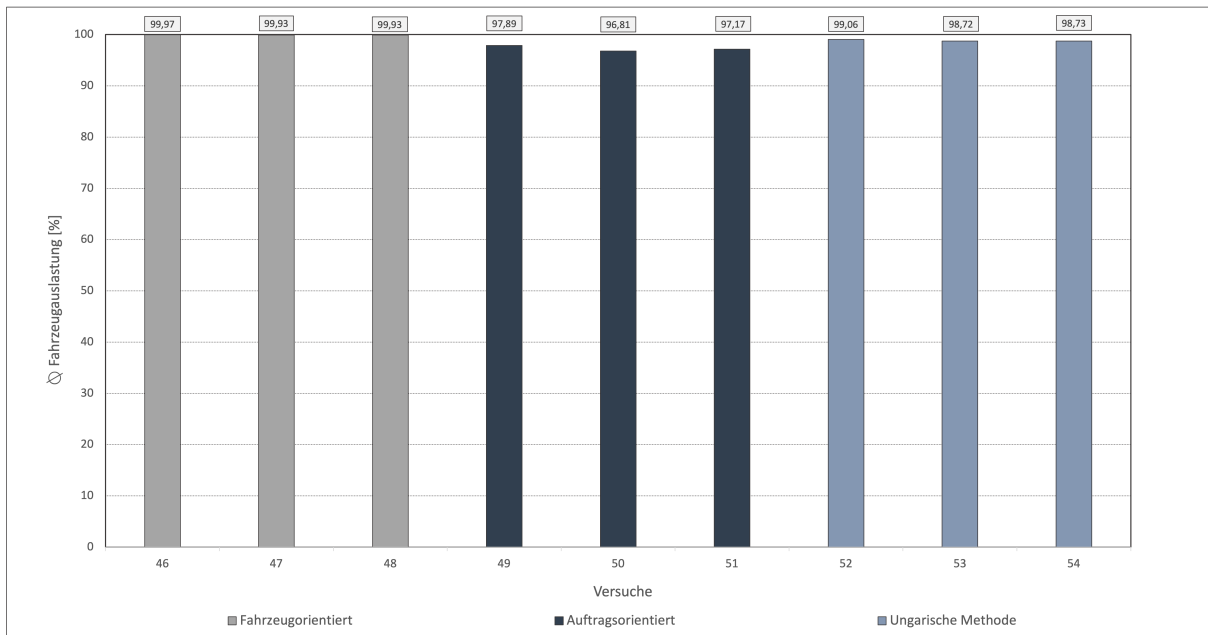


Abbildung 5-17: Durchschnittliche Fahrzeugauslastung - Versuche 46 bis 54

schen der Zeit die das Fahrzeug auf den nächsten Auftrag wartet und dessen Nutzzeit. Die nicht ausgenutzten Kapazitäten der einzelnen Fahrzeuge führt erneut zu deutlich geringeren Durchlaufzeiten. Allgemein betrachtet ist die durchschnittliche Durchlaufzeit bei der vermehrten Nutzung von Routingzügen höher. Dies war zu erwarten, da nur ein Bruchteil der Aufträge mittels Direkttransport bearbeitet werden.

