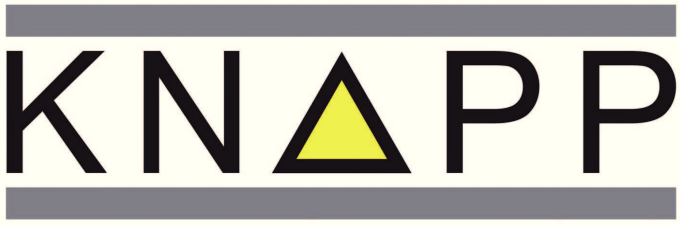
Masterarbeit

Optimierung der Auftragsverteilung eines automatisierten Lagersystems

Konzeptfindung und erste Umsetzung anhand eines Fallbeispiels aus der Praxis

vorgelegt von:	Peter Richard Andrae
Matrikelnummer:	0635227
Universität:	Montanuniversität Leoben
Studienrichtung:	Industrielogistik
Universitärer Betreuer:	Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Norbert Seifter
Institut:	Lehrstuhl für Angewandte Mathematik
Betreuer:	Dipl.-Wirtsch.-Inf. Stephan Wagner
Kooperationspartner:	KNAPP Systemintegration GmbH
Abteilung:	Warehouse Management & Control Systems



Leoben, am 17. September 2012

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Peter Richard Andrae
Matrikelnummer: 0635227
Leoben, 17. September 2012

Danksagung

An dieser Stelle danke ich allen, die mich während meines bisherigen Studiums und speziell bei dieser Arbeit unterstützt haben.

Besonderen Dank möchte ich meiner Familie und meiner Freundin aussprechen, welche mich auf meinem bisherigen Bildungsweg jederzeit moralisch und finanziell unterstützt haben. Somit war es mir möglich, auch die größten Hürden zu überwinden und anfallende Herausforderungen zu meistern.

Weiters danke ich Herrn Professor Norbert Seifter für seine umfangreiche fachliche Betreuung und der Firma KNAPP Systemintegration und meinem Betreuer Stephan Wagner, die mir das Arbeiten an diesem Projekt und somit das Verfassen dieser Arbeit ermöglicht haben.

Nicht zuletzt danke ich meinen Freunden, die mir immer zur Seite stehen.

Kurzfassung

Diese Arbeit untersucht die interne Auftragsverteilung eines zukünftigen Lagersystems der Lebensmittelindustrie für einen Systemanbieter im Bereich Lagerlogistik und Lagerautomation. Es wird bestimmt, welche Vorgangsweise für die Auftragsverteilung zwischen den Kommissioniersystemen innerhalb des Lagersystems, unter Einhaltung definierter Vorgaben, am besten geeignet ist. Als Testdaten dienen ausgewählte Auftrags- und Stammdaten eines vergleichbaren, bestehenden Lagersystems. Die Vorgangsweisen der Verfahren basieren entweder auf den Ideen des Systemanbieters oder den recherchierten Prinzipien der Fachliteratur. Als Ergebnis werden die umgesetzten Verfahren untereinander verglichen, anhand der definierten Ziele bewertet und deren Vor- und Nachteile angeführt. Die aus den Verfahren resultierende Verteilung der Aufträge wird durch Simulation auf den praktischen Einsatz geprüft. Ein Vergleich der neu entwickelten Verfahren zu der geplanten Auftragsverteilung vor Beginn dieser Arbeit bildet den Abschluss. Die Integration der vorgelagerten Touren- und Welleneinteilung führt hin zu einer umfassenderen Optimierung und Fortsetzung dieser Arbeit.

Schlüsselwörter: Optimierung; Lagersystem; Heuristik; Auftragsverteilung

Abstract

The focus of this thesis is the internal order allocation of a warehouse system for the food industry. The warehouse is provided by a company that has its key competences in warehouse logistics and automation technology. The task is to decide which strategy delivers the best allocation result for the involved picking systems in accordance with the defined constraints. The implemented algorithms are tested on real data supplied by a productive warehouse. The basic ideas of the algorithms were either developed by the system provider or taken from the literature. The comparison of the implemented procedures through defined targets is one part of this thesis. The practical usability of the order allocation is tested by simulation programs. As we show in the last section, significant improvements of the performance of the distribution system where achieved through the approaches developed in this thesis. The next step after this thesis would be a more comprehensive optimization including the disposition of departure tours and picking waves.

Keywords: optimization; warehousing; heuristics; order allocation

1. Einleitung

Hinter dieser Arbeit steht ein reales Projekt eines internationalen Systemanbieters im Bereich Lagerlogistik und Lagerautomation. Das Projekt umfasst die Planung, Entwicklung, Installation und Inbetriebnahme eines Distributionszentrums in der Branche Lebensmittelhandel. Der Fokus dieser Arbeit liegt auf der Optimierung der internen Auftragsverteilung im Lagersystem. Die Aufträge werden auf die Kommissionierstationen im Lager verteilt und von diesen abgearbeitet. Es soll eine möglichst gleichmäßige Verteilung der Arbeitslast auf den einzelnen Stationen erreicht werden. Nebenbedingungen verhindern, dass das Lager über seinen Leistungsgrenzen betrieben wird.

Es werden die im Unternehmen des Systemanbieters vorhandenen Ansätze zur Auftragsverteilung analysiert, erweitert und umgesetzt. Vorgangsweisen aus der Fachliteratur, welche für ähnliche Problemstellungen angewandt wurden, werden auf ihre Anwendbarkeit im Projekt untersucht, angepasst und nach Möglichkeit realisiert. Der gesamte Ablauf der Auftragsverteilung wird automatisiert durchgeführt, jedoch sind manuelle Eingriffe bei Bedarf möglich. Durch umfassende Testläufe werden die verschiedenen Strategien untereinander vergleichbar gemacht. Es wird bestimmt, welche Vorgangsweise für die Auftragsverteilung im Projekt, unter Verwendung der zum Zeitpunkt des Verfassens dieser Arbeit verfügbaren Informationen, am besten geeignet ist.

Das Projekt liefert die erforderlichen Testdaten und Informationen auf denen diese Arbeit basiert. Andererseits profitiert das Projekt von den Erkenntnissen und Ergebnissen dieser Arbeit. Wenn in den folgenden Abschnitten vom *Fallbeispiel* gesprochen wird, ist dieses Projekt gemeint.

Das Glossar wurde teilweise mit Hilfe der Nachschlagewerke [21] und [25] erstellt. Bei wörtlicher Zitierung ist die Seitenzahl des Werkes angegeben. Der Name des Glossar-Eintrags ist gleichzeitig das Stichwort unter dem im Nachschlagewerk weitere Informationen gefunden werden können.

2. Das Lagersystem

Das zu verbessernde System wird trivial als Lager bezeichnet. Verschiedenste Artikel werden am Wareneingang (WE) vereinnahmt, verbleiben eine bestimmte Zeitdauer im System und durchlaufen diverse Lagerprozesse, um zu guter Letzt am Warenausgang das Lager termingerecht zu verlassen. Das Lager beinhaltet ein breites Artikelspektrum von mehreren Tausend unterschiedlichen Artikeln und das Lagersortiment ändert sich über die Zeit. Da das Lager Kosten verursacht, wird versucht, die darin ablaufenden Prozesse so effizient wie möglich zu betreiben. Der Lagerbestand wird so niedrig wie möglich gehalten. Schnell verderbliche Waren können nur sehr kurz in den dafür vorgesehen, gekühlten Lagerbereichen gelagert werden. Im Idealfall erreichen die Waren das Lager erst kurz bevor sie benötigt werden.

Genauer wird das Lager als Distributionszentrum (DZ) klassifiziert. In der Fachliteratur ist sehr oft die englische Bezeichnung Distribution Center (DC) und deren Abkürzung zu finden. Das Lager stellt somit einen Knotenpunkt in der logistischen Kette dar. Ein zuliefernder Lastkraftwagen (LKW) befördert wenig verschiedene Artikel in großer Menge zum DZ (diese Waren kommen von vorgelagerten Knoten der logistischen Kette). Im Lager werden die vereinnahmten Artikel durch die Lagerprozesse in kleinere Verbundeinheiten zerteilt, gelagert, sortiert und bei Bedarf kommissioniert. LKWs, die das Lager verlassen, haben verschiedenste Artikel in unterschiedlichen Mengen geladen und beliefern die Einzelhandelsfilialen (nachgelagerten Knoten der logistischen Kette). Somit kann auch von einem Einzelhandels Distributionszentrum oder Retail Distribution Center (RDC) gesprochen werden. Mehrere Einzelhandelsfilialen werden logisch (z.B. anhand geografischer Eigenschaften) zu einer Tour zusammengefasst.

Das Lagersystem und seine Umgebung außerhalb der Systemgrenzen ist sehr vereinfacht in Abbildung 1 dargestellt. Die Fabriken (oder Lieferanten) A,B und C erzeugen (oder liefern) die Artikel a, b und c, welche mittels LKWs zum DZ transportiert werden. Im Lager erfolgen die Lagerprozesse, welche die Artikel nach Filialaufträgen neu gruppieren und zu Touren zusammenfassen. Die Filialen einer Tour werden mittels LKWs beliefert. Die Hauptaufgabe des DZ ist somit nicht die Lagerung der Artikel, sondern vielmehr die Transformation des Ladungsträgers (z.B. von Palette auf einen Rollcontainer) und das Kommissionieren von Aufträgen der Einzelhandelsfilialen.

Im *Fallbeispiel* handelt es sich um ein RDC des Lebensmitteleinzelhandels, welches sowohl Lebensmittel als auch Nicht-Lebensmittel an seine Einzelhandelsfilialen verteilt. Besonders bei schnell verderblichen Lebensmitteln mit Mindesthaltbarkeitsdatum (MHD) ist eine längere Lagerung nicht möglich.

2.1. Aufbau des Lagers

Dieser Abschnitt beschreibt das Zusammenspiel der Teilsysteme des Lagers, welche für diese Arbeit von Relevanz sind. Dem Leser wird veranschaulicht, wie sich ein Artikel durch das Lager bewegt und welche Stationen und Prozesse er dabei passiert, um schlussendlich das Lager mit seinem neuen Ziel zu verlassen.

Innerhalb des Lagersystems existieren mehreren Komponenten, welche miteinander in

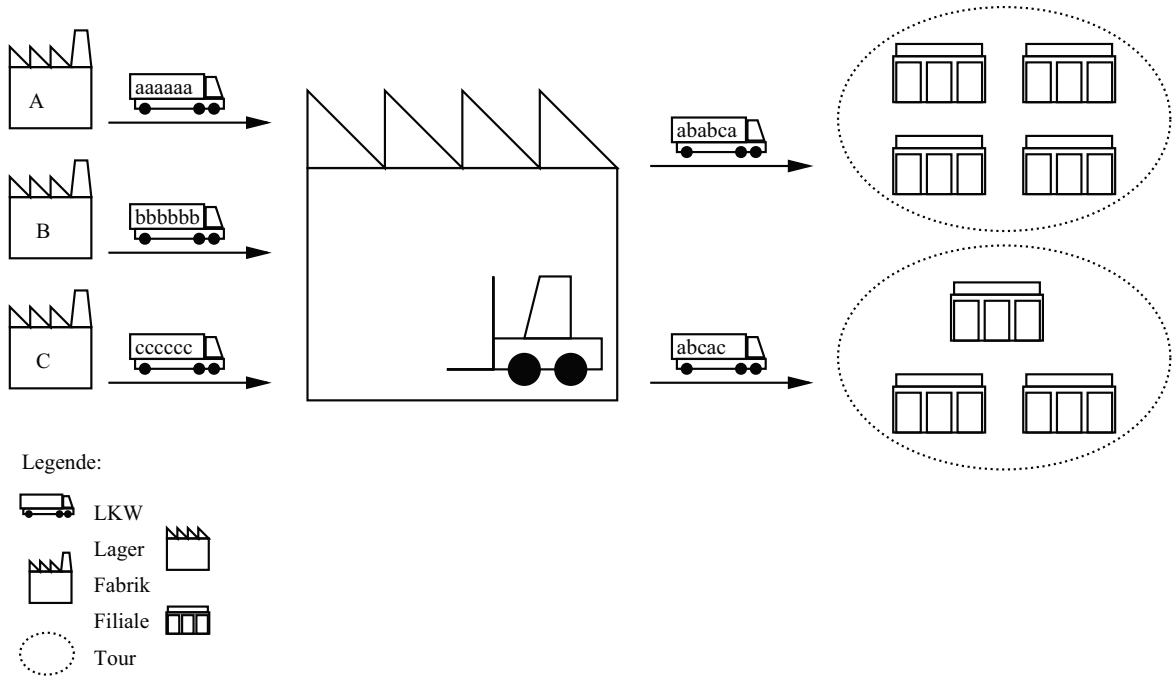


Abbildung 1: Lagersystem

Wechselbeziehung stehen. Für diese Arbeit ist nur ein Teil der Komponenten relevant, da die Optimierung nur einen Teil des realen Lagersystems betrifft. Diese Komponenten visualisiert Abbildung 2 und sie sind im folgenden Text näher beschrieben. Die farbigen Pfeile stellen den Warenfluss der Artikel auf deren Ladungsträgern (schwarzer Pfeil: Palette, roter Pfeil: Tray, grauer Pfeil: Rollcontainer, weißer Pfeil: Leerpalette) zwischen den einzelnen Komponenten dar.

Der erste zu betrachtende Bereich im Lager ist der Wareneingang und der Startpunkt jedes neu im Lager eintreffenden Artikels. Artikel werden im Wareneingang auf Paletten vereinnahmt und auf die Palettenfördertechnik gestellt. Die Fördertechnik transportiert die Paletten in der Regel zum Hochregallager (HRL) oder, falls die Artikel schon benötigt werden, direkt zum zugeteilten Kommissioniersystem. Es besteht die Möglichkeit, ganze Paletten direkt in den Warenausgang zu transportieren, falls eine Auftragszeile entsprechend viele Artikel beinhaltet und die zugehörige Einzelhandelsfiliale „Ganzpaletten“ akzeptiert. In diesem Spezialfall ist für die betroffenen Artikel der Schritt des Kommissionierens hinfällig und die Kommissioniersysteme werden entlastet.

Das (Paletten-)Hochregallager beinhaltet nur „artikelreine“ Paletten. Das heißt, auf einer Palette befinden sich mehrere Stück von einem Artikel einer Ausprägung. Die Ausprägung eines Artikels bestimmt im *Fallbeispiel* das MHD und die Charge. Diese Paletten sind entweder noch völlig mit Waren befüllt oder bereits angebrochen. Anbruchpaletten entstehen, wenn Paletten bei nachfolgenden Stationen (vom Palettenkommissioniersystem oder der Depalettierung) nicht komplett abgeräumt werden. Komplett abgeräumte Paletten verlassen das System als Leerpaletten.

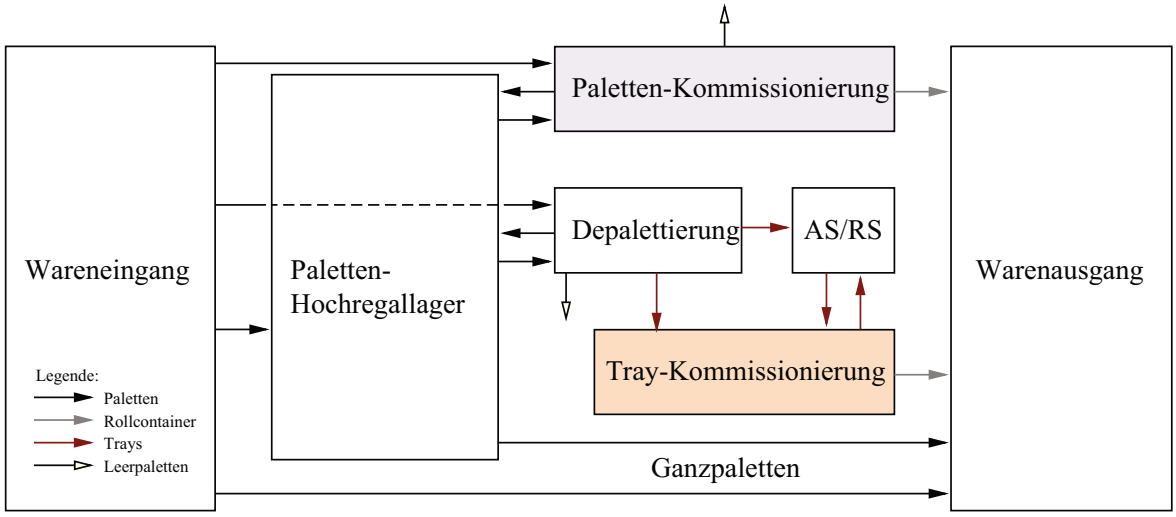


Abbildung 2: Schematischer Aufbau

Für diese Arbeit sind die Kommissioniersysteme die nach Wichtigkeit am höchsten einzustufenden Komponenten des Lagersystems. Sie unterscheiden sich durch ihre spezifischen Eigenschaften (wie z.B. Art der Ladungsträger, Kommissionierrate, Art wie die zu kommissionierenden Artikel zugeführt werden, Liste von Artikeln die auf dem System nicht kommissioniert werden können usw.). Die angesprochenen Kommissioniersysteme sind im konkreten *Fallbeispiel* das Palettenkommissioniersystem (PKS) und das Traykommissioniersystem (TKS), welche sich namentlich aufgrund des Ladungsträgers unterscheiden und beide aus mehreren Stationen bestehen.

Die Optimierung an sich wird in Abschnitt 3 beschrieben. Folgende Informationen werden dem Leser aus Gründen der besseren Verständlichkeit an dieser Stelle gegeben. Bei der Optimierung wird der Ablauf der Auftragsverteilung des Lagers analysiert und verbessert. Es handelt sich dabei um alle Aufträge der Einzelhandelsfilialen, welche zwischen den Kommissioniersystemen und -stationen aufgeteilt werden müssen. Ferner wird bei der Optimierung verstärkt auf das Palettenkommissioniersystem eingegangen, da es sich dabei um eine Neuentwicklung mit wenigen Erfahrungswerten handelt und deshalb mit größeren Optimierungspotentialen zu rechnen ist. Durch eine ungünstige Wahl von zu kommissionierenden Auftragszeilen im PKS wird dessen Leistung stark beeinträchtigt. Wechselwirkungen zu benachbarten Komponenten des Lagersystems erhöhen das Risiko, die Gesamtleistung des Lagers zu senken, bis hin zu Überlastung und Stillstand. Das Traykommissioniersystem wird für diese Arbeit als bereits optimiertes Teilsystem angesehen.

Generell können Artikel auf beiden Kommissioniersystemen bearbeitet werden. Die Palettenkommissionierung (PK) ist für Auftragszeilen mit großen Mengen bzw. Volumina gedacht, da die Artikel direkt von der Palette entnommen werden. Schnell drehende Artikel entsprechen diesem Schema. Viele verschiedene Artikel in kleinen Mengen sind von Nachteil, da jeder Wechsel zu einem neuen Artikel einen Palettenwechsel zur Folge hat. Eine Palettenwechsel ist ein manueller und zeitintensiver Arbeitsschritt. Die Arti-

kel kommen auf Paletten entweder direkt aus dem Wareneingang oder dem HRL. Die Transportgeschwindigkeit der Paletten auf der Fördertechnik im PKS ist sehr langsam im Vergleich zu den Ladungsträgern im TKS.

Die Traykommissionierung (TK) ist sehr flexibel einsetzbar und somit vorrangig für Eilaufträge und Auftragszeilen mit kleinen Auftragsmengen vieler, verschiedener Artikel konzipiert. Auftragszeilen mit langsam drehenden Artikeln und Sonderartikeln entsprechen diesen Eigenschaften. Die Waren werden zuvor von den Paletten entnommen und auf kleinere Ladungsträger (Tray genannt) verteilt. Diesen Prozess erledigt die Depalettierung, welche wiederum vom Wareneingang oder HRL mit Paletten versorgt wird. Diese Trays können in einem automatischen Hochregallagersystem, welches besser unter dem englischem Namen Automated Storage and Retrieval System (AS/RS) bekannt ist, (zwischen-)gelagert werden. Ein der Kommissionierung vorgelagerter Nachschubprozess stellt sicher, dass die benötigten Artikelmengen im AS/RS verfügbar sind. Sobald die Artikel in den Trays sind, werden diese zur Traykommissionierung transportiert und dort kommissioniert.

Während der Kommissionierung packt ein Mitarbeiter die Artikel in die vorgesehenen Rollcontainer und anschließend werden diese zum Warenausgang transportiert. Für den Transport der Rollcontainer vom Kommissionier- zum Warenausgangsbereich (siehe graue Pfeile in Abbildung 2) ist ein fahrerloses Transportfahrzeug (FTF) (englisch: Automated Guided Vehicle, AGV) zuständig.

Ein Rollcontainer gehört immer zu einer Einzelhandelsfiliale und somit auch zu einer Tour. Es spielt keine Rolle, ob Aufträge in der Paletten- oder Traykommissionierung verarbeitet werden. Es können auch Teile eines Auftrags (das heißt, gewisse Auftragszeilen) im PKS und andere Auftragszeilen im TKS verarbeitet werden. Im Warenausgang werden die Rollcontainer einer Einzelhandelsfiliale, sowie die Einzelhandelsfilialen einer Tour zusammengefasst.

2.2. Das hinterlegte Datenmodell

Da einerseits alle für diese Arbeit relevanten Informationen und anderseits der Großteil an automatisierten Optimierungsschritten in einer Datenbank gespeichert sind, ist es dem Verständnis förderlich, das hinterlegte Datenmodell zu beschreiben. Dabei handelt es sich jedoch nicht um die Datenbank des Lagersystems. Das folgende Datenmodell beinhaltet nur für die Optimierung erforderliche Informationen. In Abbildung 3 ist das vereinfachtes Entity-Relationship-Modell (ERM) nach *Chen*-Notation zu sehen. Die Entitäten (blaue Rechtecke), Beziehungen (rote Rauten) und Attribute (weiße Ellipsen) sind in der Datenbank durch die Tabellen und Spalten abgebildet. Eine Entität steht für ein Objekt, eine Person oder eine Gegenstand wie z.B. Auftrag oder Filiale im *Fallbeispiel*. Die Attribute beschreiben die Eigenschaften einer Entität und dienen zur Identifikation (unterstrichene Attribute). Die Beziehungen verbinden die Entitäten untereinander.

Ein Tag ist für die Kommissionierung in eine oder mehrere Wellen (Kommissionierwellen) unterteilt. Diese Wellen werden sequentiell abgearbeitet und besitzen eine zeitlich definierte Dauer. Bei Schichtbetrieb ist nicht der Arbeitstag sondern jede Schicht

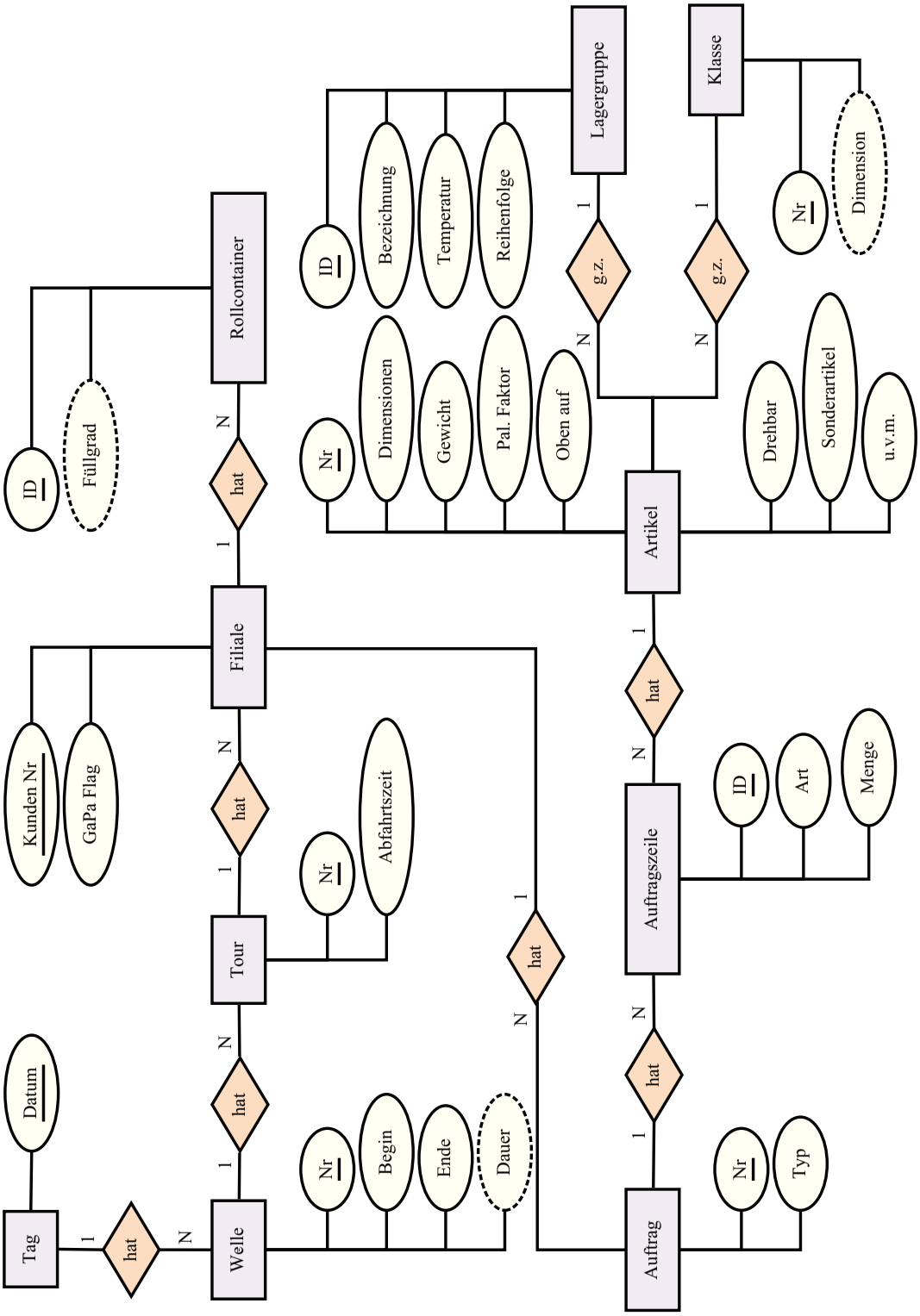


Abbildung 3: ER-Modell

in Kommissionierwellen unterteilt. Jede Welle beinhaltet Touren deren Abfahrtszeiten nach dem Ende der Welle liegen. Die Einteilung des Tages in Wellen und die Zuteilung von Filialen zu Touren ist nicht Inhalt dieser Arbeit. Diese Optimierung wird zeitlich vor der Auftragsverteilung vom Betreiber des Distributionszentrums durchgeführt. Die daraus resultierende Wellen- und Toureneinteilung wird für die Optimierung als gegeben angenommen. In der Wellen- und Toureneinteilung liegt sehr wahrscheinlich ein weiteres Optimierungspotenzial. Diese Einteilung bestimmt, wie erfolgreich die darauffolgende Auftragsverteilung durchgeführt werden kann. Werden in der Wellen- und Toureneinteilung grundlegende Fehler gemacht, kann dies selbst eine optimale Auftragsverteilung nicht mehr ausgleichen. Werden hingegen bereits bei der Wellen- und Toureneinteilung die Ziele der Auftragsverteilung verfolgt, so kann diese davon enorm profitieren. Ein einheitliches Vorgehen und eine gesamte Optimierung über die Wellen-, Toureneinteilung und Auftragsverteilung sollte in Betracht gezogen werden. Die Auftragsverteilung ist im Lösungsraum durch die vielen bereits vorab festgelegten Rahmenbedingungen stark eingeschränkt. Die Möglichkeit auf eine bessere Gesamtlösung spricht für eine durchgängige Optimierung, die damit verbundene, steigende Komplexität dagegen.

Jede Tour hat eine bestimmte Zahl an Filialen. Diese Filialen (manchmal auch „Kunden“ genannt) gehören immer nur zu einer Tour und benötigen verschiedene Artikel in bestimmten Mengen. Die Artikel werden nach (bzw. während) der Kommissionierung auf Rollcontainer gepackt. Diese Rollcontainer werden den Filialen zugestellt und müssen einen möglichst hohen Füllgrad aufweisen.

Jede Filiale bestellt mit Hilfe von Aufträgen die benötigten Waren. Eine Filiale hat genau einen Auftrag pro Betrachtungszeitraum (Tag, Schicht, Welle) und jeder Auftrag kann immer genau einer Filiale zugewiesen werden.

Im *Fallbeispiel* haben die Filialen ein relativ homogenes Bestellverhalten und unterscheiden sich hauptsächlich aufgrund ihrer Auftragsvolumina. Die Filialen werden aufgrund ihrer Bestellungen klassifiziert und in kleine, mittlere und große Filialen eingeteilt. Das Bestellverhalten der Filialen ist durch saisonale Schwankungen geprägt. Im Sommer werden beispielsweise viel mehr Getränke als im Winter bestellt.

Jeder Auftrag besteht aus einer oder mehreren Auftragszeilen. Jede Auftragszeile beinhaltet die Bestellmenge eines Artikels. Jeder Artikel wird durch seine Eigenschaften genauer beschrieben. Artikel werden zu Lagergruppen zusammengefasst. Lagergruppen bestimmen die Kommissionierreihenfolge des Artikels und den Bereich in dem der Artikel gelagert wird. Zusätzlich werden Artikel nach anderen Maßstäben klassifiziert (wie z.B. nach Gewicht oder Abmessungen). Diese Klassifizierung wird auch in die Kommissionierreihenfolge einbezogen.

Die Attribute von Tag, Welle, Tour, Rollcontainer, Auftrag, Auftragszeile und deren Beziehungen ändern sich je Betrachtungszeitraum (Tag, Schicht, Welle). Das heißt, für einen neuen Tag gehen andere Aufträge mit neuen Auftragszeilen und unterschiedlichen Artikelmengen als am Vortag von den Filialen ein. Es werden die Wellen und Touren für diesen Tag neu eingeteilt. Für die Kommissionierung werden neue Rollcontainer benötigt, die dann an die Filialen versandt werden.

Die Attribute von Filiale, Artikel, Lagergruppe, Klasse und deren Beziehungen sind Stammdaten. Sie ändern sich nicht täglich. Die Abmessungen eines Artikel oder die

Adresse einer Filiale bleiben über einen längeren Zeitraum gleich. Durch die Stammdatenpflege werden notwendige Änderungen durchgeführt. Sobald beispielsweise ein neuer Artikel in das Sortiment des Lagers aufgenommen wird, müssen die benötigen Informationen beschafft und eingepflegt werden.

Das ERM ist für das *Fallbeispiel* innerhalb einer Datenbank verwirklicht. Die Kerntabelle dieser Datenbank beinhaltet alle Auftragszeilen der Einzelhandelsfilialen. Sie hat mit Abstand die meisten Einträge und ist für die Optimierung von größter Bedeutung, da die Optimierung die Auftragszeilen auf die beiden Kommissioniersysteme verteilt. Sie kann mit allen anderen Tabellen der Datenbank sinnvoll verknüpft werden.[1] Das vollständige ERM des *Fallbeispiels* und die resultierende Datenbank mit ihren Tabellen ist weitaus komplexer und umfangreicher. Abbildung 3 soll lediglich einen Einblick in das Datenmodell geben, ohne unwesentliche Details zu zeigen.

Das Datenmodell beinhaltet keine Bestandsdaten. Damit sind beispielsweise Bestände im HRL gemeint. Diese Informationen wären für den Optimierungsschritt von Bedeutung, aber sind zum Zeitpunkt der Ausführung nicht verfügbar. Es wird für die Optimierung davon ausgegangen, dass alle benötigten Bestände zur Auftragserfüllung spätestens zum Zeitpunkt der Kommissionierung im Lager verfügbar sind.

Um realistische Ergebnisse zu erhalten, sind reale Auftragsdaten eines bereits bestehenden, vergleichbaren RDC des Lebensmitteleinzelhandels als Testdaten in Verwendung. Diese Daten wurden bereits vor dieser Arbeit gesammelt, geprüft, verarbeitet und aufbereitet. Die Daten wurden ergänzt, um den zukünftigen Gegebenheiten des neuen Lagers zu entsprechen. Diese Tätigkeiten sind nicht Teil dieser Arbeit und werden an dieser Stelle nicht weiter beschrieben. Bei der Beschreibung der Herangehensweise wird noch einmal darauf eingegangen und die Beschaffung dieser Daten angesprochen. Diese Daten werden im weiteren Verlauf der Arbeit als *Testdaten* bezeichnet.

2.3. Ausgangssituation

Die Realisierung des Lagersystems ist vom Systemanbieter in Form eines Projektes organisiert. Diese Projekte starten mit dem Verkauf einer Lösung und enden mit andauernder Betriebsunterstützung. Die Dauer solcher Projekte erstreckt sich meist über mehrere Jahre. Das mit dieser Arbeit in Zusammenhang stehende Projekt befindet sich in der Planungsphase. Es ist bereits eine Lösung vom Systemanbieter an den Kunden, in diesem Fall einen Konzern der Lebensmitteleinzelhandelsbranche, verkauft. Somit sind die zu erreichenden Vorgaben bereits fixiert. Wie diese Ziele nun innerhalb des Lagersystems im Detail erreicht werden, steht dem Systemanbieter weitestgehend offen. Hier setzt diese Arbeit an, um einen bereits angedachten Ablauf innerhalb des Systems weiter zu verbessern.

[1] Anmerkung: Die direkte Verbindung von Auftragszeile zu Rollcontainer ist in Abbildung 3 nicht dargestellt. Diese Beziehung ist etwas anspruchsvoller aufzulösen, da eine Auftragszeile durchaus auf mehrere Rollcontainer verteilt werden kann. Dies geschieht, wenn eine Teilmenge der Auftragszeile auf einen neuen Rollcontainer kommissioniert werden muss, da der alte Rollcontainer bereits seine maximale Kapazität erreicht hat.

Die Ausgangssituation für die Optimierung ist eine bereits teilweise automatisierte Auftragsverteilung der Planungsabteilung des Systemanbieters. Diese basiert auf den selben *Testdaten* wie die Optimierung. Der Ansatz der Planungsabteilung verfolgt keine schriftlich klar definierten Ziele. Die Eigenschaften der Komponenten des Lagersystems und die Datenbasis bestimmen einen heuristischen Algorithmus. Dieser besteht aus mehreren Schritten, versucht unterschiedliche Ziele zu erreichen und hält einige Nebenbedingungen ein. Es sind verschiedene Office-Software-Produkte kombiniert in Verwendung, um eine Lösung zu erhalten. Es sind auch einige nicht automatisierte Schritte notwendig, welche die Bedienung für den Benutzer erschweren und die Fehleranfälligkeit erhöhen. Eine nicht automatisierte Version ist für die Planung brauchbar, aber muss für den endgültigen Betrieb im Lagersystem erst automatisiert werden.

Ein konkurrierender Lösungsansatz ist seitens der technischen Projektleitung für die Problemlösung besser geeignet. Dieser verfolgte ein etwas anderes Vorgehen und ist nur als Idee im Unternehmen des Systemanbieters verfügbar.

Beide Varianten befassen sich hauptsächlich mit der Verteilung der Aufträge zwischen den beiden Kommissioniersystemen. Sie entscheiden welche Auftragszeile in welchem Bereich des Lagers kommissioniert werden. Die Verteilung auf die Kommissionierstationen erfolgt bei Bedarf (beispielsweise für die Simulation) durch ein einfaches Rundlaufverfahren. Es sind im Unternehmen des Systemanbieters viele Ideen für weitere Verbesserungsmöglichkeiten vorhanden. Diese können auf Grund von Zeitmangel, zu stark ansteigender Komplexität, fehlender Kompatibilität zum bisherigen Vorgehen oder der verwendeten Software nicht oder nur schwer umgesetzt werden.

Ausgehend von diesem Sachverhalt soll eine Optimierung nach wissenschaftlichem Vorgehen erfolgen. Es werden Ziele, Nebenbedingungen und die Vorgehensweise niedergeschrieben. Die Problemstellung (bzw. Problemstellungen) soll analysiert, abgegrenzt und beschrieben werden. Ein neuer durchgängiger Lösungsweg soll entwickelt werden. Ein automatisierter Ablauf soll die Möglichkeit bieten, die realisierten Lösungsansätze auf die (Test-)Daten anzuwenden und untereinander zu vergleichen.

2.4. Herangehensweise

Der erste Schritt ist das Beschaffen von Informationen. Ohne qualitativ hochwertige *Testdaten* ist beispielsweise das Überprüfen und Vergleichen der Lösungsansätze nicht möglich. Wird auf Grund von falschen Informationen entschieden, so entspricht das Ergebnis der Optimierung nicht den gewünschten Zielvorgaben.

Das Aufbereiten der *Testdaten* stellt einen erheblichen Aufwand dar. Zuerst müssen die Daten in einen konsistenten Zustand gebracht werden. Die Beschaffung von zusätzlich benötigten Informationen ist mit zeitlichen Verzögerungen verbunden. Ein Großteil dieser Arbeit ist bereits von der Planungsabteilung im Voraus erledigt worden. Es ist schwer, den Ursprung der *Testdaten* zurück zu verfolgen, da keine schriftliche Dokumentation verfügbar ist. Das in Abschnitt 2.2: „Das hinterlegte Datenmodell“ vorgestellte ERM ist in Anlehnung an die *Testdaten* entworfen.

Der Großteil an relevanten Informationen für die Optimierung wurde vor dieser Arbeit mündlich kommuniziert. Beispielsweise wurden Ziele und Nebenbedingungen verbal

festgelegt ohne diese schriftlich festzuhalten. Vorliegende Zielkonflikte sind dann nicht transparent und Vorgaben unscharf formuliert. Da mehrere Personen am Lösungsfindungsprozess beteiligt sind, stehen einige widersprüchliche Aussagen ungeklärt im Raum.

Um ein wissenschaftliches Arbeiten zu ermöglichen, müssen diese Informationen gesammelt, geprüft, in Einklang gebracht und festgehalten werden. Die *Testdaten* werden analysiert, ein neues Datenbankmodell entworfen und zusätzliche Daten angefordert. Mit den beteiligten Personen werden klärende Gespräche geführt, welche Ziele, Nebenbedingungen, Einschränkungen und Lösungswege definieren sowie Unklarheiten bereinigen.

Dieser Informationsbeschaffungsprozess und das Einfügen der Testdaten in das neue Datenbankmodell dominieren die vorbereitenden Schritte zur Optimierung des *Fallbeispiels*. Immer wieder müssen neues Wissen und unerwartete Änderungen integriert werden. Dies ist unter anderem darauf zurückzuführen, dass einige Vorgaben während der Planungsphase nicht fixiert werden können.