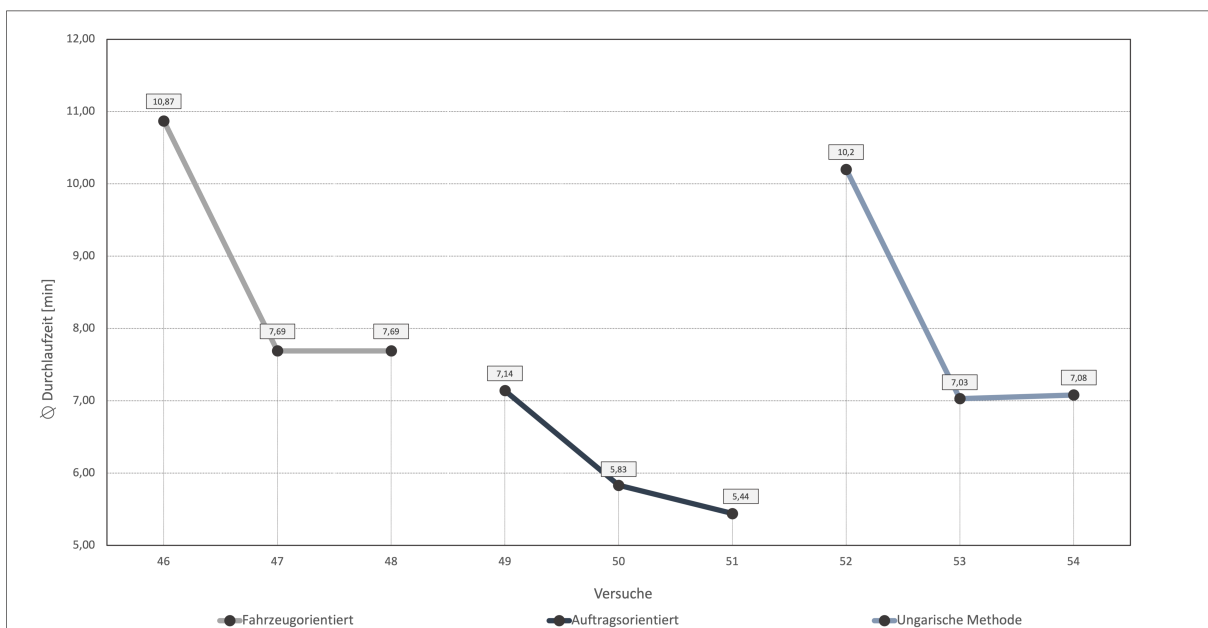


Abbildung 5-18: Durchschnittliche Durchlaufzeit - Versuche 46 bis 54

5.2 Kritische Bewertung

In dem vorliegenden Modell wurde sowohl eine Routingoptimierung durch die 2-opt Heuristik als auch eine Optimierung der Auftragszuweisung mittels der Ungarischen Methode eingeführt. Beide Implementierungen führten, wie dies zu erwarten war, zu Verbesserungen der Systemleistung. Es fällt jedoch auf, dass die Routenoptimierung ein deutlich größeres Verbesserungspotential aufweist als die Optimierung der Auftragszuweisung. Mögliche Gründe hierfür sind die strikte örtliche Trennung der Quellen und Senken, wodurch jedes Transportmittel nach der Lastenabgabe einen vergleichbar langen Weg zurücksetzen muss um die nächste Quelle zu erreichen. Wären hingegen die Quellen und Senken anders verteilt, so hätte die Auswahl der Transportmittel einen größeren Einfluss, da der aktuelle Ort des Fahrzeugs eine weitaus größere Rolle spielen würde.

Beispielsweise könnte sich ein Fahrzeug nach Lastenabgabe an einer Senke, in unmittelbarer Nähe der als nächstes anzufahrenden Quelle befinden. Die Wahl dieses Transportmittels für den jeweiligen Auftrag, verglichen mit einem Fahrzeug welches sich an einer Senke am anderen Ende der Fabrik befindet, hätte eine große Auswirkung auf die Leistung des Systems.

Des Weiteren fällt auf, dass die Leistung des Systems bei Verwendung einer auftragsorientierten Auftragszuweisung ohne jegliche Optimierung geringer ist als die einer reinen fahrzeugorientierten Zuweisung. Der Grund hierfür ist zum Einen, die Häufigkeit der Funktionsaufrufung. Bei der auftragsorientierten Zuweisung wird der Zuweisungs-Algorithmus aufgerufen, sobald ein neuer Auftrag erstellt wurde. Bei der fahrzeugorientierten Zuweisung hingegen, rufen die einzelnen Fahrzeuge die Funktion nach Lastabgabe auf und für den Fall dass keine Aufträge verfügbar sind, tun sie dies jede wartende Sekunde erneut. Zum Anderen wird bei der Auswahl des Transportmittels, ein Zufälliges gewählt um eine mögliche Bevorzugung von Fahrzeugen zu vermeiden. Jedoch führt dies dazu, dass die Routenzüge für ihre Fahrten häufig nicht komplett ausgelastet sind.

Die zur Optimierung der Auftragszuweisung verwendete Ungarische Methode ist zwar eine exakte Methode, jedoch wurde bei der Erstellung der hierfür benötigten Kostenmatrix eine Annahme getroffen. Die einzelnen Kosten, unabhängig davon ob die reine Distanz oder die Durchlaufzeit minimiert werden soll, hängen offensichtlich direkt von der zurückzulegenden Strecke ab.

Bei einem Direkttransport, beispielsweise durch einen Gabelstapler oder einem Fahrerlosen Transportsystem lässt sich diese problemlos berechnen. Die Berechnung der Route bei einem Routenzug hingegen ist deutlich komplexer, denn die Kosten der einzelnen Aufträge sind in diesem Fall nicht unabhängig voneinander. Die Route eines Routenzugs für den Auftrag A ändert sich sobald dem Fahrzeug ein weiterer Auftrag B zugeordnet wird.

Um diese Abhängigkeit zu umgehen wurde die folgende Annahme getroffen:

Da sich die Quellen näherungsweise am selben Ort befinden ist der zurückzulegende Weg vom aktuellen Ort zu eben diesen für alle in Frage kommenden Aufträge ähnlich, wodurch sich diese Strecke mit ausreichender Genauigkeit mitteln lässt. Die Kosten beziehungsweise die zurückzulegenden Strecken zur Bearbeitung eines Auftrags, vereinfachen sich daher zu dem Weg von der Quelle zu der jeweiligen Senke plus dem gemittelten Weg vom aktuellen Ort zu dem Ort der Quellen.

Die durchgeführten Versuche zeigen eine klare Verbesserung bei der Verwendung der Ungarischen Methode, was die getroffene Annahme rechtfertigt, auch wenn dies dadurch nicht zwangsweise immer zu der optimalen Lösung führt.

Die hier erhaltenen Versuchsergebnisse sollten wie bei jeder Simulation, nicht ohne einen kritischen Vergleich des Simulationsmodells mit der Realität bewertet werden, denn ein reales Produktions-, beziehungsweise Logistiksystem ist weitaus komplexer als die meisten Simulationsmodelle. In den durchgeführten Versuchen wurden beispielsweise keine Störungen, Wartungsarbeiten, mögliche Anlaufprobleme bei Schichtwechseln, Lagerkapazitäten oder vorgegebene Lieferzeiten berücksichtigt, um nur ein paar zusätzliche Einflüsse zu nennen.

6 Ausblick

Anhand der im Zuge dieser Arbeit durchgeführten Versuche wird das enorme Verbesserungspotential eines nicht optimierten Systems deutlich. Mittels zweier vergleichsweise simpel implementierten Algorithmen konnte die Auftragserfüllung erhöht und die Fahrzeugauslastung verringert werden, beides Ziele welche ein gewinnmaximierendes Unternehmen verfolgen sollte.

Um die Vergleichbarkeit dieses Simulationsmodell noch zu steigern und dadurch die Wertigkeit der Versuchsergebnisse zu erhöhen, sollten zusätzliche Implementierungen beziehungsweise Abbildungen der Realität in Betracht gezogen werden.

Beispielsweise wird bei der Berechnung der Route durch die 2-opt Heuristik nicht berücksichtigt, dass es auf dieser Route zu Staus durch andere Fahrzeuge kommen könnte, wodurch eine vermeintlich schlechtere Route eventuell die günstigere Alternative gewesen wäre. Eine Möglichkeit dies zu verbessern wäre eine dynamische Routenberechnung.