In der Folge werden die Probleme identifiziert und nach passenden Lösungswegen gesucht. Dabei wird mit der Abgrenzung und Formulierung der Problemstellung/en begonnen. Eine Literaturrecherche liefert in der Regel mehrere erfolgversprechende Ansatzpunkte für die Umsetzung. Bereits vorhandene Ansätze sollen die Suche nach neuen Lösungsalternativen nicht einschränken. Außerdem wird eine Zielfunktion entworfen und diskutiert. Mit ihrer Hilfe erfolgt die weitere Optimierung und die bisherigen Ansätze sind untereinander und mit den neuen Ansätzen vergleichbar.

Drittens folgt die Umsetzung und Automatisierung der Lösungswege. In diesem Schritt steckt viel programmiertechnischer Aufwand. Alle identifizierten Lösungsansätze werden entwickelt, bei Bedarf erweitert und automatisiert. Entstehen zu viele verschiedene Alternativen, sodass nicht alle umgesetzt werden können, erfolgt eine sinnvolle Einschränkung.

Abschließend werden die einzelnen Lösungswege an den Testdaten analysiert. Die Lösungen werden an Hand der Zielfunktion und anderer Kennzahlen bewertet. Die Rechenzeit darf eine definierte Grenze nicht überschreiten, da für die Optimierung im echten Betrieb des *Fallbeispieles* nur eine begrenzte Zeit zur Verfügung steht. Eine Simulation des Lagersystems ermöglicht es, die Optimierung der Auftragsverteilung so zu testen, als ob die Aufträge in einem realen Lager verarbeitet würden. Dies erlaubt das Ziehen weitere Rückschlüsse, welche sonst nur im echten Betrieb des Lagersystems möglich sind. Alle aus dieser Herangehensweise resultierenden Erkenntnisse und Ergebnisse werden in dieser Arbeit zusammengefasst und genauer beschrieben.

2.5. Testdaten des Fallbeispiels

Die Testdaten stammen von einem vergleichbaren Distributionszentrum des Lebensmittelhandels. Sie umfassen die Auftragsdaten von zwei aufeinander folgenden Wochen. Die Anzahl an Filialen, die Zusammensetzung der Aufträge, die Kommissioniermengen und das Artikelspektrum entsprechen den erwarteten Gegebenheiten des neuen Lagers. Die Testdaten wurden von der Planungsabteilung des Systemanbieters für das Projekt aufbereitet. Sie beinhalten neben den zwei Wochen an echten Auftragsdaten zwei künstlich erzeugte Tage. Diese spiegeln einen Durchschnittstag und einen Spitzentag der Zukunft

wider. Sie wurden durch Vervielfältigung von Auftragszeilen, Aufträgen und Filialen erzeugt.

Die Testdaten umfassen eine Artikelspektrum von circa 5700 verschiedenen Artikeln. In einem Lagersystem werden die geführten Artikel üblicherweise als Stock Keeping Unit (SKU) bezeichnet. Die rund 800000 Auftragszeilen stammen von mehr als 130 Filialen. Es werden in den 14 Tagen circa 1,4 Millionen GVEs an Waren kommissioniert. GVE steht für Großverbrauchereinheit oder Großhandels-Verkaufs-Einheit (GVE) und ist die kleinste Einheit in die eine Palette von Artikeln im Distributionszentrum aufgespalten wird. Zum Beispiel werden Mineralwasserflaschen immer nur in Packungen zu je 6 Stück bestellt, kommissioniert und geliefert. Es ist nicht möglich, eine einzelne Mineralwasserflasche zu bestellen. In den Filialen werden die Waren dann auch in kleineren Verbundeinheiten verkauft. Ein anderes Beispiel für eine GVE ist eine Lage von 12 Joghurts in einem gelochten Karton.

Das Distributionszentrum ist in zwei völlig separate Bereiche, den gekühlter Bereich (GKB) und den ungekühlter Bereich (UKB), unterteilt. Im gekühlten Bereich werden schnell verderbliche und temperatursensible Artikel wie u.a. Wurst, Käse, Milch, Obst und Gemüse gelagert und kommissioniert. Im nicht gekühlten Bereich befinden sind alle anderen Waren. Das Artikelspektrum umfasst, von Getränken bis Hygiene-Artikel, alle Arten von Waren, die in einer typischen Einzelhandelsfiliale der Branche erhältlich sind. Im Layout des Distributionszentrums sind die Bereiche gespiegelt angeordnet. Jeder Bereich verfügt über alle notwendigen Teilsysteme. Die Bereiche unterscheiden sich in der Auslegung und Kapazität der einzelnen Teilsysteme. Zum Beispiel haben die beiden Bereiche unterschiedlich viele PKS- und TKS-Stationen zur Verfügung. Das Hochregallager ist im GKB beispielsweise unterteilt in 4 und 12 °C Lagergassen mit einfach tiefen[2] Lagerplätzen, wobei im UKB nur nicht temperierte, doppelt tiefe Lagerplätze benötigt werden.

Jeder Tag außer dem Spitzentag ist in 4 Kommissionierwellen unterteilt. Der Spitzentag besitzt einmal im Bereich GKB 8 Kommissionierwellen. Somit stehen für die Auswertungen der Verfahren insgesamt 56 bzw. 60 Kommissionierwellen je Bereich zur Verfügung. Das heißt, die Optimierung der Auftragsverteilung kann an über Hundert verschiedenen Kommissionierwellen validiert werden.

[2]Anmerkung: In einem einfach tiefen Lagerplatz kann nur eine Palette gelagert werden. In einem doppelt tiefen Lagerplatz können bis zu zwei Paletten hintereinander gelagert werden.

3. Optimierung und deren Ziele

Folgende zwei Fragen sollen gestellt werden, bevor eine Optimierungsaufgabe durchgeführt wird:

> Warum besteht Bedarf zur Optimierung?
>
> Was soll durch die Optimierung erreicht werden?

Lediglich der Wunsch nach Verbesserung rechtfertigt den dabei entstehenden Aufwand nicht. Für das *Fallbeispiel* werden die beiden Fragen wie folgt beantwortet:

> Eine Optimierung ist notwendig, da mögliche Potentiale nicht ausgeschöpft werden. Die nicht optimierte (bisherige) Vorgehensweise erfüllt nicht alle Vorgaben.
>
> Erreicht werden soll eine gleichmäßige Verteilung der Arbeitslast, ein wirtschaftlicher Betrieb der Anlage und eine Entlastung von Flaschenhals-Ressourcen.

Dieses Optimierungsprojekt unterscheidet sich von einer klassischen Optimierungsaufgabe mit eindeutig messbarem Ist- und Soll-Zustand. Die Ausgangssituation ist nicht real greifbar, sondern als geplantes Vorgehen gegeben. Zusätzlich ist dieses Vorgehen nur als undokumentierte Idee verfügbar und bisherige Umsetzungsversuche sind für die weitere Optimierung unbrauchbar.

Die Analyse der gesamten Auftragsverteilung zeigt, dass der geplante Verteilungsablauf in mehrere, sinnvoll trennbare Teile zerlegt werden kann (siehe dazu Abschnitt 3.1: „Ablauf der Auftragsverteilung“). Diese Teilaufgaben werden sequentiell abgearbeitet und reduzieren die Komplexität der Auftragsverteilung. Es werden zwei weitestgehend voneinander unabhängige Problemstellungen (Auftragszeilenverteilung und Teilsystemoptimierung) formuliert und gelöst, wobei das Ergebnis der ersten als Eingabe der zweiten dient. Auch das Lösen des gesamten Problems in einem Schritt sowie die Rückführung von Ergebnissen plus Wiederholung wurden in Betracht gezogen. Aufgrund der zu hohen Komplexität der Problemstellung und der damit verbundenen Rechenzeit ist dieses Vorgehen für das *Fallbeispiel* nicht geeignet.

3.1. Ablauf der Auftragsverteilung

Der Auftragsverteilung ist die Welleneinteilung vorgelagert. Bei der Welleneinteilung werden die Touren nach Abfahrtszeiten gruppiert und den entstehenden Kommissionierwellen zugeteilt. Die daraus resultierenden Auftragsdaten dienen als Eingabe für die Auftragsverteilung. Der Ablauf der Auftragsverteilung wird grundsätzlich in mehrere sequentiell abzuarbeitende Teilaufgaben zerlegt. Diese sind in Abbildung 4 grafisch dargestellt. Sobald alle zu kommissionierenden Aufträge für einen Tag bzw. eine Welle vom übergeordneten System übermittelt sind, kann mit der Auftragsverteilung begonnen werden. Die Auftragsverteilung wird in zwei übergeordnete Teilaufgaben (wobei jede der beiden eine der Problemstellungen enthält) zerlegt. Benötigte Artikel, die sich zur Zeit der Auftragsverteilung noch nicht im Lager befinden, werden als „voraussichtlicher

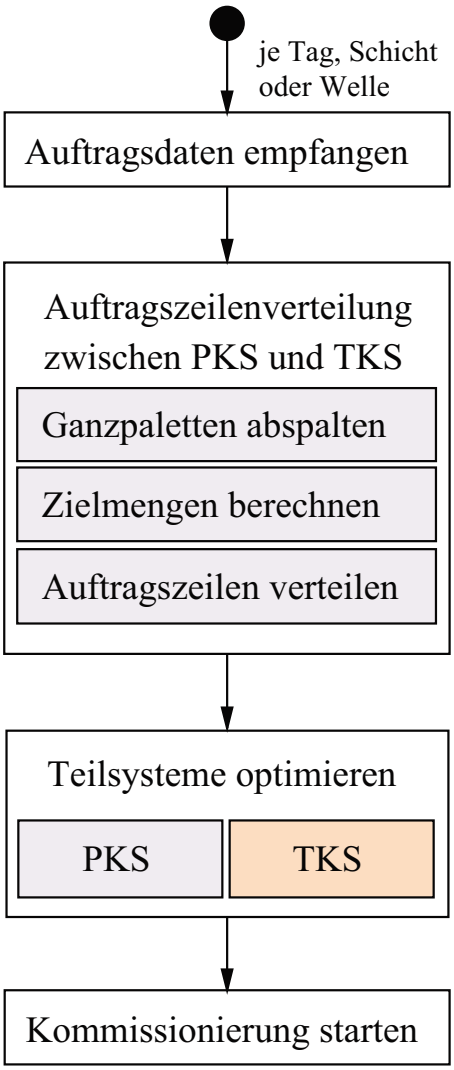


Abbildung 4: Geplanter Ablauf

Bestand“ angenommen. Das heißt, es wird erwartet, dass diese Artikel noch bis zum Zeitpunkt der Kommissionierung im Lager eintreffen.

Die Auftragszeilenverteilung befasst sich mit der Aufteilung der Aufträge zwischen den Kommissioniersystemen und wird wiederum in drei separate Teile aufgegliedert. Die Ganzpaletten-Abspaltung separiert Teile von Aufträgen, die ohne Kommissionierung direkt zum Warenausgang transportiert werden. Die Zielmengenberechnung liefert mögliche Betriebsarten des Lagers und Zielvorgaben für die einzelnen Kommissionierbereiche. Die Auftragszeilenverteilung erledigt die eigentliche Aufgabe der Verteilung zwischen PKS und TKS und ist der Kern dieser ersten Teilaufgabe.

Die Teilsystemoptimierung betrachtet die beiden Kommissioniersysteme des *Fallbeispiels* getrennt voneinander und weist den verfügbaren Arbeitsstationen Filialen (bzw. deren Auftragszeilen) zu. Die Optimierung erfolgt hier parallel. In dieser Arbeit wird nur die Teilsystemoptimierung des PKS behandelt, da es sich dabei um eine Neuentwicklung mit wenigen Erfahrungswerten handelt und deshalb mit großen Optimierungspotentialen zu rechnen ist. Die Teilsystemoptimierung des TKS ist ein bereits in sich optimierter und abgeschlossener Teil und kann nicht verändert werden.

Nach diesen Schritten startet der Nachschubprozess und die Kommissionierung. Der Nachschubprozess versorgt das AS/RS mit den benötigten Artikeln damit diese später auf den TKS-Stationen kommissioniert werden können. Der Nachschub eines Artikels wird automatisch ausgelöst, sobald eine Mindestbestand im AS/RS unterschritten wird oder bekannt ist, dass für die nächste Kommissionierwelle nicht genügend Stück (gemessen in GVE) dieses Artikel im AS/RS eingelagert sind. Für die Stationen des PKS ist kein Nachschubprozess nötig, da diese direkt aus dem HRL oder Wareneingang versorgt werden.

3.2. Auftragszeilenverteilung

Die Aufteilung der Auftragszeilen zu PKS und TKS erfolgt im ersten Schritt anhand der Minimierung der gesamten Kommissionierbelegschaft und aktiven Kommissionierstationen. Dies geschieht über alle Kommissioniersysteme und -stationen hinweg. Vorgegeben ist die maximale Kommissionierdauer der Welle, welche nach Möglichkeit maximal ausgeschöpft werden soll. Mittels einfacher Berechnung wird die beste Kombination aus aktiven PKS- und TKS-Stationen, welche die benötigte Kommissioniermenge der Welle in der definierten Dauer abarbeiten kann, bestimmt. Diese Berechnung bedient sich durchschnittlicher Leistungswerte der Anlange, welche regelmäßig auf Richtigkeit und Aktualität geprüft werden müssen. Optional erfolgt die Auswahl der aktiven Stationen manuell.

Stehen die aktiven PKS- und TKS-Stationen fest, werden die Aufträge auf die beiden Bereiche verteilt. Diese Aufteilung hat das Ziel, eine zeitlich gleichmäßige Auslastung der verfügbaren Systeme und Stationen zu erreichen, sowie die Anzahl der unterschiedlichen Artikel im PKS zu minimieren. Neben den bereits erwähnten Vorgaben von verfügbarer Belegschaft und aktiven Stationen sowie der zeitlichen Begrenzung der Welle, werden die spezifischen Einschränkungen der zu kommissionierenden Artikel und die Leistungseigenschaften beteiligter Systeme (z.B. Ein- und Auslagerleistung des HRL, maximale

Anzahl an Palettenwechsel pro Stunde im PKS, verfügbare Schächte pro PKS-Station usw.) beachtet. Es entstehen durchaus Situationen, in denen keine zulässige Lösung gefunden wird. In diesem Fall werden die Eingabedaten oder andere Parameter geprüft und verändert.

Das HRL findet an dieser Stelle besondere Erwähnung, da es während der Planungsphase als Flaschenhals-Ressource identifiziert wurde. Das HRL kann nur bis zu einer rechnerischen Maximalleistungsgrenze betrieben werden. Die Berechnung erfolgt über eine Spielzeitberechnung der Lagerspiele. Diese Grenze wird im *Fallbeispiel* durch die maximalen Ein- und Auslagerungen von Paletten pro Stunde beschrieben und hängt unter anderem von der Anzahl an Lagergassen und der Betriebsart des HRL ab. Zusätzlich muss die vom HRL hin- und wegführende Fördertechnik über genügend Leistung und Kapazität (der Pufferstrecken) verfügen. Sobald Grenzen überschritten werden, sinkt die Leistung des gesamten Systems. Da das HRL und die Fördertechnik von mehreren Lagerprozessen genutzt werden, wird diese maximale Leistung nicht direkt als Bedingung herangezogen. Die finalen Leistungswerte werden im echten Betrieb überprüft und bei Bedarf angepasst. Die Auftragszeilenverteilung kann die tatsächlich auszulagernden Paletten nicht im Voraus bestimmen, da keine Bestandsdaten zum Berechnungszeitpunkt vorliegen. Die notwendigen Ein- und Auslagerungen im HRL schätzt die Auftragszeilenverteilung näherungsweise.

Jeder Artikel, der sich im HRL befindet und kommissioniert werde soll, muss dieses auch verlassen, egal wo der dazugehörige Auftrag bearbeitet wird. Werden jedoch Artikel im TKS kommissioniert, so werden diese aus dem AS/RS ausgelagert. Jedoch wird dieses wiederum vom HRL über den Nachschubprozess mit Artikeln aufgefüllt. Somit können Auslagerungen aus dem HRL zeitlich verschoben werden, wenn die Kommissionierung der entsprechenden Artikel von einem Kommissioniersystem in das andere verschoben wird. Die Auftragsverteilung versucht, die Ressource HRL in Bezug auf ihre zulässige Leistung zu schonen und warnt vorzeitig bei kritischen Werten.

3.3. Teilsystemoptimierung

Dieser Abschnitt behandelt ausschließlich die weitere Optimierung des PKS. Auf die weitere Optimierung des TKS (wie beispielsweise Packbildoptimierung) wird hier nicht weiter eingegangen, da diese nicht im Umfang dieser Arbeit enthalten ist.

Das Palettenkommissioniersystem wird in sich weiter optimiert. Dabei handelt es sich hauptsächlich um die Zuteilung der Filialen zu den einzelnen PKS-Stationen. Die Palettenkommissionierung besteht in der Regel aus mehreren Stationen. An jeder Station arbeitet ein Bediener, der die gewünschten Artikel von zwei Quellpaletten in bis zu zwölf verschiedene Zielschächte, welche Rollcontainer aufnehmen, kommissioniert. Pro Welle soll jedem Schacht immer nur eine Filiale zugewiesen werden. Jeder Station kann somit nur eine bestimmte Anzahl an Filialen zugewiesen werden. Diese Stationen müssen eine vorgegebene Nennleistung erreichen und möglichst gleichmäßig ausgelastet werden. Ob die vorgegebene Nennleistung erreicht wird, hängt unter anderem von der Erfahrung des Bedieners und der Struktur der zu kommissionierenden Auftragszeilen ab. Nur Letzteres kann von der Auftragsverteilung beeinflusst werden. Es werden mengenmäßig viele

GVE von möglichst wenig verschiedenen Artikeln an den PKS-Stationen kommissioniert, da einerseits jeder Palettenwechsel (die Quellpaletten werden durch neue ersetzt) Zeit kostet und die Ein-/Auslagerrate des HRL begrenzt ist. Palettenwechsel zwischen den PKS-Stationen sind auch zu vermeiden. Dabei ist anzumerken, dass Paletten immer „artikelrein“ sind (das heißt, auf einer Palette befinden sich immer nur ein oder mehrere GVE von einem Artikel einer Charge eines MHD). Werden Paletten während des Kommissionierprozesses komplett abgeräumt, so werden diese nicht erneut in das HRL eingelagert, sonder verlassen den Kreislauf als Leerpaletten. Zwischen HRL und PKS existieren begrenzte Pufferstrecken. Sie sollen ermöglichen, dass der Warenfluss zur Kommissionierung nicht unterbricht.

Von der Planungsabteilung wurde durch Simulation der Anlage folgendes Verhalten erkannt: Wenn zu viele verschiedene Artikel im PKS kommissioniert werden, kann das HRL nicht genügend schnell die benötigten Quell-Paletten auslagern. Aus dem Layout und der Fördertechnik des *Fallbeispiels* kann eine maximale Leistungsgrenze für den An- und Abtransport von Paletten zum PKS (z.B. 80 Paletten pro Stunde) errechnet werden. Diese Leistung ist von der jeweils schwächsten Komponente im System abhängig. Zusätzlich verringern vermehrte Palettenwechsel die maximal mögliche Leistung einer PKS-Station. Für die Optimierung im PKS wurde festgelegt, dass die maximale Anzahl an verschiedenen Artikeln pro Stunde (SKU/h) innerhalb einer Welle als Kennzahl und Stellgröße fungiert. Wird der festgelegte Wert überschritten, so ist mit Leistungseinbusen zu rechnen.

Die Ergebnisse diverser Simulationsläufe sind die Basis für das Diagramm in Abbildung 5. Es zeigt den Verlauf der Leistung des PKS in GVE pro Stunde bei ansteigender SKU-Anzahl. Durch die einzelnen aus der Simulation stammenden Datenpunkte ist ein Trendlinie (nach polynomischer Regression 2.Ordnung) gelegt worden. Dieses Diagramm festigt die Annahmen der Planungsabteilung, dass die PKS Leistung ab einer bestimmten SKU-Anzahl stark abfällt. Um das Leistungsverhalten der PKS zu bestätigen, sollten weitere Simulationsläufe oder praktische Tests mit Leistungsmessungen durchgeführt werden.

Unter der Annahme, dass das Einhalten einer Obergrenze an verschiedenen Artikeln im PKS und an deren Stationen einen wirtschaftlich effizienten Betrieb ermöglicht, wird die Teilsystemoptimierung ausgelegt. Bei der Optimierung des PKS werden diese Ziele verfolgt und Bedingungen eingehalten:

- Primäre Ziele
  - Minimierung der Anzahl verschiedener Artikel pro Station
  - Gleichmäßige Verteilung der Arbeitslast auf die (aktiven) Stationen
  - Minimierung der Wegzeiten und -strecken bei der Schachtzuordnung [3]

[3]Das heißt, die Filialen einer Tour sollen auf den Schächten innerhalb einer PKS-Station räumlich möglichst nahe beieinander platziert werden. Dies verringert die Wegzeiten und -strecken beim Abholen der Rollcontainer von den Schächten.

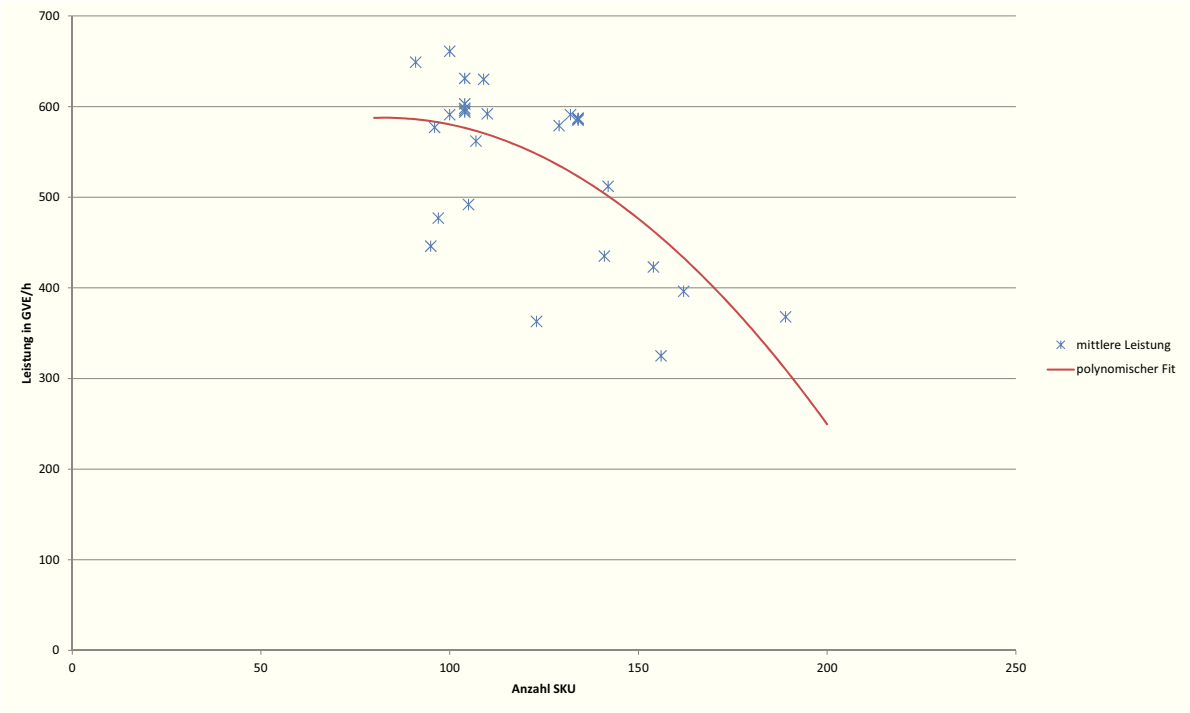


Abbildung 5: Diagramm - Leistungsverlauf des PKS

- Sekundäre Ziele
  - Minimierung der Palettenwechsel innerhalb einer und zwischen den Stationen
  - Maximierung der Leerpaletten
  - Maximierung des Füllgrads der Rollcontainer
- Nebenbedingungen
  - Anzahl an Filialen pro Station ist begrenzt
  - Geforderte Nennleistung muss erreicht werden
  - Maximale Leistung des HRL beachten
  - Andere Einschränkungen beachten (wie z.B. eine maximale Anzahl an erlaubten Palettenwechseln oder verschiedener Artikel)
- Optional
  - Aufteilung in Normal- und Aktionsaufträge (diese müssen dann auf separate Rollcontainer kommissioniert werden)

Zu den sekundären Zielen ist anzumerken, dass die ersten beiden indirekt vom primären Ziel der Minimierung der Anzahl verschiedener Artikel beeinflusst werden. Der Füllgrad der Rollcontainer wird geschätzt, da keine Packbildberechnung für den Bereich PKS vorgesehen ist und somit der tatsächlich erreicht Füllgrad sehr stark vom Können des Bedieners der Station abhängig ist. Die Artikel sind für die Kommissionierung in Warengruppen (z.B. Getränke oder Obst) eingeteilt. Diese Gruppen sind sequentiell abzuarbeiten und bestimmen die Kommissionierreihenfolge. Innerhalb der Warengruppen werden weitere Gruppen definiert, mit dem Ziel ein besseres Packbild zu erhalten. Diese sind beispielsweise nach Gewicht oder Verpackung eingeteilt. Die Waren einer Untergruppe werden bei der Kommissionierung zusammengehalten. Um die Fördertechnik effizient zu betreiben, müssen in jeder Untergruppe genügend viele Artikel kommissioniert werden. Die Reihenfolge der Artikel innerhalb dieser Untergruppen ist im Vorhinein nicht bestimmbar. Für die Kommissionierung im PKS ist auf Grund der Fördertechnik keine explizite Kommissioniersequenz auf Artikelebene garantiert, was für eine Packbildberechnung essentiell ist.

3.4. Ziel und Zielfunktion

Erst ein Ziel ermöglicht eine sinnvolle Optimierung, da es die Richtung vorgibt. Bereits die Definition eines Zieles kann im betrieblichen Umfeld zu unerwarteten Diskussionen führen. Erheblich ausschweifender werden diese Diskussionen, wenn es mehrere konkurrierende Ziele gibt. Eine Priorisierung der Ziele löst diesen Konflikt. In der Fachliteratur sind Ansätze zur Bewertung und Gewichtung von Zielen zu finden.

Um eine Lösung des Problems an Hand der Zielerreichung zu bewerten, ist die Definition einer Zielfunktion unumgänglich. Diese Funktion beinhaltet alle Ziele in gewichteter Form und liefert den Zielwert der Lösung. Dieser Schritt ist unter Umständen recht aufwändig, wenn die Werte der Ziele in unterschiedlichen Dimensionen oder Einheiten gemessen werden und diese untereinander nicht zusammengefasst werden können. In der Praxis können oft Ziele (und Nebenbedingungen) nicht mathematisch „scharf“ formuliert werden. Diese Unschärfen können mit Hilfe des Konzepts der *Fuzzy-Mengen* in eine mathematisch stimmiges Model übergeführt werden.

Der Ansatz des *Fuzzyfizierens* wurde in einem frühen Stadium der Arbeit in Erwägung gezogen, da viele unscharfe Nebenbedingungen vor der detaillierten Analyse des Gesamtproblems erwartet wurden. Diskussionen und Gespräche machten eine scharfe Formulierung der Ziele und Nebenbedingungen möglich. Somit wurde der Einsatz von *Fuzzyfizerung* obsolet. Außerdem gefiel dem Systemanbieter die Verwendung von unscharfen Zielen und Bedingungen nicht besonders gut. Für den praktischen Einsatz ist es leichter nachvollziehbar, wenn ein konkreter Schwellwert anpasst werden kann, um die Einhaltung einer geforderten Bedingung zu gewährleisten.

4. Auftragszeilenverteilung zwischen TKS und PKS

Zuerst werden die Daten aufbereitet und die Ganzpaletten von der Auftragsmenge abgespalten. Diese abgespaltenen GVE-Stückmengen sind für die weiteren Betrachtungen nicht mehr von Bedeutung, da Ganzpaletten nicht über die Kommissioniersysteme abgearbeitet werden. Danach werden die Zielmengen der einzelnen Kommissionierstationen rechnerisch ermittelt. Dabei ist zu berücksichtigen, dass hierfür die definierten Nennleistungen der Stationen herangezogen werden. Der folgenden Schritt weist alle Auftragszeilen unter Berücksichtigung der errechneten Zielmengen einem der beiden Kommissioniersysteme zu.

4.1. Problemstellung

Es soll erreicht werden, dass alle zu kommissionierenden Auftragszeilen möglichst effizient auf die beiden Kommissioniersysteme verteilt werden. Dabei wird die verfügbare Kommissionierdauer möglichst vollständig ausgeschöpft, ohne diese zu überschreiten. Die Anzahl der aktiven Stationen (entspricht den Mitarbeitern) wird minimiert. „Effizient“ bedeutet in diesem Kontext, dass zuerst die PKS Stationen mit genügend, strukturell passenden, Aufträgen versorgt werden. Alle anderen Auftragszeilen werden im TKS bearbeitet. Für die Durchführung dieser Aufteilung steht eine begrenzte Zeitdauer zur Verfügung. Erst wenn alle Aufträge vom übergeordneten System übermittelt worden sind, kann mit der Auftragszeilenverteilung begonnen werden. Damit das System mit der Kommissionierung bis zur ersten Tourenabfahrtszeit fertig ist, muss spätestens zu einem berechenbaren Zeitpunkt davor mit der Kommissionierung und dem, der Kommissionierung im TKS vorgelagerten, Nachschub begonnen werden. Somit ist nur der Zeitraum zwischen dem letzten übermittelten Auftrag der Kommissionierwelle und dem spätest möglichen Beginn des Nachschubs und der Kommissionierung für die Auftragszeilenverteilung und deren Berechnungen verfügbar.

4.2. Datenaufbereitung

Dieser vorbereitende Arbeitsschritt dient dazu, die Daten zu prüfen und bei Bedarf Modifikationen durchzuführen. Es werden beispielsweise zusätzliche Informationen aus den bestehenden Daten für die spätere Verarbeitung generiert. Der für das *Fallbeispiel* wichtigste Schritt ist die Abspaltung der Ganzpaletten, welcher nun ausführlich beschrieben wird. Unnötige Kommissionierung wird durch ihn vermieden, Flaschenhalskomponenten des System entlastet und die gesamte Leistungsfähigkeit des Lagersystems erhöht.

Es werden alle Auftragszeilen, deren Filialen Ganzpaletten akzeptieren und deren Auftragszeilenmenge größer oder gleich dem Palettenfaktor des betreffenden Artikels ist, ermittelt. Der Palettenfaktor gibt die maximale GVE-Stückanzahl eines Artikels auf einer Palette an. Außerdem muss der Artikel einige weitere Eigenschaften aufweisen. Zum Beispiel darf der Artikel kein Sonder- oder Gewichtsartikel sein.

Es soll im realen Betrieb die Möglichkeit bestehen, Anbruchpaletten wie Ganzpaletten zu behandeln, um die Kommissioniersysteme weiter zu entlasten. Jedoch werden Ganz-

paletten den Anbruchpaletten vorgezogen. Anbruchpaletten sind Paletten im HRL, von denen bereits zum Teil Artikel entnommen wurden. Beispielsweise wurden von 10 möglichen Lagen einer Palette bereits 2 Lagen depalettiert und ins AS/RS nachgeschoben. Diese Anbruchpalette hat somit im Vergleich zu einer Ganzpalette einen Füllgrad von 80 Prozent. Der Palettenfaktor wird mit einem Faktor kleiner Eins multipliziert, um einen minimal erlaubten Füllgrad für Anbruchpaletten zu definieren. Anbruchpaletten über diesem Füllgrad dürfen abgespalten werden. Das heißt, ein Faktor von 0,9 bedeutet, dass Anbruchpaletten ab 90 Prozent Füllgrad akzeptiert werden.

Alle möglichen Auftragszeilen werden von den restlichen abgespalten und die Ganz- bzw. Anbruchpaletten im Lagerbestand reserviert. Überbleibende Restmengen der abgespaltenen Auftragszeilen werden weiterhin über die Kommissioniersysteme behandelt.

4.3. Zielmengen berechnen

Durch die Zielmengenberechnung wird bestimmt, wie viel an jeder aktiven Station kommissioniert werden soll. Die Berechnung der Zielmengen bedient sich folgender Gleichung:

$$k = (p\ r_{PKS} + t\ r_{TKS})z$$

$$\begin{aligned} \text{mit } p &\ldots \text{Anzahl aktiver PKS-Stationen} \\ t &\ldots \text{Anzahl aktiver TKS-Stationen} \\ z &\ldots \text{Kommissionierzeit in h} \\ k &\ldots \text{Kommissioniermenge in GVE} \end{aligned} \tag{1}$$

$$\begin{aligned} \text{wobei } r_{PKS} &= \text{Nennleistung PKS-Station in GVE/h} \\ r_{TKS} &= \text{Nennleistung TKS-Station in GVE/h} \end{aligned}$$

GVE (Großverbrauchereinheit, siehe dazu Abschnitt 2.5) ist die Maßeinheit in der die Kommissionierleistung im *Fallbeispiel* gemessen wird. In dieser Arbeit werden Kommissioniermengen, Auftragsmengen, Bestellmengen und Zielmengen in GVE angegeben und gemessen.

Ist die Anzahl der aktiven p PKS- und t TKS-Stationen sowie die benötigte Kommissioniermenge k der Welle vorgegeben, wird die gesamt benötigte Kommissionierzeit z wie folgt ausgedrückt und berechnet:

$$z(p,t,k) = \frac{k}{(p\ r_{PKS} + t\ r_{TKS})} \tag{2}$$

Beispiel Zielmenge: Ein Lagersystem mit 3 PKS-Stationen und 5 TKS-Stationen möchte eine Kommissioniermenge von 19.600 GVE in einer Kommissionierwelle erledigen. Die PKS-Stationen haben eine Nennleistung von r_{PKS}:= 550 GVE/h und die

TKS-Stationen eine Nennleistung von r_{TKS}:= 650 GVE/h. Wie lange benötigt das Lagersystem für die Kommissionierung dieser Welle wenn alle verfügbare Stationen aktiv sind?

$$z = \frac{19600\ [\text{GVE}]}{3 \cdot 550\ [\text{GVE/h}] + 5 \cdot 650\ [\text{GVE/h}]} = 4\ [\text{h}]$$

Die benötigte Kommissionierzeit beträgt 4 Stunden.

Wird die Kommissionierzeit z mit der Nennleistung r_{PKS} einer PKS-Station oder r_{TKS} einer TKS-Station multipliziert, so wird die Zielmenge der jeweiligen Station bestimmt. Um die gesamte Ziel-Kommissioniermengen k_{PKS} und k_{TKS} der Kommissioniersysteme PKS und TKS zu erhalten, muss dieser Wert noch mit der Anzahl an aktiven Stationen je Kommissioniersystem multipliziert werden.

Beispiel Zielmenge (Fortsetzung): Wie hoch sind die Ziel-Kommissioniermengen der Kommissioniersysteme?

$$k_{PKS} = z\ r_{PKS}\ p = 4\ [\text{h}] \cdot 550\ [\text{GVE/h}] \cdot 3 = 6600\ [\text{GVE}]$$
$$k_{TKS} = z\ r_{TKS}\ t = 4\ [\text{h}] \cdot 650\ [\text{GVE/h}] \cdot 5 = 13000\ [\text{GVE}]$$

Das heißt, im PKS sollen 6.600 GVEs und im TKS 13.000 GVEs in dieser Welle kommissioniert werden.

Die definierten Nennleistungen der Stationen können sich im Betrieb verändern und müssen daher bei Bedarf angepasst werden. Nach der Inbetriebnahme des Lagersystems können sicherlich nicht die endgültig möglichen Kommissionierleistungen an den Stationen erreicht werden, sondern erst nach einer Anlaufphase. Die Lernkurve der eingesetzten Mitarbeiter spielt dabei eine wesentliche Rolle. Leistungsschwankungen können jedoch immer auftreten.

Um die Zielmengenberechnung automatisiert durchzuführen, werden alle möglichen Kombinationen von aktiven PKS- und TKS-Stationen herangezogen. Dabei steht p_{max} für die maximal mögliche Anzahl an aktiven PKS-Stationen und t_{max} für die maximal mögliche Anzahl an aktiven TKS-Stationen. Diese Werte sind durch die physikalisch verfügbaren Kommissionierstationen je Kommissioniersystem und Lagerbereich sowie durch die verfügbaren, dafür qualifizierten Mitarbeiter bestimmt. Gleichung 1 ändert sich wie folgt:

$$
\begin{aligned}
k &= (p\ r_{PKS} + t\ r_{TKS}) z_{p,t} \qquad \forall (p,t) \in (P,T) \\
&\text{mit } z \ldots \text{Kommissionierzeit in h} \\
&\quad\ k \ldots \text{Kommissioniermenge in GVE} \\
&\text{wobei } P := \{0,1,\ldots,p_{max}\} \qquad p_{max} \in \mathbb{N} \\
&\quad\ T := \{0,1,\ldots,t_{max}\} \qquad t_{max} \in \mathbb{N} \\
&\quad\ r_{PKS} = \text{Nennleistung PKS-Station in GVE/h} \\
&\quad\ r_{TKS} = \text{Nennleistung TKS-Station in GVE/h}
\end{aligned}
\tag{3}
$$

Für alle daraus resultierenden Szenarien (das heißt, PKS-TKS-Kombinationen) können die weiteren Schritte (wie Auftragszeilenverteilung) durchgeführt werden. Um den Rechenaufwand einzuschränken, werden die Szenarien auf eine kleine Anzahl reduziert. Ein gültiges Szenario muss die geforderte Kommissioniermenge während der Wellendauer abarbeiten können. Außerdem müssen genügend PKS-fähige Artikel in der Kommissionierwelle des Szenarios verfügbar sein, um die geforderte Kommissioniermenge im PKS Bereich erreichen zu können. Ist dies nicht der Fall, so scheidet das Szenario aus. Für den Bereich TKS ist diese Prüfung nicht erforderlich, da alle Artikel[4] über das TKS kommissioniert werden können.

Zusätzlich darf das Szenario eine definierte Obergrenze an SKU pro Stunde (SKU/h) im PKS nicht übersteigen. Dieser Wert ist im *Fallbeispiel* eine gute Näherung für die Kennzahl Paletten pro Stunde (Pal/h), mit welcher die Nebenbedingung zum Einhalten der maximalen Leistung des HRL definiert ist. Das Einhalten dieser Bedingung kann erst nach der Auftragszeilenverteilung geprüft werden, da erst dann die Anzahl verschiedener SKU im PKS für das Szenario feststeht.