Interessant wäre weiterhin, welchen Einfluss die Dimension der Kostenmatrix auf die Ergebnisse der Zuweisung hat. Denn je nach Auslastungsgrad kann es vorkommen, dass nur wenige Fahrzeuge verfügbar sind, jedoch eine Vielzahl von Aufträgen und vice versa. Möglicherweise wäre es in einem solchen Fall sinnvoll, abzuwarten bis entweder mehr Fahrzeuge oder mehr Aufträge verfügbar sind, um dadurch ein globaleres Optimum zu finden.

Ebenfalls unberücksichtigt blieben einzuhaltende Liefertermine. Die durchschnittliche Durchlaufzeit wurde zwar minimiert, jedoch bestraft der Algorithmus keine verpassten Liefertermine der jeweiligen Aufträge.

Der Einfluss den die Vereinfachung der Transportkostenberechnung auf die optimale Lösung des Zuweisungsproblems hat, sollte untersucht und eventuell angepasst werden. Zudem wäre es ratsam weitere Algorithmen für die Lösung des Zuweisungsproblems wie auch zur Lösung des Routingproblems zu implementieren und zu vergleichen.

Literaturverzeichnis

- [Cha-1999] Chandra, B.; Karloff, H.; Tovey, C.: New Results on the Old k -opt Algorithm for the Traveling Salesman Problem. In: SIAM J. Comput. 28 (Aug. 1999), S. 1998–2029.
- [DAR-2012] Darmawansyah, T: A Comparative Analysis of Assignment Problem. In: IOSR Journal of Engineering (Aug. 2012).
- [Ede-2011] Edelkamp, S.: Heuristic Search. second. Morgan Kaufmann: 2011.
- [Eng-1999] Englert, M.; Röglin, H.; Vöcking, B.: Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP. In: Algorithmica 68.1 (2014), S. 190–264.
- [Ger-2011] Gerdt, M.: Mathematische Optimierungsverfahren des Operations Research. first. De Gruyter: 2011.
- [Gah-2015] Ghadle, K. P.; Muley, Y. M.: A New Approach to Solve Balance and Unbalance Assignment Problem. In: International Journal of Extensive Research (Apr. 2015).
- [Gri-2018] Grimme, C.: Einführung in die Optimierung. first. Springer Vieweg: 2018.
- [Gün-2014] Günthner, W. A.; Boppert, J.: Lean Logistics. Methodisches Vorgehen und Praktische Anwendung in der Automobilindustrie. first. Springer: 2014.
- [Hei-1979] Heinemeyer, W.: Durchlaufzeiten. In: Kern, W., Handwörterbuch der Produktionswirtschaft (1979), S. 412–434.
- [Hüf-2006] Hüftle, M.: Klassifikation wichtiger heuristischer Methoden. 2006. Url: <http://optim.de/methoden/HeuriOpt/index.htm?3> (besucht am 31. 12. 2021).
- [Iye-2008] Iyer, P.: Operations Research. first. McGraw Hill Education: 2008.
- [Kit-2017] Kitzmann, Q; Brenk, W.: Lean Construction – Das Managementhandbuch. first. Springer Gabler: 2017.
- [Kri-2019] Krieger, W.: Produktionslogistik. 2019. Url: <https://wirtschaftslexikon.gabler.de/definition/produktionslogistik-43280/version-266611> (besucht am 30. 12. 2021).