Ob die gesamte Kommissionierzeit minimiert oder die Belegschaft bzw. die Anzahl aktiver Stationen bei maximal möglicher Kommissionierzeit minimiert werden soll, muss an dieser Stelle entschieden werden. Die Kombinationen werden nach dem festgelegtem Kriterium gereiht. Im *Fallbeispiel* erfolgt die Reihung nach Übereinstimmung von Kommissionierdauer zu Wellendauer. Auch eine manuelle Auswahl ist möglich.

Das Problem kann unter Anlehnung an Gleichung 3 und Gleichung 2 wie folgt formuliert werden:

$$
\text{Minimiere} \qquad \Delta z = w - z(p,t,k) \tag{4}
$$

$$
\text{Nebenbedingung} \qquad w \geq z \tag{5}
$$

Dabei steht w für die Wellendauer in Stunden und Δz für die Differenz von Wellendauer zu benötigter Kommissionierzeit. Die Kommissioniermenge k bleibt konstant und die Variablen p und t sind ganzzahlige Wertepaare der Form $(p,t) \in (P,T)$. Die Lösung des Minimierungsproblems liefert die beste gültige Lösung mit minimaler Zeitdifferenz.

[4] Anmerkung: Ausgenommen sind übergroße und -schwere Artikel, welche immer über die Sonderkommissionierung bearbeitet werden müssen und daher nicht über die TK.

In der Praxis wird ein Tag bzw. eine Schicht in mehrere Wellen unterteilt. Da die Mitarbeiteranzahl während einer Schicht konstant bleibt, soll auch die Gesamtanzahl an $p+t$ aktiven Stationen während einer Schicht, das heißt, für alle i Wellen dieser Schicht, konstant bleiben. Die Mitarbeiter sind in der Lage, beide Arten von Kommissionierstationen zu bedienen und können daher zwischen den Wellen einer Schicht ihren Arbeitsplatz wechseln. Eine zusätzliche Bedingung der Form $m = p+t$ kann dies gewährleisten, wenn eine Anzahl von m Mitarbeitern angenommen wird. Das Minimierungsproblem ändert sich für die i-te Welle mit m Mitarbeitern wie folgt.

$$\text{Minimiere} \qquad \Delta z_i = w_i - z_i(p_i, t_i, k_i) \tag{6}$$

$$\text{Nebenbedingung} \qquad w_i \geq z_i \tag{7}$$

$$m = p_i + t_i \tag{8}$$

Soll die minimale Summe der Zeitdifferenzen der Wellen eines Tages, bei gleichbleibender Mitarbeiteranzahl während des Tages, bestimmt werden, dann muss das bisherige mathematische Modell erweitert werden. Es sollen alle von den Stationen her möglichen Mitarbeiterzahlen, die für jede Welle des Tages mindestens eine gültige Lösung liefern, miteinbezogen werden. Bei einer ungültigen Lösung ist die Kommissionierzeit z größer als die Wellendauer w. Ein übergeordnetes Minimierungsproblem entsteht, welche das bisherige Modell als Teilproblem enthält. Die Menge aller möglichen Anzahlen von Mitarbeiter, ohne Prüfung ob eine gültige Lösung des Teilproblems existiert, kann als $M := \{1, 2, \cdots, p_{max} + t_{max}\}$ definiert werden.

Sind die zu kommissionierenden Mengen unter den Wellen einer Schicht ungleichmäßig verteilt (gemessen an der Wellendauer), dann sind unter Umständen keine zufriedenstellenden Lösungen möglich. Ein Arbeitstag bzw. eine Schicht besteht beispielsweise aus vier Kommissionierwellen. Die ersten drei Wellen werden am besten mit fünf Mitarbeitern abgearbeitet. Die vierte Welle benötigt mindestens acht Mitarbeitern, um eine gültige Lösung zu erhalten. Da die Mitarbeiteranzahl über den Tag konstant bleiben muss, muss auch in den ersten drei Wellen mit acht Mitarbeitern gearbeitet werden. Die Zeitdifferenz von Wellendauer zu Kommissionierzeit wird für diese drei Wellen größer als notwendig. Es wäre besser, die Arbeitslast ausgeglichener auf die Wellen zu verteilen. An dieser Stelle wird ersichtlich, warum die dem Problem vorgelagerte Wellen- und Toureneinteilung erheblichen Einfluss auf die Qualität der Lösung hat.

4.4. Varianten der Auftragszeilenverteilung

Nachdem die Ziel-Kommissioniermengen für PKS und TKS berechnet sind, werden die Auftragszeilen der Filialen dementsprechend auf die Kommissioniersysteme verteilt. Der Betrachtungszeitraum für die folgenden Verfahren ist immer eine Kommissionierwelle eines Tages.

In der frühen Planungsphase des Lagersystems wurden zwei konkurrierende Ansätze zur Verteilung der Auftragszeilen von verschiedenen Abteilungen des Systemanbieters

zur Diskussion gestellt. Es wurde von unterschiedlichen Rahmenbedingungen ausgegangen und somit konnte nicht objektiv bewertet werden, welcher der beiden Ansätze nun besser sei. Durch diesen Umstand ist es notwendig, eine einheitliche Vergleichsbasis zu schaffen, um eine objektive Entscheidung treffen zu können. Beide Ansätze zielen darauf ab, dem PKS eine Menge an Auftragszeilen mit möglichst wenig verschiedenen Artikeln zuzuweisen, sodass die angestrebte Ziel-Kommissioniermenge k_{PKS} des PKS erreicht wird. Sie dienten als Grundlage zur Auslegung des Lagers und deren Kapazitäten. Die Grundideen der beiden Ansätze ist in den ersten beiden Varianten der Auftragsverteilung umgesetzt. Es wurde eine weitere Varianten entwickelt, welche die Vorteile beider Ansätze vereint.

4.4.1. Ziel

Das Ziel legt die Vergleichsbasis fest und grenzt optionale Nebenbedingungen und -ziele klar ab. Die Auftragszeilenverteilung zielt darauf ab, die Anzahl der SKU für das PKS zu minimieren. Je weniger verschiedene SKU auf dem PKS kommissioniert werden, desto besser. Wenn die durchschnittliche SKU-Anzahl pro Stunde im PKS einen definierten Wert sph_{max} übersteigt, wird eine Warnung ausgegeben. Als zweite Zielgröße für die Bewertung der Algorithmen wird deren benötigte Rechenzeit herangezogen, da nur eine begrenzte Zeitdauer für die Auftragszeilenverteilung zur Verfügung steht. Die Berechnung einer Kommissionierwelle dauert im Durchschnitt nur wenige Sekunden. Es sollen jedoch im *Fallbeispiel* mehrere Konstellationen von PKS- und TKS-Stationen (Szenarien genannt) mit anschließender Teilsystemoptimierung durchgerechnet werden und daher steht für eine Verteilung der Auftragszeilen zwischen PKS und TKS nur mehr sehr wenig Zeit zur Verfügung.

Ob Artikel als Aktions- und Normalartikel getrennt werden, kann als optionale Nebenbedingung angegeben werden. Die Markierung, ob ein Artikel als Aktions- oder Normalartikel bestellt wird, ist in der Auftragsstruktur gespeichert. Es kann somit eine Filiale ein und den selben Artikel auf beide Arten bestellen. Ist die Option gewählt, so werden die Aktions- und Normalartikel separat in verschiedene Rollcontainer kommissioniert. Das heißt, eine Filiale mit beiden Artikelarten, wird schlussendlich wie zwei (fiktive) Filialen behandelt. Die eine Filiale belegt zwei Schächte im PKS.

Eine weitere Option ist die nachträgliche Verbesserung des Rollcontainerfüllgrads. Sie zielt darauf ab, dass keine halbleeren Rollcontainer im PKS entstehen. Die Berechnung erfolgt ausschließlich aufgrund ermittelter Durchschnittswerte und den Volumina der Rollcontainer und Artikel.

4.4.2. Die stärksten Filialen und deren Artikel (kurz: sFuA)

Zuerst werden die mengenmäßig stärksten Filialen identifiziert. Das sind jene Filialen, welche mengenmäßig die meisten GVE an Artikeln durch ihren Auftrag bestellen. Beschränkt wird die Anzahl an Filialen durch die maximal verfügbare Anzahl s_{max} an Schächten. Diese lässt sich aus den aktiven PKS-Stationen und den Schächten pro PKS-Station berechnen ($s_{max} := p \cdot s$). Danach werden die bestellten Artikel dieser iden-

tifizierten Filialen nach Menge gereiht. Es werden so viele Artikel ausgewählt, bis die Bestellmenge der Auftragszeilen der Filialen dieser Artikel in Summe größer oder gleich der Ziel-Kommissioniermenge k_{PKS} für das PKS ist. Diese Auftragszeilen werden in Folge dem PKS zugeteilt. Der Rest der Auftragszeilen wird im TKS kommissioniert. Sollten die stärksten Filialen nicht genügend Auftragszeilen haben, um die Zielmenge zu erreichen, obwohl bereits alle Artikel ausgewählt sind, dann wird eine Warnung ausgegeben.[5] Die verwendeten Symbole sind in Tabelle 1 mit Beschreibung zusammengefasst.

Tabelle 1: Symbole Auftragszeilenverteilung

Symbol	**Beschreibung**
$\mathcal{A}$	Menge aller Artikel
$\mathcal{A}_{PKS}$	Menge aller Artikel die im PKS kommissioniert werden, wobei gilt $\mathcal{A}_{PKS} \subset \mathcal{A}$
$\mathcal{A}_x$	Menge ausgewählter Artikel, wobei gilt $\mathcal{A}_x \subset \mathcal{A}$
j	ein Artikel
$\mathcal{F}$	Menge aller Filialen
$\mathcal{F}_{PKS}$	Menge aller Filialen die im PKS kommissioniert werden, wobei gilt $\mathcal{F}_{PKS} \subset \mathcal{F}$
$\mathcal{F}_x$	Menge ausgewählter Filialen, wobei gilt $\mathcal{F}_x \subset \mathcal{F}$
i	eine Filiale
$\mathcal{Z}$	Menge aller Auftragszeilen
$\mathcal{Z}_{i,j}$	Menge aller Auftragszeilen der Filiale i mit dem Artikel j
l	eine Auftragszeile
k	Kommissioniermenge in GVE
k_{PKS}	Ziel-Kommissioniermenge im PKS in GVE
s_{max}	Maximale Anzahl an Schächten im PKS
sph_{max}	Maximale Anzahl an SKU pro Stunde
z	Kommissionierzeit in h

Die Prozedur SumMenge in Algorithmus 1 wird in allen Varianten benutzt und daher an dieser Stelle vorgestellt. Sie gibt die kumulierte Kommissioniermenge k einer Menge von Artikeln $\mathcal{A}_x$ zurück. Es werden dafür die Auftragszeilen aller Aufträge einer Menge bestimmter Filialen $\mathcal{F}_x$ herangezogen. Die Berechnung erfolgt über die in der Datenbank abgebildeten Auftragsstruktur. Das heißt, in der Datenbank können die Aufträge einer Filiale bestimmt werden. Die Aufträge beinhalten die Auftragszeilen mit Information über den bestellten Artikel und die Bestellmenge. Werden die Bestellmengen aller Auftragszeilen einer Filiale aufsummiert, so wird die gesamte Bestellmenge dieser Filiale

[5] Anmerkung: Dieser Fall sollte jedoch praktisch nie eintreten, da schon bei der Zielmengenverteilung eine entsprechende Prüfung durchgeführt wird. Es wird die maximal mögliche Kommissioniermenge in GVE für den Bereich PKS berechnet. Das heißt, es werden die Auftragsmengen der mengenmäßig stärksten Filialen, deren Anzahl durch die maximal mögliche Anzahl von Schächten begrenzt ist, aufsummiert. Szenarien, deren berechnete Zielmenge größer dieser aufsummierten Menge ist, sind ungültig.

berechnet. Diese Menge an GVE muss im Distributionszentrum kommissioniert werden, um die Aufträge bzw. die Bestellung der Filiale zu erfüllen.

Algorithmus 1 Summe der Bestellmenge von bestimmten Artikeln und Aufträgen

1: **procedure** SUMMENGE($\mathcal{A}_x$, $\mathcal{F}_x$)
2: Initialisiere Kommissioniermenge $k \leftarrow 0$ ▷ in GVE
3: **for all** Filiale $i \in \mathcal{F}_x$ **do**
4: **for all** Artikel $j \in \mathcal{A}_x$ **do**
5: $\mathcal{Z}_{i,j} \leftarrow$ Ermittle alle Auftragszeilen der Filiale i mit dem Artikel j
6: $k \leftarrow k + \sum_{l \in \mathcal{Z}_{i,j}}$ Bestellmengen der Auftragszeile l
7: **end for**
8: **end for**
9: **return** k
10: **end procedure**

Soll beispielsweise die gesamt benötige Kommissioniermenge für eine Filiale $i \in \mathcal{F}$ berechnet werden, so würde der Aufruf SUMMENGE($\mathcal{A}, \{i\}$) das gewünschte Ergebnis liefern. Ist gefragt, wie oft ein Artikel $j \in \mathcal{A}$ insgesamt von allen Filialen bestellt wurde, so kann dies mit SUMMENGE($\{j\}, \mathcal{F}$) berechnet werden. Diese einfachen Berechnungen werden unter anderem für die Sortierung innerhalb von Listen in den späteren Algorithmen verwendet, um die Filialen oder Artikel nach ihren aufsummierten Bestellmengen bzw. Kommissioniermengen zu reihen.

Algorithmus 2 zeigt den im *Fallbeispiel* umgesetzten Ablauf, welcher als Aufzählung wie folgt dargestellt werden kann.

1. Sortiere alle Filialen (mengenmäßig stärkste zuerst).
2. Wähle die mengenmäßig stärksten Filialen aus (begrenzt durch Schächte).
3. Sortiere all jene Artikel, die in den Aufträgen dieser Filialen vorkommen (mengenmäßig stärkste zuerst).
4. Wähle so viele unterschiedliche Artikel aus, bis die Zielmenge erreicht ist oder alle Artikel gewählt sind.
5. Kann die Zielmenge nicht mit allen Artikeln der stärksten Filialen erreicht werden, dann gib eine Warnung aus und wähle alle Artikel.
6. Teile die Filialen und Auftragszeilen dieser Artikel dem PKS zu.

Beim ursprünglichen Ansatz, der die Grundidee für diese Varianten liefert, wurde bei den Artikeln immer zwischen Normal- und Aktionsaufträgen unterschieden. Es wurde versucht, die Normal- und Aktionsartikeln in zwei zeitlich getrennten Wellen innerhalb einer Kommissionierwelle zu kommissionieren, um die Trennung auf den Rollcontainern zu erreichen. Ein Vorteil ist, dass dadurch die Anzahl an verfügbaren Schächten

Algorithmus 2 Die stärksten Filialen und deren Artikel

Require: $\mathcal{A}$, $\mathcal{F}$, $\mathcal{Z}$, s_{max}, k_{PKS}

Initialisiere Menge $\mathcal{F}_{PKS} \leftarrow leereMenge$
Initialisiere Menge $\mathcal{A}_{PKS} \leftarrow leereMenge$
$FilialListe \leftarrow$ Sortiere $\mathcal{F}$ nach der Bestellmenge aller Auftragszeilen $l \in \mathcal{Z}$
$\mathcal{F}_{PKS} \leftarrow$ Wähle die $x \leq s_{max}$ ersten Filialen aus der $FilialListe$
$\mathcal{A}_x \leftarrow$ alle Artikel $j | j \in \mathcal{A}$ und j in einer Auftragszeile $l \in \mathcal{Z}$ der Filialen $i \in \mathcal{F}_{PKS}$
$ArtikelListe \leftarrow$ Sortiere $\mathcal{A}_x$ nach der Bestellmenge aller Auftragszeilen $l \in \mathcal{Z}$ der Filialen $i \in \mathcal{F}_{PKS}$
Initialisiere Kommissioniermenge k
repeat
    $\mathcal{A}_{PKS} \overset{\text{add}}{\leftarrow}$ Hole ersten Artikel aus $ArtikelListe$
    $k \leftarrow$ SumMenge$(\mathcal{A}_{PKS}, \mathcal{F}_{PKS})$
until $k \geq k_{PKS}$ **or** $ArtikelListe$ **is** $leer$
if $k < k_{PKS}$ **then**
    Warnung: Zielmenge kann nicht erreicht werden!
end if
if $\frac{|\mathcal{A}_{PKS}|}{z} > sph_{max}$ **then**
    Warnung: Zu viele SKU bzgl. HRL-Leistung gewählt!
end if
Weise alle Auftragszeilen $l \in \mathcal{Z}$ der Filialen $i \in \mathcal{F}_{PKS}$ mit dem Artikel $j \in \mathcal{A}_{PKS}$ dem PKS zu

pro Kommissionierwelle verdopplt wird, und dadurch doppelt so viele Filialen im PKS verarbeitet werden können. Dies machte den Vergleich mit anderen Varianten, die ohne diese doppelte Schachtbelegung pro Kommissionierwelle arbeiten, fast unmöglich. Dieses Vorgehen ist in seiner Umsetzung im Kontext des *Fallbeispiels* problematisch, da innerhalb einer Kommissionierwelle die Reihenfolge der Lagergruppen von der Steuerung der Fördertechnik eingehalten wird, und diese von größeren GVE-Stückmengen innerhalb einer Lagergruppe pro Welle profitiert. Somit ist das weitere Zerlegen von Kommisssionierwellen in kleinere Welle nicht optimal. Zusätzlich muss ein Artikel der sowohl als Aktions- als auch Normalartikel innerhalb einer Kommissionierwelle bestellt wird, beim Kommissionieren mindestens zweimal ausgelagert und zur Kommissionierstation transportiert werden. Werden sowohl Aktions- als auch Normalartikel innerhalb einer Welle kommissioniert, so können Synergien genutzt werden, und es muss im besten Fall nur einmal ausgelagert werden. Daher werden Aktions- und Normalartikel in der hier vorgestellten Version, falls die Nebenbedingung „Trennung Aktion-Normal“ gewünscht ist, in der selben Welle jedoch auf unterschiedlichen Schächten kommissioniert. Dies erfolgt durch eine zuvor durchgeführte Aufteilung einer Filiale in zwei fiktive Filialen, wobei die eine nur Aktionsartikel und die andere nur Normalartikel in ihrem Auftrag beinhaltet. Somit ist eine Trennung auf den Rollcontainern garantiert. Das gleiche Vorgehen ist bei allen anderen Varianten möglich und somit kann in Folge ein Vergleich stattfinden.

4.4.3. Die stärksten Artikel und deren Filialen (kurz: sAuF)

Bei dieser Variante wird iterativ immer ein neuer Artikel $i \in \mathcal{A}$ zu einer Menge von ausgewählten Artikeln $\mathcal{A}_{PKS}$ hinzugefügt. Hierfür werden alle Artikel zuvor nach ihren Bestellmengen absteigend sortiert, um eine gute Reihenfolge festzulegen. Danach wird die beste Kombination von maximal s_{max} Filialen für die ausgewählten Artikel bestimmt, sodass die gesamte Kommissioniermenge maximal wird. Dies wird wiederholt, bis so viele Auftragszeilen $l \in \mathcal{Z}$ ermittelt worden sind, dass die Kommissioniermenge dieser Auftragszeilen die notwendige Ziel-Kommissioniermenge k_{PKS} des PKS übersteigt. Kann die angestrebte Ziel-Kommissioniermenge nicht erreicht werden, so wird eine Warnung ausgegeben und mit der maximal möglichen Kommissioniermenge fortgefahren. Filialen können im Gegensatz zu Artikeln zu Beginn des Verfahren in der Menge $\mathcal{F}_{PKS}$ ausgewählter Filialen vorkommen und im späteren Verlauf wieder daraus entfernt werden.

Beispiel Auftragszeilenverteilung Es stehen zwei Filialen mit Auftragszeilen für insgesamt vier unterschiedliche Artikel zur Wahl. Details können aus Tabelle 2 entnommen werden. Das Lagersystem besitzt nur eine PKS-Station mit einem verfügbaren Schacht. Es kann also nur eine Filiale im PKS kommissioniert werden. Eine Ziel-Kommissioniermenge wird für dieses Beispiel nicht benötigt, da nur gezeigt werden soll, warum eine Filiale wieder aus der Menge aus identifizierten Filialen entfernt werden kann. Wie würde diese Variante der Auftragsverteilung vorgehen?

Tabelle 2: Beispiel Auftragszeilenverteilung Daten

Filiale/Artikel	**A1**	**A2**	**A3**	**A4**
F1	100	40	35	0
F2	90	70	5	20

Eine sortierte Liste der Artikel nach ihren Kommissioniermengen würde wie folgt aussehen: $ArtikelListe := \{(A1, 190); (A2, 110); (A3, 40); (A4, 20)\}$. In der ersten Iteration würde Artikel A1 zur Menge ausgewählter Artikel für das PKS hinzugefügt werden. Die Wahl der Filiale würde auf F1 fallen, da $100 > 90$ ist. In der zweiten Iteration würde Artikel A2 zur Menge ausgewählter Artikel hinzugefügt werden. Die Wahl der „besten“ Filiale fällt nun auf F2, da $100+40 < 90+70$ ist. Filiale F1 würde (falls der Ablauf jetzt abbrechen würde) nicht mehr im PKS kommissioniert werden. Die nächsten Iterationen würde im weiteren Verlauf dieses einfachen Beispiels, den Artikel A3 zur Menge hinzufügen und zuerst F1 (da $175 > 165$) auswählen und schlussendlich Artikel A4 zur Menge hinzufügen und wieder F2 (da $175 < 185$) auswählen.

Die Beschreibung der Vorgehensweise ist folgende:

1. Lege zwei leere Mengen für identifizierte Artikel und Filialen an.
2. Sortiere alle Artikel (mengenmäßig stärkste zuerst).
3. Solange die berechnete Kommissioniermenge kleiner als die Ziel-Kommissioniermenge ist und noch nicht alle Artikel in der Menge sind, wiederhole folgendes:

a) Wähle den ersten Artikel der Sortierung, welcher noch nicht in der Menge ist, aus.

b) Füge diesen Artikel zur Menge der identifizierten Artikel hinzu.

c) Aktualisiere die Menge der Filialen (begrenzt durch Schächte) so, dass die besten Filialen zu den identifizierten Artikeln ausgewählt sind (das heißt, die Kommissioniermenge der identifizierten Artikel wird maximiert).

d) Berechne die Kommissioniermenge der bisher identifizierten Artikel und deren bester Filialen.

4. Kann die Zielmenge nicht erreicht werden, dann gib eine Warnung aus.

5. Teile die Auftragszeilen der Filialen und identifizierten Artikel dem PKS zu.

Algorithmus 3 zeigt den im Fallbeispiel umgesetzten, iterativen Ablauf.

Algorithmus 3 Die stärksten Artikel und deren Filialen

Require: $\mathcal{A}$, $\mathcal{F}$, $\mathcal{Z}$, s_{max}, k_{PKS}

Initialisiere Menge $\mathcal{F}_{PKS} \leftarrow leereMenge$
Initialisiere Menge $\mathcal{A}_{PKS} \leftarrow leereMenge$
$ArtikelListe \leftarrow$ Sortiere $\mathcal{A}$ nach der Bestellmenge aller Auftragszeilen $l \in \mathcal{Z}$
Initialisiere Kommissioniermenge k
repeat
    $\mathcal{A}_{PKS} \overset{\text{add}}{\leftarrow}$ Hole ersten Artikel aus $ArtikelListe$
    $FilialListe \leftarrow$ Sortiere $\mathcal{F}$ nach der Bestellmenge der gewählten Artikel
        ▷ d.h., nach SUMMENGE$(\mathcal{A}_{PKS}, \{i\})$ $\quad \forall i \in \mathcal{F}$
    $\mathcal{F}_{PKS} \leftarrow$ Wähle die $x \leq s_{max}$ ersten Filialen aus der $FilialListe$
    $k \leftarrow$ SUMMENGE$(\mathcal{A}_{PKS}, \mathcal{F}_{PKS})$
until $k \geq k_{PKS}$ **or** $ArtikelListe$ **is** $leer$
if $k < k_{PKS}$ **then**
    WARNUNG: Zielmenge kann nicht erreicht werden!
end if
if $\frac{|\mathcal{A}_{PKS}|}{z} > sph_{max}$ **then**
    WARNUNG: Zu viele SKU bzgl. HRL-Leistung gewählt!
end if
Weise alle Auftragszeilen $l \in \mathcal{Z}$ der Filialen $i \in \mathcal{F}_{PKS}$ mit dem Artikel $j \in \mathcal{A}_{PKS}$ dem PKS zu

Beim ursprünglichen Ansatz dieser Variante wurde grundsätzlich auf die Unterscheidung zwischen Normal- und Aktionsartikel verzichtet. Dadurch steht im Gegensatz zum ersten Ansatz immer ein größerer Vorrat an Auftragszeilen pro Filiale für die Optimierung zur Verfügung. Somit können in der Regel immer kleinere SKU-Kennzahlenwerte, das heißt, weniger verschiedene Artikel müssen im PKS kommissioniert werden, für den PKS-Bereich erreicht werden.

4.4.4. Die stärksten Artikel und deren Filialen (2-stufig, kurz: Phase)

Diese Variante ist eine Weiterentwicklung der vorigen mit dem Ziel, die Laufzeit zu verbessern. Die obige Variante muss nach jedem Hinzufügen eines neuen Artikels die Suche nach den besten Filialen erneut durchführen. Dieser in Summe sehr zeitintensive Schritt wird in dieser erweiterten Variante zu Beginn nach Möglichkeit vermieden.

Dazu wird dem Ablauf eine Initialisierungs-Phase vorgelagert. In dieser Phase wird die Menge von identifizierten Artikeln mit den mengenmäßig stärksten Artikeln befüllt, bis deren kumulierte Menge die zu erreichende Kommissioniermenge übersteigt (ohne dabei auf die begrenzte Filialenanzahl zu achten). Danach werden die Filialen identifiziert und die tatsächlich zugewiesene Kommissioniermenge berechnet. Im besten Fall sind dann keine oder nur noch wenige Iterationen nach dem alten Schema nötig. Dies hängt jedoch stark von der Auftragsstruktur und den Eingabedaten ab. An den *Testdaten* des *Fallbeispiels* wurden gute Ergebnisse erzielt.

Es folgt der modifizierte Ablauf als Aufzählung:

1. Lege zwei leere Mengen für identifizierte Artikel und Filialen an.
2. Sortiere alle Artikel (mengenmäßig stärkste zuerst).
3. Solange die kumulierte Bestellmenge der gewählten Artikel kleiner als die Ziel-Kommissioniermenge ist und noch nicht alle Artikel in der Menge sind, wiederhole folgendes:
   a) Wähle den ersten Artikel der Sortierung, welcher noch nicht in der Menge ist, aus.
   b) Füge diesen Artikel zur Menge der identifizierten Artikel hinzu.
4. Aktualisiere die Menge der Filialen (begrenzt durch Schächte) so, dass die besten Filialen zu den identifizierten Artikeln ausgewählt sind (das heißt, die Kommissioniermenge der identifizierten Artikel wird maximiert).
5. Berechne die Kommissioniermenge der bisher identifizierten Artikel und deren bester Filialen.
6. Solange die berechnete Kommissioniermenge kleiner als die Ziel-Kommissioniermenge ist und noch nicht alle Artikel in der Menge sind, wiederhole folgendes:
   a) Wähle den ersten Artikel der Sortierung, welcher noch nicht in der Menge ist, aus.
   b) Füge diesen Artikel zur Menge der identifizierten Artikel hinzu.
   c) Aktualisiere die Menge der Filialen so, dass die besten Filialen zu den identifizierten Artikeln ausgewählt sind.
   d) Berechne die Kommissioniermenge der bisher identifizierten Artikel und deren bester Filialen.

7. Kann die Zielmenge nicht erreicht werden, dann gib eine Warnung aus.

8. Teile die Filialen und Auftragszeilen der identifizierten Artikel dem PKS zu.

Der im *Fallbeispiel* umgesetzte Algorithmus 4 setzt sich aus Teilen der bereits beschriebenen Algorithmen 2 und 3 in folgender Form zusammen:

Algorithmus 4 Die stärksten Artikel und deren Filialen (2-stufig)

Require: $\mathcal{A}$, $\mathcal{F}$, $\mathcal{Z}$, s_{max}, k_{PKS}
1: Initialisiere Menge $\mathcal{F}_{PKS} \leftarrow leereMenge$
2: Initialisiere Menge $\mathcal{A}_{PKS} \leftarrow leereMenge$
3: $ArtikelListe \leftarrow$ Sortiere $\mathcal{A}$ nach der Bestellmenge aller Auftragszeilen $l \in \mathcal{Z}$
4: Initialisiere Kommissioniermenge k
5: **repeat** ▷ Phase 1
6: $\quad \mathcal{A}_{PKS} \overset{\text{add}}{\leftarrow}$ Hole ersten Artikel aus $ArtikelListe$
7: $\quad k \leftarrow$ SummMenge$(\mathcal{A}_{PKS}, \mathcal{F})$
8: **until** $k \geq k_{PKS}$ **or** $ArtikelListe$ **is** $leer$
9: **if** $k < k_{PKS}$ **then**
10: $\quad$ Warnung: Zielmenge kann nicht erreicht werden!
11: **end if**
12: **repeat** ▷ Phase 2
13: $\quad$ **if** $k < k_{PKS}$ **and** $ArtikelListe$ **is not** $leer$ **then**
14: $\quad\quad \mathcal{A}_{PKS} \overset{\text{add}}{\leftarrow}$ Hole ersten Artikel aus $ArtikelListe$
15: $\quad$ **end if**
16: $\quad FilialListe \leftarrow$ Sortiere $\mathcal{F}$ nach der Bestellmenge der gewählten Artikel
17: $\quad\quad$ ▷ d.h., nach SummMenge$(\mathcal{A}_{PKS}, \{i\})$ $\quad \forall i \in \mathcal{F}$
18: $\quad \mathcal{F}_{PKS} \leftarrow$ Wähle die $x \leq s_{max}$ ersten Filialen aus der $FilialListe$
19: $\quad k \leftarrow$ SummMenge$(\mathcal{A}_{PKS}, \mathcal{F}_{PKS})$
20: **until** $k \geq k_{PKS}$ **or** $ArtikelListe$ **is** $leer$
21: **if** $k < k_{PKS}$ **then**
22: $\quad$ Warnung: Zielmenge kann nicht erreicht werden!
23: **end if**
24: **if** $\frac{|\mathcal{A}_{PKS}|}{z} > sph_{max}$ **then**
25: $\quad$ Warnung: Zu viele SKU bzgl. HRL-Leistung gewählt!
26: **end if**
27: Weise alle Auftragszeilen $l \in \mathcal{Z}$ der Filialen $i \in \mathcal{F}_{PKS}$ mit dem Artikel $j \in \mathcal{A}_{PKS}$ dem PKS zu

4.4.5. Erweiterung: Rollcontainerfüllgrad

Diese zusätzliche Option vermeidet teilweise befüllte Rollcontainern im PKS und erhöht den durchschnittlichen Rollcontainerfüllgrad. Während der Kommissionierung befüllt ein Mitarbeiter auf der PKS Station pro Schacht die Rollcontainer einer Filiale. Sobald ein

Rollcontainer keine weiteren Artikel mehr aufnehmen kann, wird dieser transportfähig gemacht, zum Sammelplatz für das FTF (fahrerlose Transportfahrzeug) geschoben und durch einen neuen, leeren Rollcontainer ersetzt. Dieser Vorgang wiederholt sich, bis alle bestellten Artikel für diese Filiale (welche dem PKS Bereich zur Kommissionierung zugeteilt wurden) kommissioniert und auf die Rollcontainer verteilt sind. Deshalb sind die letzten Rollcontainer jedes PKS-Schachts möglicherweise nur zum Teil befüllt. Dies soll durch diese Erweiterung vermieden werden.

Vor der Auftragszeilenverteilung wird die Zielmenge erhöht. Diese Erhöhung ist von der Auftragsstruktur abhängig und kann entweder über einen parametriesierbaren Prozentsatz oder eine variable Berechnung[6] erfolgen. Sind alle Auftragszeilen auf PKS und TKS verteilt, wird der Füllgrad der PKS-Rollcontainer basierend auf Kommissionierreihenfolge und Volumen vorberechnet. Durch diese Vorberechnung ist bekannt, wie viele Rollcontainer für eine Filiale wahrscheinlich benötigt werden und wie gut das Volumen der Rollcontainer voraussichtlich ausgenutzt wird. Der letzte Rollcontainer einer Filiale, dessen errechneter Füllgrad eine untere Schwelle nicht übersteigt, wird im PKS vermieden, indem die Auftragszeilen dieses Rollcontainers dem TKS zugewiesen werden.

Problematisch ist bei dieser Erweiterung die Genauigkeit der Vorberechnung des Rollcontainerfüllgrades über das Volumen. Die genaue Reihenfolge innerhalb einer Warengruppe wird von der PKS Steuerung nicht garantiert und somit kann auch kein Packbild vorberechnet werden. Es ist jedoch die Sequenz der Warengruppen und die Sequenz gleichartiger Artikel innerhalb dieser Warengruppen bekannt. Bei der Vorberechnung wird ein Rollcontainer immer bis zu einem aus Erfahrungswerten bestimmten Prozentsatz (z.B. 85 Prozent) seines rechnerischen Volumens mit Waren befüllt. Danach wird der nächste „leere“ Rollcontainer für die Vorberechnung herangezogen. Dies wird wiederholt, bis alle Artikel einer Bestellung einer Filiale auf Rollcontainer verteilt sind. Stimmt die Vorberechnung des Füllgrades der Rollcontainer nicht hinreichend gut mit der Realität überein, so werden zu viele oder zu wenige Auftragszeilen verschoben.

Andererseits ist die Erhöhung der Zielmenge kritisch, um schlussendlich die gewünschte Kommissioniermenge zu erreichen. Eine erhöhte Zielmenge bedeutet zusätzlich, dass bei gleicher Auftragsstruktur mehr verschiedene Artikel im PKS kommissioniert werden müssen. Dies stellt einen Konflikt zum eigentlichen Ziel der Auftragszeilenverteilung dar. Auch wenn das nachträgliche Verschieben von Auftragszeilen die Zielmenge wieder reduziert, bleibt die Anzahl verschiedener Artikel höher als nötig.

Die Erhöhung des Rollcontainerfüllgrades ist daher nicht so einfach mit den bisherigen Varianten der Auftragszeilenverteilung kombinierbar ohne deren Ergebnisse zu verschlechtern. Diese optionale Erweiterung verfolgt ein neues Ziel. Die Entwicklung einer neuen Variante mit adaptiertem Ziel kann diesen Zielkonflikt lösen.

[6] Anmerkung: Es wird davon ausgegangen, dass durchschnittlich an jedem Schacht ein halbvoller Rollcontainer anfällt und jeder Rollcontainer mit einer durchschnittlichen Anzahl an Artikeln befüllt ist.

4.5. Bewertung der Varianten

Die Bewertung der Varianten erfolgt anhand der *Testdaten* des *Fallbeispiels*. Die Abkürzungen *sFuA*, *sAuF* und *Phase* stehen für drei Varianten der Auftragsverteilung (Phase steht für sAuF 2-stufig). Siehe dazu Abschnitt 4.4. Eine Berechnung erfolgt immer pro Kommissionierwelle pro Tag je Bereich. Es stehen insgesamt 116 verschiedene Kommissionierwellen an Testdaten für die Bewertung zur Verfügung. Pro Bereich sind das 14 Tage zu je 4 Wellen (plus einmal 8 Wellen). Im GKB können 0 bis 4 und im UKB 0 bis 3 PKS-Stationen gleichzeitig betrieben werden. Bei den TKS-Stationen sind es im GKB 1 bis 8 und im UKB 1 bis 4 Stationen. Es muss immer zumindest eine TKS-Station betrieben werden. Somit sind in Summe rechnerisch 3296 Kommissionierwellen-Szenarien denkbar. Ein Szenario scheidet aus, wenn es nicht genügend (PKS-fähige) Artikel beinhaltet, um alle aktiven PKS-Stationen ausreichend zu versorgen, oder wenn ein Szenario die maximale Dauer der Kommissionierwelle überschreitet. Szenarien mit keiner PKS-Station sind für die Bewertung nicht von Interesse.

4.5.1. Kennzahlen

Es wurden Kennzahlen definiert, um den Vergleich der Varianten (sFuA, sAuF, Phase) zu erleichtern. Die Kennzahlen beziehen sich immer auf die Berechnung einer Kommissionierwelle. Sie sind in Tabelle 3 mit Beschreibung gelistet. Am wichtigsten für die Auftragszeilenverteilung ist die Anzahl unterschiedlicher Artikel (SKU) und durchschnittliche Anzahl an Artikeln pro Stunde (SKU/h).

Zur Kennzahl der Rechenzeit ist zu sagen, dass die Berechnungen auf einem Laptop *Dell Latitude* E6520 (mit folgenden Leistungseigenschaften: *Intel® Core™* i5-2410M mit 2,3 GHz und 4 GB RAM) durchgeführt wurden. Als Betriebssystem diente *Windows 7* (32-Bit) mit Service Pack 1 und als Datenbank eine *Oracle 11g* Datenbank. Die Angabe der Rechenzeit dient zum direkten Vergleich des PL/SQL Quellcodes der im Rahmen dieser Arbeit für das *Fallbeispiel* umgesetzt wurde.

4.5.2. Maximale Anzahl an Stationen

Für die Bewertung werden im ersten Schritt alle Wellen mit maximaler Anzahl aktiver Stationen berechnet. Das heißt, immer mit 4 PKS- bzw. 8 TKS-Stationen im Bereich GKB und mit 3 PKS- bzw. 4 TKS-Stationen im Bereich UKB. Tabelle 4 beinhaltet die Mittelwerte der Kennzahlen über alle ausgewerteten Tage. Es werden alle drei Varianten untereinander verglichen.