- [Len-1974] Lenstra, Jan Karel; Kan, Alexander H. G. Rinnooy: Some Simple Applications of the Travelling Salesman Problem. In: Journal of the Operational Research Society 26 (1974), S. 717–733.
- [Mär-2011] März, L.: Simulation und Optimierung in Produktion und Logistik. Praxisorientierter Leitfaden mit Fallbeispielen. first. Springer: 2011.
- [Ner-2019] Neri, F.: Linear Algebra for Computational Sciences and Engineering. second. Springer: 2019.
- [Nil-2003] Nilsson, C: Heuristics for the Traveling Salesman Problem. In: (Jan. 2003).
- [Özn-2019] Öznur, I. G.; Burcu, D.; Dursun, A.: Different Approaches to Solution of The Assignment Problem Using R Program. In: Journal of Mathematics and Statistical Science 5 (Juni 2019), S. 129–145.
- [Rul-1994] Ruland, K. S.: Polyhedral Solution to the Pickup and Delivery Problem. Dissertation. Washington: Sever Institute of Washington University, 1994.
- [Sax-2015] Saxena, P. u. a.: Simulation Tool for Assignment Model: ASSOLVE. In: 340 (Jan. 2015), S. 807–813.
- [See-2010] Seeck, S.: Erfolgsfaktor Logistik. first. Springer Gabler: 2010.
- [Sei-2008] Seib, E.: Data Mining - Methoden in der Simulation. GRIN Verlag: 2008. Url: <https://books.google.de/books?id=tf2vnmKE8-wC>.
- [Suh-2009] Suhl, L.: Optimierungsprobleme. Modelle, Verfahren, Software, Anwendungen. second. Springer: 2009.
- [Tes-2019] Teschemacher, S.: Dynamische Routenzugoptimierung bei kurzfristigen Materialabrufen. Dissertation. München: Lehrstuhl für Betriebswissenschaften und Montagetechnik, Technische Universität München, 2019.
- [Tur-2015] Turau, V.: Algorithmische Graphentheorie. fourth. De Gruyter: 2015.
- [Wen-2000] Wendehals, M.: Qualität. Kostenorientiertes Qualitätscontrolling. first. Deutscher Universitätsverlag, Wiesbaden: 2000.
- [Wil-2013] Willibald, A.: Lean Logistics. Methodisches Vorgehen und praktische Anwendung in der Automobilindustrie. first. Springer Vieweg: 2013.

[Zim-2008] Zimmermann, H.: Operations Research. Methoden und Modelle ; für Wirtschaftsingenieure, Betriebswirte, Informatiker. second. Vieweg: 2008.

Abbildungsverzeichnis

Abbildung 2-1	Magisches Dreieck der Logistik	3
Abbildung 2-2	Einteilung der Logistikbereiche	5
Abbildung 2-3	Komplexitätsklassen in Anlehnung an [Ner-2019, S. 426]	8
Abbildung 2-4	Beispiele für Graphen: (a) ungerichtet, (b) gerichtet, (c) vollständig	8
Abbildung 2-5	Graphenbeispiel eines Logistik Optimierungsproblems; in Anlehnung an [Tes-2019, S. 29]	9
Abbildung 2-6	Kürzeste-Wege-Problem - Graph-Darstellung	10
Abbildung 2-7	Kürzeste-Wege-Problem - Ereignisbaum	11
Abbildung 2-8	Beispiel Problem des Handlungsreisenden	12
Abbildung 2-9	Einteilung von algorithmischen Lösungsverfahren; in Anlehnung an [Hüf-2006]	16
Abbildung 3-2	Schritt 2-5: Optimalitätstest und Erzeugung weiterer Nullen	21
Abbildung 3-1	Schritt 1 und 2: Subtraktion der Zeilen- und Spaltenminima	22
Abbildung 3-3	Finale Zuordnung	22
Abbildung 3-4	Beispielgraph für 2-opt Illustration	24
Abbildung 3-5	Beispielhafter 2-opt Iterationsschritt für die initiale Route in (a)	25
Abbildung 4-1	Schematisches Layout	28
Abbildung 4-2	Flussdiagramm - Fahrzeugorientierte Auftragszuweisung	29
Abbildung 4-3	Flussdiagramm - Auftragsorientierte Auftragszuweisung ohne Optimierung	30
Abbildung 4-4	Flussdiagramm - Fahrzeugorientierte Auftragszuweisung mit Ungarischem Algorithmus	31
Abbildung 4-5	Flussdiagramm - Ungarische Methode	32
Abbildung 4-6	Flussdiagramm - 2-opt Methode	33
Abbildung 5-1	Durchschnittlicher Grad der Auftragserfüllung - Versuche 1 bis 9	38
Abbildung 5-2	Durchschnittliche Fahrzeugauslastung - Versuche 1 bis 9	39
Abbildung 5-3	Durchschnittliche Durchlaufzeit - Versuche 1 bis 9	39

Abbildung 5-4	Durchschnittlicher Grad der Auftragserfüllung - Versuche 10 bis 18	40
Abbildung 5-5	Durchschnittliche Fahrzeugauslastung - Versuche 10 bis 18	41
Abbildung 5-6	Durchschnittliche Durchlaufzeit - Versuche 10 bis 18	41
Abbildung 5-7	Durchschnittlicher Grad der Auftragserfüllung - Versuche 19 bis 27	42
Abbildung 5-8	Durchschnittliche Fahrzeugauslastung - Versuche 19 bis 27	42
Abbildung 5-9	Durchschnittliche Durchlaufzeit - Versuche 19 bis 27	43
Abbildung 5-10	Durchschnittlicher Grad der Auftragserfüllung - Versuche 28 bis 36	44
Abbildung 5-11	Durchschnittliche Fahrzeugauslastung - Versuche 28 bis 36	44
Abbildung 5-12	Durchschnittliche Durchlaufzeit - Versuche 28 bis 36	45
Abbildung 5-13	Durchschnittlicher Grad der Auftragserfüllung - Versuche 37 bis 45	45
Abbildung 5-14	Durchschnittliche Fahrzeugauslastung - Versuche 37 bis 45	46
Abbildung 5-15	Durchschnittliche Durchlaufzeit - Versuche 37 bis 45	46
Abbildung 5-16	Durchschnittlicher Grad der Auftragserfüllung - Versuche 46 bis 54	47
Abbildung 5-17	Durchschnittliche Fahrzeugauslastung - Versuche 46 bis 54	48
Abbildung 5-18	Durchschnittliche Durchlaufzeit - Versuche 46 bis 54	48

Tabellenverzeichnis

Tabelle 2-1	Die 7 Arten von Verschwendungen	5
Tabelle 2-2	Algorithmen zur Lösung von Kürzeste-Wege-Problemen	11
Tabelle 2-3	Algorithmen zum Kreieren von Touren nach	12
Tabelle 2-4	Algorithmen zur Optimierung von Touren nach	13
Tabelle 2-5	Komplexität des Zuordnungsproblems	15
Tabelle 2-6	Algorithmen und Heuristiken zur Lösung von Zuweisungsproblemen	16
Tabelle 3-1	Optimale Zuordnung des Zuweisungsproblems	22
Tabelle 5-1	Versuche zur Evaluierung des Optimierungsalgorithmus - Vermehrt Direkttransport	36
Tabelle 5-2	Versuche zur Evaluierung des Optimierungsalgorithmus - Vermehrt Routenzüge	37

A Anhang

Tabelle A-1: Resultate bei vermehrt Direkttransport und nicht ausgelastetem System

Nr.	Abzuarbeitende Aufträge	Abgearbeitete Aufträge	Ø Auftrags- erfüllung	Ø Fahrzeug- auslastung	Ø Durchlauf- zeit
1	27.116	27.104	99,96%	58,30%	3,80 min
2	27.116	27.102	99,95%	50,77%	3,80 min
3	27.116	27.102	99,95%	50,77%	3,80 min
4	27.116	27.098	99,93%	75,12%	4,29 min
5	27.116	27.101	99,94%	74,10%	4,29 min
6	27.116	27.103	99,95%	74,83%	4,06 min
7	27.116	27.109	99,97%	23,73%	1,76 min
8	27.116	27.109	99,97%	23,73%	1,76 min
9	27.116	27.109	99,97%	23,73%	1,76 min

Tabelle A-2: Resultate bei vermehrt Direkttransport und überlastetem System

Nr.	Abzuarbeitende Aufträge	Abgearbeitete Aufträge	Ø Auftrags- erfüllung	Ø Fahrzeug- auslastung	Ø Durchlauf- zeit
10	81.406	44.399	54,54%	99,98%	6,63 min
11	81.406	44.628	60,96%	99,98%	5,95 min
12	81.406	44.628	60,96%	99,98%	5,95 min
13	81.406	41.472	50,95%	99,70%	5,20 min
14	81.406	42.122	51,75%	99,66%	5,01 min
15	81.406	42.198	51,84%	99,66%	5,01 min
16	81.406	45.292	55,64%	98,39%	6,30 min
17	81.406	50.437	61,96%	98,38%	5,69 min
18	81.406	50.432	61,95%	98,39%	5,68 min