Die Ergebnisse in Tabelle 4 zeigen, dass die Lösungen mit Ausnahme der Rechenzeit sehr nahe beieinander liegen. Das gewünschte GVE-Ziel wird genügend genau erreicht (unter einem Prozent Abweichung). Die Variante sFuA weist dem PKS Bereich im Durchschnitt mehr Kommissioniermenge zu, da sie durch ihren Ablauf die GVE-Kommissioniermenge nur durch das Hinzufügen des nächsten Artikels in der sortierten Liste regulieren kann. Die wichtigste Kennzahl, die Anzahl verschiedener SKU ist für sAuF und Phase beinahe gleich und sFuA benötigt im Durchschnitt etwas mehr Artikel. Die Anzahl an Filialen ist für die Varianten ungefähr gleich. Auch die folgenden

Tabelle 3: Kennzahlen der Auftragszeilenverteilung

Kennzahl	**Beschreibung**
Abweichung GVE-Ziel	Gibt an, wie stark die Kommissionierleistung an der PKS vom gewünschten Zielwert in Prozent abweicht.
Abweichung Wellendauer (WD)	Gibt an, wie stark die voraussichtliche Kommissionierzeit an der PKS von der maximalen Wellendauer abweicht.
unterschiedliche SKU	Gibt an, wie viele verschiedene Artikel im PKS kommissioniert werden.
unterschiedliche Filialen	Gibt an, wie viele unterschiedliche Filialen im PKS kommissioniert werden.
voraussichtliche Rollcontainer	Gibt an, wie viele Rollcontainer voraussichtlich im PKS entstehen werden.
durchschnittlicher Füllgrad (FG)	Gibt den durchschnittlichen Füllgrad der Rollcontainer an.
GVE pro Rollcontainer (GVE/RC)	Gibt an, wie viele GVEs durchschnittlich auf einen Rollcontainer kommissioniert werden.
GVE pro SKU (GVE/SKU)	Gibt an, wie viele GVEs durchschnittlich pro Artikel kommissioniert werden.
SKU pro Stunde (SKU/h)	Gibt an, wie viele Artikel durchschnittlich pro Stunde kommissioniert werden.
Rechenzeit (RZ) in Sekunden	Gibt an, wie lange für die Berechnung einer Kommissionierwelle gerechnet wird (sec).

drei Kennzahlen (GVE/RC, GVE/SKU und SKU/h) liegen sehr nahe beieinander. Eine Ausnahme ist die benötigte Rechenzeit. Hier erreicht die Variante sFuA den eindeutig besten Wert aufgrund der Einfachheit des Vorgehens. Variante sAuF benötigt so viele Iterationen wie Artikel zur Zielerfüllung benötigt werden. Bei jeder Iteration werden passende Filialen bestimmt, was in Summe viel Zeit in Anspruch nimmt. Die Variante Phase benötigt weit weniger dieser rechenintensiven Iterationen, da sie über ein Initialisierungsphase verfügt, in der eine große Anzahl an Artikeln im Vorhinein festgelegt wird, ohne das die besten Filialen bestimmt werden.

Das Diagramm in Abbildung 6 zeigt, wie die benötige Rechenzeit mit einer höheren Anzahl an benötigten Artikeln steigt. Bei der Variante sFuA ist nur ein kleiner Anstieg zu verzeichnen, wobei der Anstieg der Variante sAuF wesentlich höher ausfällt. Die Variante Phase ist bei wenigen Filialen und vielen Artikel meist schneller als Variante sFuA, jedoch nicht wenn viele Filialen mit wenigen Artikel in einer Kommissionierwelle vorkommen. Es werden dann nach der Initialisierungsphase noch entsprechend viele Iterationen mit Ermittlung der Filialen benötigt, was die Rechenzeit erhöht.

Das Diagramm in Abbildung 7 zeigt, wie sich die Rechenzeit verändert, wenn sich die Anzahl an Filialen ändert. Je mehr Filialen benötigt werden, desto mehr Schächte

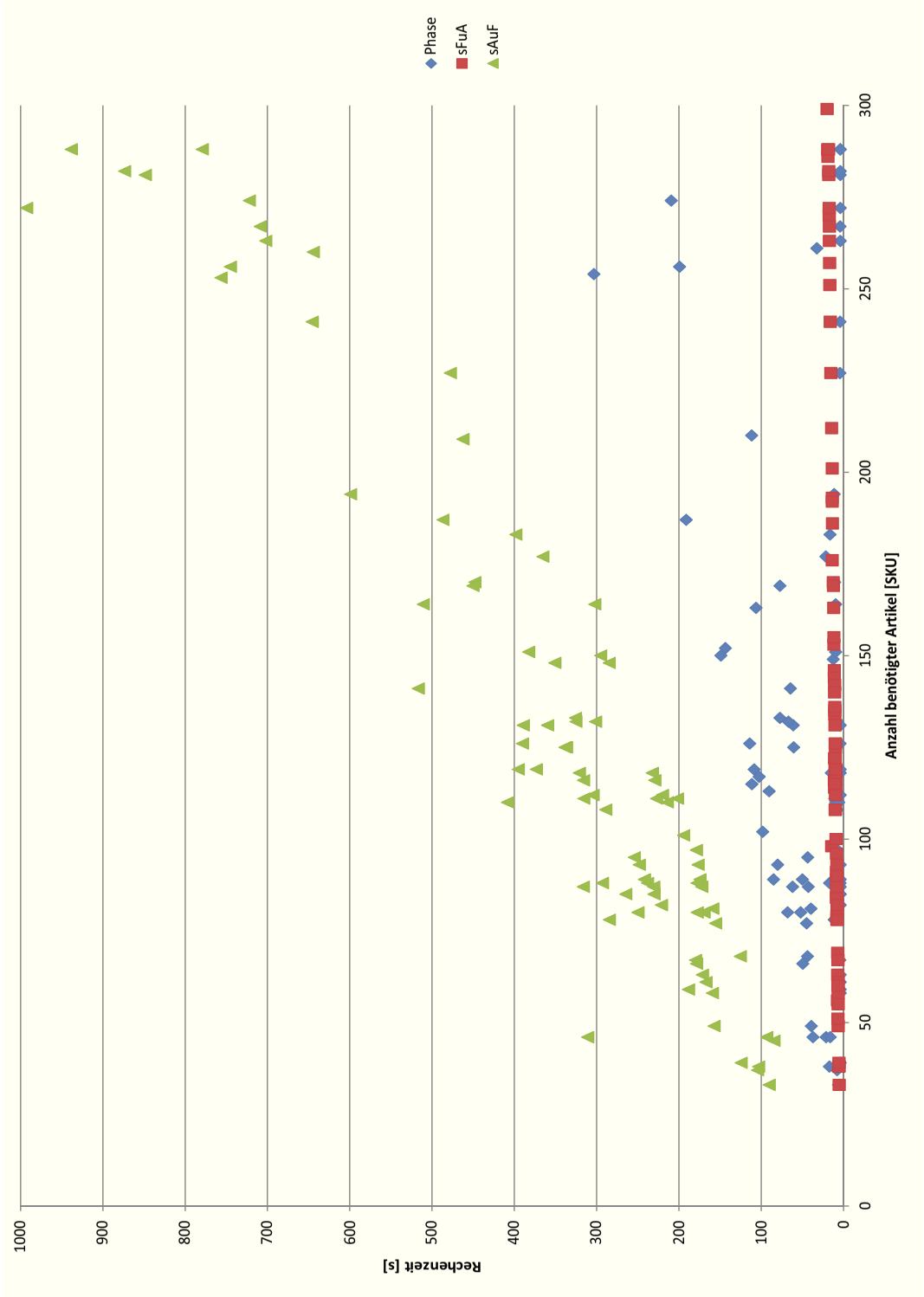


Abbildung 6: Diagramm - Rechenzeit zu Anzahl Artikel

Tabelle 4: Ergebnisse der Auftragszeilenverteilung (maximale Anzahl an Stationen)

Kennzahlen	**Varianten**		
(Mittelwert)	**sFuA**	**sAuF**	**Phase**
Abweichung GVE-Ziel	0,537%	0,273%	0,273%
SKU	85,026	84,860	84,851
Filialen	26,798	26,789	26,789
GVE/RC	48,206	48,255	48,281
GVE/SKU	13,240	13,243	13,244
SKU/h	56,846	56,792	56,791
Rechenzeit [s]	7,955	232,465	9,495

sind insgesamt, über alle Stationen verteilt, in Verwendung. Es wird nur ein Ausschnitt der gesamten Daten angezeigt. Die Varianten Phase und sFuA werden verglichen. Die Variante sFuA wird stärker von einer Erhöhung der Filialen beeinflusst und die Variante Phase kann in vielen Fällen schneller eine Lösung berechnen.

4.5.3. Mittlere Anzahl an Stationen

Als nächstes wird eine mittlere Anzahl an aktiven Stationen für die Bewertung untersucht. Als Ziel soll eine definierte Wellendauer möglichst gut ausgenutzt werden. Es soll mit 0 bis 2 PKS- bzw. 1 bis 8 TKS-Stationen im Bereich GKB und mit 0 bis 2 PKS- bzw. 1 bis 4 TKS-Stationen im Bereich UKB gearbeitet werden. Aus diesen Einstellungen resultieren die gemittelten Kennzahlen in Tabelle 5.

Tabelle 5: Ergebnisse der Auftragszeilenverteilung (mittlere Anzahl an Stationen)

Kennzahlen	**Varianten**		
(Mittelwert)	**sFuA**	**sAuF**	**Phase**
Abweichung GVE-Ziel	0,458%	0,186%	0,210%
Abweichung WD (soll)	-7,66%	-7,66%	-7,66%
Abweichung WD (ist)	-7,25%	-7,49%	-7,46%
SKU	156,532	153,564	153,713
Filialen	19,957	20,000	20,000
GVE/RC	52,483	52,868	52,939
GVE/SKU	19,067	19,589	19,565
SKU/h	43,353	42,554	42,593
Rechenzeit [s]	11,761	431,619	48,445

Auch bei einer mittleren Anzahl an Stationen zeigt sich ein ähnliches Verhalten der Varianten. Es ist zu beachten, dass bei maximaler Anzahl an verfügbaren Stationen mit durchschnittlich 9,6 aktiven Stationen gearbeitet wird. Bei einer mittleren Anzahl sind es nur noch rund 5 aktive Stationen, welche die selbe Kommissioniermenge in längerer Zeit abarbeiten müssen. Deshalb steigt die Anzahl der verschiedenen zu kommissionierenden

Abbildung 7: Diagramm - Rechenzeit zu Anzahl Filialen

SKU im PKS. Die Varianten Phase und sAuF können nun deutlich bessere Kennzahlen-Werte als die Variante sFuA erreichen (als Beispiel Kennzahl SKU: bei maximaler Anzahl Stationen durchschnittlich 0,15 SKU besser - bei mittlerer Anzahl Stationen ungefähr 3 SKU besser). Sie benötigen dafür jedoch verhältnismäßig mehr Rechenzeit.

Eine nach Bereichen (gekühlt und ungekühlt) getrennte Auswertung der Kennzahlen zeigt, dass die Verbesserung hauptsächlich im GKB erzielt wird (siehe Tabelle 6).

Tabelle 6: Ergebnisse der AZV nach Bereich (mittlere Anzahl an Stationen)

Kennzahlen	**UKB**			**GKB**		
(Mittelwert)	**sFuA**	**sAuF**	**Phase**	**sFuA**	**sAuF**	**Phase**
Abweichung GVE-Ziel	0,652%	0,211%	0,210%	0,293%	0,165%	0,210%
Abweichung WD (soll)	-9,88%	-9,88%	-9,88%	-5,79%	-5,79%	-5,79%
Abweichung WD (ist)	-9,33%	-9,69%	-9,69%	-5,51%	-5,63%	-5,59%
SKU	146,442	146,186	146,186	165,039	159,784	160,059
Filialen	17,977	17,860	17,860	21,627	21,804	21,804
GVE/RC	48,376	48,622	48,638	55,945	56,448	56,565
GVE/SKU	21,782	22,013	22,012	16,779	17,546	17,502
SKU/h	42,043	41,973	41,973	44,459	43,044	43,116
Rechenzeit [s]	10,999	438,172	13,927	12,404	426,095	77,549

Daraus lässt sich schließen, dass die Auftragsstruktur im gekühlten Bereich wesentlich besser für Varianten, welche die mengenmäßig stark vertretenen Artikel bevorzugen, geeignet ist. Variante Phase benötigt durchschnittlich wesentlich mehr Rechenzeit für die Berechnung im GKB als im UKB.

4.5.4. Aufteilung Normal- und Aktion

Es wird die Aufteilung nach Normal- und Aktionsartikel mit in die Auftragszeilenverteilung einbezogen. Es werden nur noch die Variante sFuA und Phase miteinander verglichen. Es soll gezeigt werden, wie sich die Ergebnisse durch diese zusätzliche Bedingung verändern. Die Ergebnisse sind in Tabelle 7 zu sehen.

Wenn eine Trennung nach Normal- und Aktionsartikel durchgeführt wird, verdoppelt sich im schlechtesten Fall die Anzahl an Filialen bzw. Aufträgen. Jede Filiale besitzt dann pro Kommissionierwelle zwei Aufträge. Ein Auftrag beinhaltet alle Normalartikel und der andere alle Aktionsartikel. Diese Aufträge müssen in verschiedenen Schächten kommissioniert werden, da eine Vermischung der Artikel in den Rollcontainern nicht erlaubt ist. Für die Problemstellung bedeutet dies, dass aus einer „alten“ Filiale bis zu zwei „neue“ Filialen entstehen können. Das heißt, die Problemstellung steigt in ihrer Größe an. Die zu kommissionierenden Mengen im PKS Bereich bleiben die selben wie zuvor. Es wird jedoch schwerer die Zielmenge im PKS mit so wenig wie möglich verschiedenen Artikeln zu erreichen, da die Anzahl an verfügbaren Schächten unverändert bleibt.

Positiv wirkt sich die Trennung auf die Abweichung zum GVE-Ziel aus. Da die Filialen kleiner Artikelmengen bestellen und mehr Filialen zur Auswahl stehen, kann das

GVE-Ziel noch genauer erreicht werden. Alle anderen Kennzahlen verschlechtern sich erwartungsgemäß. Besonders die Erhöhung der SKU Anzahl wirkt sich für die Kommissionierung im PKS Bereich sehr schlecht aus. Es müssen mehr Artikel und somit auch Paletten aus dem HRL ausgelagert werden. Auf dies weist auch der Anstieg der Kennzahl SKU/h hin. Es werden mehr Palettenwechsel notwendig sein und weniger Artikel können im Schnitt in einen Rollcontainer gepackt werden. Besonders auffällig ist der Anstieg der benötigen Rechenzeit. Die Variante Phase kann im Vergleich zu sFuA bei der Kennzahl SKU mit Trennung noch besser abschneiden. Variante Phase spart nun im Schnitt circa einen Artikel pro Kommissionierwellenberechnung ein.

Tabelle 7: Ergebnisse der Auftragszeilenverteilung (Trennung N/A)

Kennzahlen	**ohne Trennung**		**Trennung N/A**		**Zuwachs**	
(Mittelwert)	**sFuA**	**Phase**	**sFuA**	**Phase**	**sFuA**	**Phase**
Abweichung GVE-Ziel	0,537%	0,273%	0,513%	0,260%	-0,025%	-0,013%
SKU	85,026	84,851	100,781	99,746	15,754	14,895
Filialen	26,798	26,789	36,219	36,342	9,421	9,553
Rollcontainer	92,825	92,491	99,632	98,395	6,807	5,904
Füllgrad	76,58%	76,56%	74,23%	74,03%	-2,34%	-2,53%
GVE/RC	48,206	48,281	45,660	46,062	-2,546	-2,219
GVE/SKU	13,240	13,244	11,672	11,812	-1,568	-1,432
SKU/h	56,846	56,791	62,651	62,159	5,805	5,369
Rechenzeit [s]	7,955	9,495	27,329	74,861	19,373	65,366

4.5.5. Rollcontainerfüllgrad

Es wird geprüft, wie sich die Verbesserung des Rollcontainerfüllgrades auf die Kennzahlen auswirkt. Es ist zu erwarten, dass der Rollcontainerfüllgrad ansteigt und die Anzahl an benötigten Rollcontainern abnimmt. Ein Rollcontainer soll im *Fallbeispiel* maximal zu 85 Prozent seines rechnerischen Volumens befüllt werden. Danach wird mit der Befüllung des nächsten Rollcontainers begonnen. Tabelle 8 zeigt die Ergebnisse.

Wie erwartet steigt der Füllgrad der Rollcontainer stark an und liegt nun nahe am erreichbaren Optimum von 85 Prozent. Die Anzahl der benötigen Rollcontainer sinkt und die benötigte Rechenzeit steigt nur schwach an. Problematisch ist der Anstieg der verschiedenen Artikel (SKU) und der Kennzahl SKU/h. Diese erhöhen sich um 47 bzw. 57 Prozent. Es werden in der Version mit Rollcontainerfüllgrad-Verbesserung viel mehr verschiedene Artikel, als eigentlich für die angestrebte Kommissioniermenge nötig wären, im Bereich PKS kommissioniert. Durch diese „Verschlechterung“ ist bei manchen Kommissionierwellen der Testdaten sogar mit Leistungseinbrüchen im Bereich PKS zu rechnen. Ferner kann die gewünschte Kommissioniermenge im Bereich PKS, durch die zusätzliche Verbesserung des Füllgrades der Rollcontainer, nicht mehr genügend genau erreicht werden. Im Schnitt werden zu viele Artikel kommissioniert. In einigen Fällen (Kommissionierwellenberechnungen) liegt die tatsächliche Kommissioniermenge unter

Tabelle 8: Ergebnisse der Auftragszeilenverteilung (Rollcontainerfüllgrad)

Kennzahlen	**ohne RC-FG**		**mit RC-FG**		**Zuwachs**	
(Mittelwert)	**sFuA**	**Phase**	**sFuA**	**Phase**	**sFuA**	**Phase**
Abweichung GVE-Ziel	0,534%	0,263%	7,082%	6,565%	6,548%	6,303%
SKU	86,568	86,387	127,739	127,297	41,171	40,910
Filialen	27,477	27,468	25,856	25,838	-1,622	-1,631
Rollcontainer	95,225	94,883	85,766	85,261	-9,459	-9,622
Füllgrad	76,70%	76,70%	83,19%	83,25%	6,49%	6,55%
GVE/RC	48,419	48,493	54,066	54,095	5,648	5,601
GVE/SKU	13,541	13,545	10,107	10,105	-3,434	-3,440
SKU/h	50,162	50,105	78,926	78,728	28,764	28,623
Rechenzeit [s]	8,035	9,655	10,081	10,566	2,046	0,911

dem Zielwert. Der Grund dafür ist, dass vor der Auftragszeilenverteilung nicht genau bestimmt werden kann, wie viele Rollcontainer einen zu geringen Füllgrad aufweisen werden. Wird die gewünschte Kommissioniermenge zu sehr verfehlt, passen Wellendauer und benötige Kommissionierzeit nicht mehr zusammen. Im schlimmsten Fall kann die Tourenabfahrtszeit nicht mehr eingehalten werden. Auf jeden Fall ist die Arbeitslast nicht mehr gleichmäßig auf PKS und TKS Stationen verteilt.

4.5.6. Diagramme

Die folgenden Diagramme zeigen die Ergebnisse aus anderen Perspektiven. Sie zeigen zwei allgemein gültige Eigenschaften der Auftragszeilenverteilung im Palettenkommissioniersystem. Das Diagramm in Abbildung 8 stellt die Anzahl benötigter Artikel in ein Verhältnis zur Anzahl der aktiven Stationen. Je mehr Stationen im PKS aktiv sind, desto mehr verschiedene Artikel müssen durchschnittlich im PKS kommissioniert werden, um das Kommissionierziel zu erreichen. Für diese Auswertung wurden der Durchschnittstag (DST) und der Spitzentag (SPT) herangezogen. Außerdem ist die Auswertung nach den beiden Lagerbereichen GKB und UKB getrennt. Es ist festzustellen, dass im ungekühlten Bereich der Anstieg an verschiedenen Artikeln größer ist, als im gekühlten Bereich. Dies ist darauf zurückzuführen, dass die Arbeitslast im GKB auf mehrere Stationen verteilt werden kann, als im UKB. Es ist anzumerken, dass der Betrieb mit 4 PKS Stationen im UKB im *Fallbeispiel* nicht vorgesehen ist, jedoch berechnet wurde, um das folgende Diagramm vollständig darstellen zu können.