Tabelle A-3: Resultate bei vermehrt Direkttransport und gut ausgelastetem System

Nr.	Abzuarbeitende Aufträge	Abgearbeitete Aufträge	Ø Auftrags- erfüllung	Ø Fahrzeug- auslastung	Ø Durchlauf- zeit
19	48.712	44.429	91,21%	99,91%	6,22 min
20	48.712	48.680	99,93%	98,91%	5,75 min
21	48.712	48.680	99,93%	98,91%	5,75 min
22	48.712	41.071	84,31%	99,42%	4,99 min
23	48.712	41.689	85,58%	99,33%	4,80 min
24	48.712	41.643	85,49%	99,34%	4,80 min
25	48.712	45.541	93,49%	97,25%	6,17 min
26	48.712	48.678	99,93%	96,36%	5,42 min
27	48.712	48.679	99,93%	96,31%	5,39 min

Tabelle A-4: Resultate vermehrtem Einsatz von Routenzügen und nicht ausgelastetem System

Nr.	Abzuarbeitende Aufträge	Abgearbeitete Aufträge	Ø Auftrags- erfüllung	Ø Fahrzeug- auslastung	Ø Durchlauf- zeit
28	43.349	43.312	99,91%	84,51%	4,69 min
29	43.349	43.315	99,92%	81,71%	4,17 min
30	43.349	43.315	99,92%	81,71%	4,17 min
31	43.349	41.925	96,72%	89,51%	6,70 min
32	43.349	43.023	99,25%	84,42%	5,32 min
33	43.349	43.029	99,26%	84,45%	5,33 min
34	43.349	43.313	99,92%	85,02%	4,66 min
35	43.349	43.317	99,93%	83,41%	4,26 min
36	43.349	43.313	99,92%	83,40%	4,26 min

Tabelle A-5: Resultate bei vermehrtem Einsatz von Routenzügen und überlastetem System

Nr.	Abzuarbeitende Aufträge	Abgearbeitete Aufträge	Ø Auftrags- erfüllung	Ø Fahrzeug- auslastung	Ø Durchlauf- zeit
37	107.348	60.812	56,65%	99,98%	10,83 min
38	107.348	85.811	79,94%	99,98%	7,84 min
39	107.348	85.811	79,94%	99,98%	7,84 min
40	107.348	50.220	46,77%	98,39%	7,16 min
41	107.348	56.923	53,01%	97,86%	5,93 min
42	107.348	56.937	53,03%	97,91%	5,94 min
43	107.348	62.416	58,14%	99,29%	10,36 min
44	107.348	87.791	81,78%	99,14%	7,52 min
45	107.348	87.760	81,75%	99,15%	7,52 min

Tabelle A-6: Resultate bei vermehrtem Einsatz von Routenzügen und gut ausgelastetem System

Nr.	Abzuarbeitende Aufträge	Abgearbeitete Aufträge	Ø Auftrags- erfüllung	Ø Fahrzeug- auslastung	Ø Durchlauf- zeit
46	84.678	60.344	71,26%	99,97%	10,87 min
47	84.678	84.553	99,85%	99,93%	7,69 min
48	84.678	84.553	99,85%	99,93%	7,69 min
49	84.678	49.986	59,04%	97,89%	7,14 min
50	84.678	55.751	65,85%	96,81%	5,83 min
51	84.678	56.070	66,22%	97,17%	5,44 min
52	84.678	62.948	74,34%	99,06%	10,20 min
53	84.678	84.576	99,88%	98,72%	7,03 min
54	84.678	84.585	99,89%	98,73%	7,08 min

Erklärung

Ich versichere hiermit, dass ich die von mir eingereichte Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum, Unterschrift