Als Abschluss folgt ein Diagramm über die benötigte Kommissionierzeit im Verhältnis zu den aktiven PKS Stationen. Auch dieses Diagramm bezieht sich auf den DST und den SPT. Dem Diagramm liegen die Ergebnisse der selben Berechnung wie im vorherigen Diagramm zu Grunde. Es ist in Abbildung 9 zu sehen. Je mehr Stationen aktiv sind, desto weniger Kommissionierzeit wird benötigt. Die Schwierigkeit bei der Berechnung liegt darin, genau jede Kombination von PKS und TKS Stationen zu finden, welche die Wellendauer mit ihrer Kommissionierzeit möglichst gut ausfüllt, alle Nebenbedingungen einhält und zugleich die Mitarbeiteranzahl über den Tag hinweg minimiert.

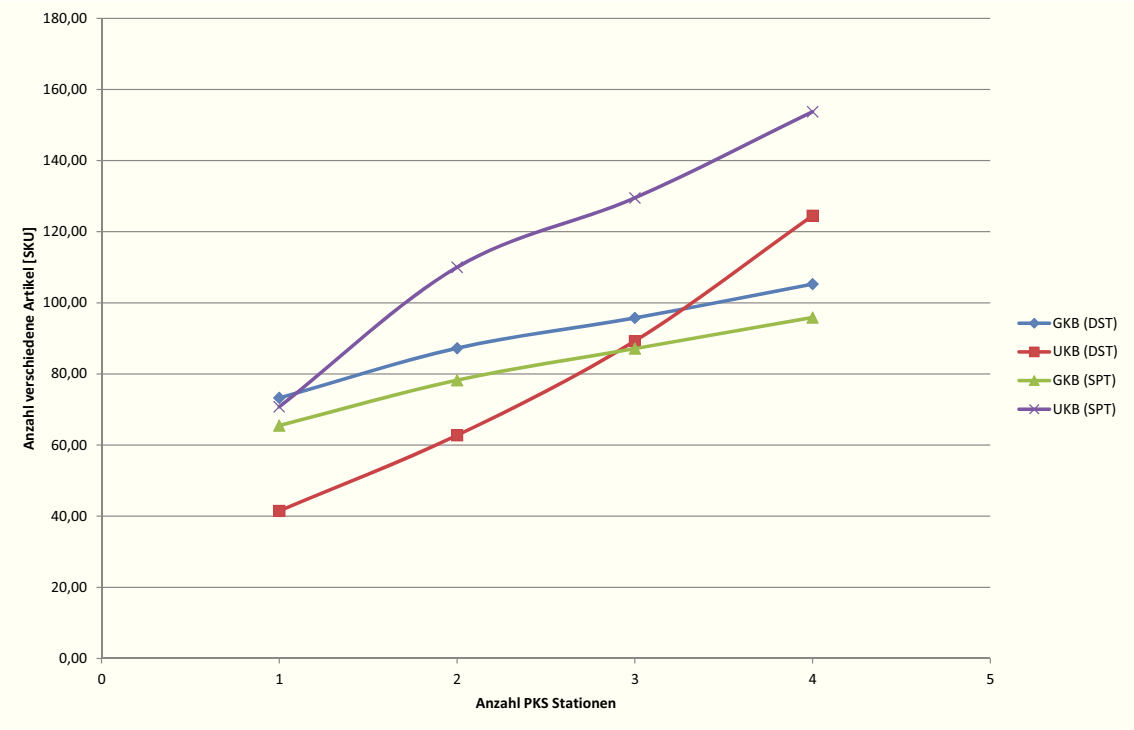


Abbildung 8: Diagramm - benötigter Artikel zu Anzahl aktiver Stationen

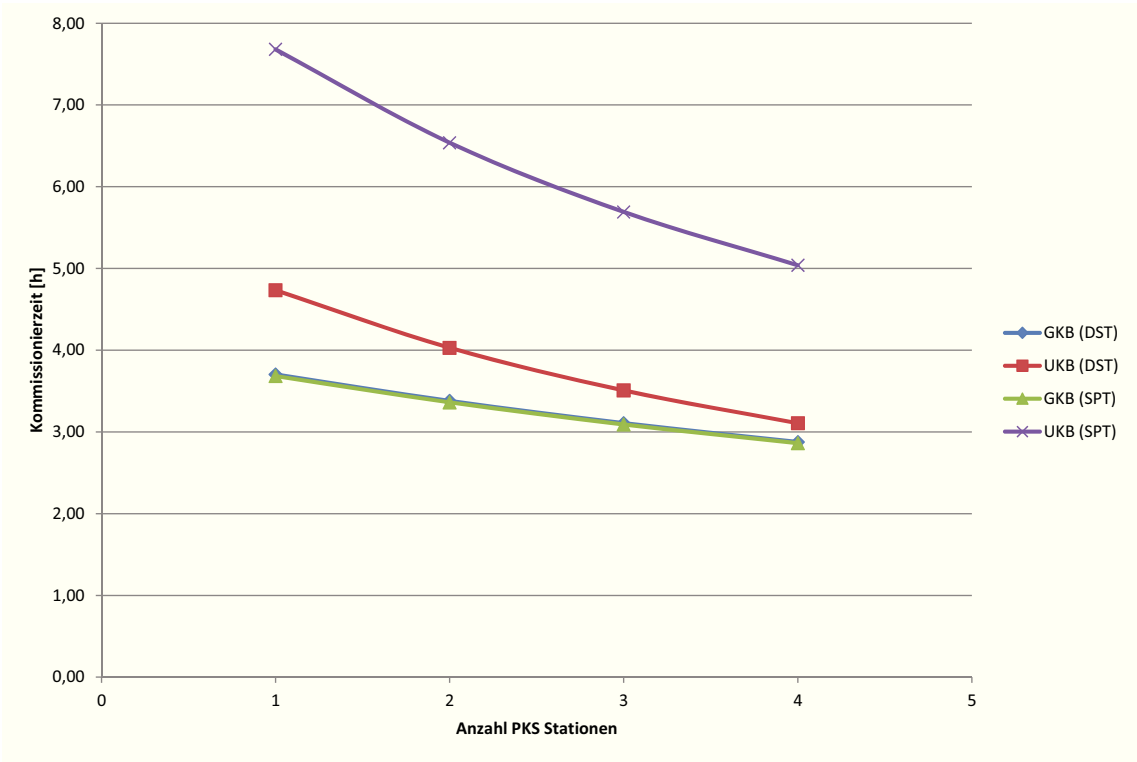


Abbildung 9: Kommissionierzeit zu Anzahl aktiver Stationen

Auszüge der berechneten Daten sind unter Anhang C zu finden.

4.5.7. Fazit

Für das *Fallbeispiel* bedeutet dies, dass Variante sFuA gute Ergebnisse in kurzer Zeit liefert. Es werden jedoch teilweise zu viele verschiedene Artikel als nötig im PKS kommissioniert, da bevorzugt die mengenmäßig stärksten Filialen gewählt werden. Es wird nicht geprüft, ob für mengenmäßig schwächere Filialen die selbe Kommissioniermenge mit weniger verschiedenen Artikel kommissioniert werden könnte.

Variante sAuF liefert im Schnitt etwas bessere Ergebnisse, jedoch in unakzeptabler Zeit, wenn die Anzahl an benötigten Artikeln steigt. Es werden bevorzugt die mengenmäßig stärksten Artikel im PKS kommissioniert werden. Bei ungünstiger Auftragsstruktur und wenn mehr Filialen als aktive Schächte in einer Kommissionierwelle zur Wahl stehen, werden unter Umständen mehr verschiedene Artikel als nötig im PKS Bereich kommissioniert.

Variante Phase liefert vergleichbar gute Ergebnisse wie Variante sAuF in sehr viel kürzerer Zeit. Sie ist in jedem Fall der Variante sAuF vorzuziehen. Je nach dem, ob die Zeit oder die Lösungsgüte von größerer Bedeutung ist, sollte im *Fallbeispiel* Variante sFuA bezüglich der Zeit oder Phase bezüglich der Lösungsgüte gewählt werden. Diese Schlussfolgerung beruht lediglich auf den Auswertungen der Ergebnisse der *Testdaten des Fallbeispieles*. Bei einer anderen Auftragsstruktur würden die Ergebnisse möglicherweise anders ausfallen.

Keines der hier vorgestellten und umgesetzten Vorgehen garantiert, dass immer das Optimum gefunden wird. Die Beispiele, welche im Anhang unter Abschnitt A.1 und Abschnitt A.2 zu finden sind, zeigen dies. Hierfür müsste ein Vorgehen, das den gesamten Lösungsraum untersucht, umgesetzt werden. Ein solches Vorgehen ist aufgrund der benötigten Rechenzeit für den praktischen Einsatz im *Fallbeispiel* unbrauchbar.

Die Trennung nach Aktions- und Normalartikel erhöht die Größe der Problemstellung. Es ist im Fallbeispiel (bei gleichbleibender Kommissioniermenge und Auftragsstruktur) mit höherem Rechenaufwand und schlechteren Kennzahlenwerten zu rechnen.

Die Verbesserung des Rollcontainerfüllgrades ist mit den vorgestellten Verfahren nur bedingt erfolgreich. Der Füllgrad wird auf Kosten des eigentlichen Zieles, die Anzahl verschiedener Artikel im PKS-Bereich so gering wie möglich zu halten, erhöht. Die hier behandelten Varianten der Auftragszeilenverteilung sind von ihrem Konzept her nicht auf eine Maximierung des Füllgrades ausgelegt. Ein Vorgehen, welches auf die Optimierung des Rollcontainerfüllgrad ausgelegt ist, könnte vermutlich wesentlich bessere Lösungen liefern.

5. Auftragsverteilung im PKS

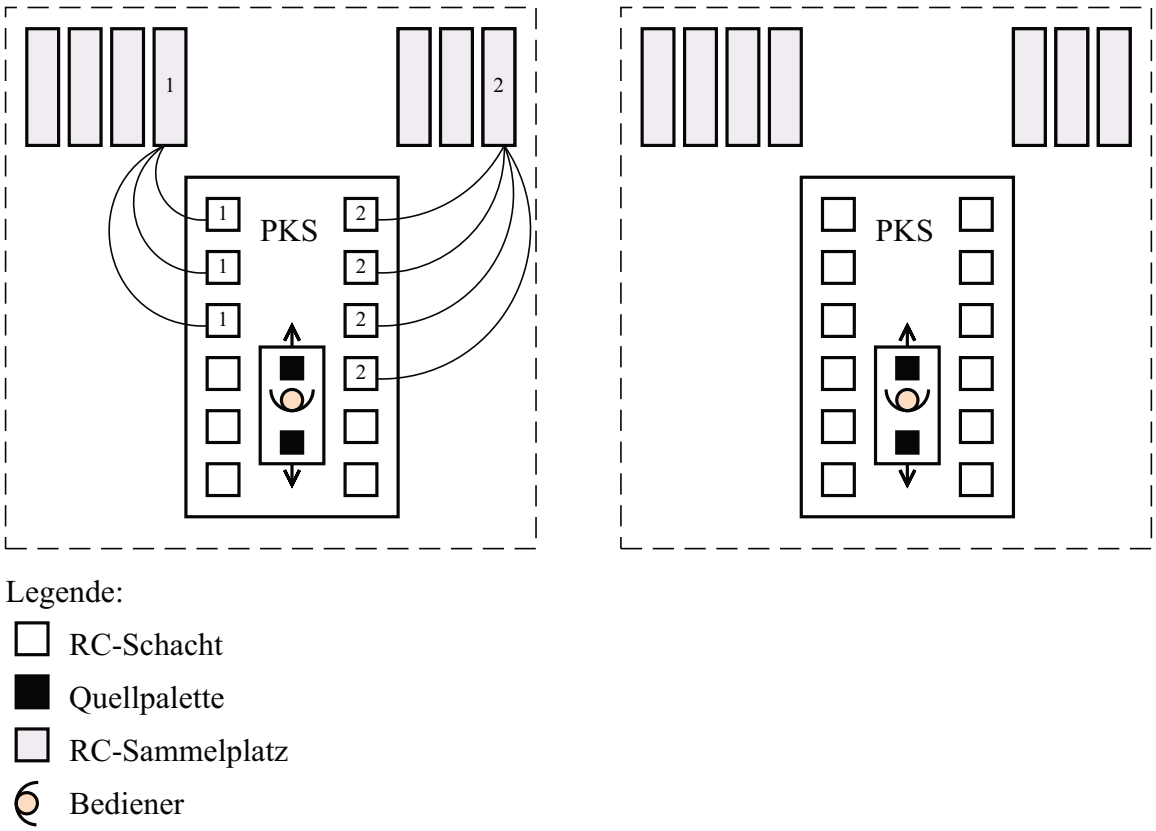


Abbildung 10: PKS-Station

Der schematische Aufbau einer PKS-Station wird in Abbildung 10 gezeigt und deren Charakteristik nun näher beschrieben. Der Bediener steht auf einem vertikal verschiebbaren Wagen. Dieser Wagen kann bis zu zwei Quellpaletten aufnehmen. Die Aufnahme und Entsicherung von Quellpaletten erfolgt durch den Bediener und benötigt Zeit. Der Wagen bewegt sich automatisiert zu den Schächten, in welche die aktuelle am Wagen befindliche Ware kommissioniert wird. Der Bediener führt die ihm visualisierte Kommissioniertätigkeit aus und bestätigt diese dem System. Danach werden die nächsten Schächte angefahren. Wenn keine Waren der aktuellen Quellpaletten mehr kommissioniert werden sollen oder beide Quellpaletten leer sind, werden die Quellpaletten getauscht. Nicht völlig geleerte Paletten werden entweder ins HRL zurück transportiert und eingelagert oder zum nächsten Ziel (z.B. die benachbarte PKS-Station) weitergeleitet. Eine Leerpalette (LP) wird von der Fördertechnik zu einem Leerpalettenstapler oder zu einer Abgabestelle transportiert, wo die LP von der Fördertechnik genommen und auf einem speziellen Leerpaletten-Sammelplatz abgelegt wird. Leerpaletten entlasten das System, da kein Rücktransport ins HRL stattfindet.

Der Bediener kommissioniert die geforderte GVE-Stückanzahl des Artikels in Roll-

container (RC). Diese befinden sich in den Schächten und werden, sobald keine weiteren Artikel mehr in den RC passen, durch eine leeren RC ausgetauscht. Dieser manuelle Schritt erfolgt durch einen Mitarbeiter und benötigt Zeit. Pro Schacht und Welle wird immer nur eine Filiale kommissioniert. Volle RC werden auf RC-Sammelplätze gestellt. Diese Sammelplätze werden vom FTF (fahrerloses Transportfahrtzeug) angefahren. Das FTF nimmt die RCs (bis zu drei Stück) auf und transportiert sie zu den entsprechenden Touren-Sammelplätzen im Warenausgang (WA). Um das FTF effizienter zu betreiben, sollen die RC-Sammelplätze immer „tourenrein“ gehalten werden. Bei der Schachtzuordnung der Filialen auf der PKS-Station wird darauf geachtet, dass Filialen einer Tour räumlich nahe zusammen gehalten und wenn möglich nur auf einer Seite der Station positioniert sind. Dies erleichtert die Arbeit der Bediener.

Aus der Beschreibung der PKS-Station geht hervor, dass einige Tätigkeiten Zeit kosten bzw. einen Aufwand darstellen und somit nach Möglichkeit vermieden werden sollen. Außerdem müssen Vorgaben eingehalten werden, um nachfolgende Prozesse effizient zu betreiben. All diese Informationen zu erfassen und zu einem mathematischen Modell zu verarbeiten, ist Teil dieser Arbeit. Durch sich ständig ändernde Vorgaben und Bedingungen wird dieser Schritt erschwert. Besonders in der Planungsphase eines Projektes sind noch nicht alle Vorgaben fixiert.

5.1. Problemstellung

Um die Problemstellung der Auftragsverteilung im PKS bzw. die Teilsystemoptimierung PKS klar zu definieren, werden zuerst die Rahmenbedingungen vorgegeben. Das Palettenkommissioniersystem besteht aus p aktiven PKS-Stationen mit jeweils s Schächten. Die aktiven Stationen werden von der zuvor durchgeführten Auftragszeilenverteilung vorgegeben. Diese bestimmt auch die f Filialen bzw. Anzahl an Filialaufträgen und Auftragszeilen, welche das PKS kommissioniert.

Die Aufgabe besteht nun darin, die Filialen[7] auf die einzelnen Stationen der PKS und deren Schächte möglichst optimal zu verteilen. Die Bediener der einzelnen Stationen sollen in Summe gleich viele Artikel kommissionieren. Nichtwertschöpfende Tätigkeiten der Bediener werden vermieden oder auf einem Minimum gehalten. Außerdem wird die räumliche Positionierung der Filialen zueinander bei der Schachtzuteilung miteinbezogen. Die diesbezüglichen Zielvorgaben wurden bereits in der Teilsystemoptimierung definiert und erörtert. Für die Durchführung dieser Aufgabe steht eine begrenzte Zeitdauer zur Verfügung.

Das Problem, x Filialen auf y Schächte zuzuordnen, mit dem Ziel den Materialfluss unter Anbetracht räumlicher Distanzen zu minimieren, ist unter dem Namen Quadratisches Zuordnungsproblem (englisch: Quadratic Assignment Problem, QAP) in der Literatur zu finden. Als Paradebeispiele gelten die Standortbestimmung von Fabriken oder das Verkabeln von elektronischen Geräten. Dabei steht die Minimierung des Flusses zwischen den Komponenten in Abhängigkeit zu ihrer Platzierung im Mittelpunkt. Das Quadra-

[7] Anmerkung: Da jede Filiale i.d.R genau einen Auftrag pro Kommissionierwelle besitzt, wird in dieser Arbeit von der Auftragsverteilung gesprochen.

tic Assignment Problem (QAP) gehört zu den am schwersten lösbaren Problemen der Kombinatorik.

Das hier vorgestellte Problem soll im Gegensatz dazu, die Anzahl an Artikeln und Touren innerhalb einer PKS Station minimieren. Das heißt, es wird innerhalb einer Gruppe von Schächten minimiert. Zusätzlich soll die Kommissioniermenge über alle Stationen (Gruppen von Schächten) hinweg möglichst gleichmäßig verteilt werden. Diese Vorgaben unterscheiden sich signifikant vom klassischen QAP. Ein möglicher Ansatz, wie das hier vorgestellt Problem in ein QAP umgewandelt werden könnte, ist unter Anhang B zu finden. Dort befinden sich auch eine Definition des QAP und die Verweise zur Fachliteratur.

5.2. Zielfunktion PKS

Die Zielfunktion liefert für jede gefundene Lösung der Auftragsverteilung im PKS einen vergleichbaren Wert. Alle angestrebten Ziele werden bei der Berechnung dieses Wertes miteinbezogen. Bei einem Problem mit mehreren Zielkriterien entstehen meist Zielkonflikte und daher müssen die Ziele untereinander gewichtet werden. Es ist zu beachten, dass unterschiedliche Maßeinheiten den Vergleich erschweren oder unmöglich machen. Die Grundidee, wie bei Entscheidungen mit mehreren Zielkriterien vorzugehen ist, stammt aus [26, S. 28 ff].

Für die Optimierung des PKS wird folgende Zielfunktion definiert:

$$\min f(x,y,z) = \sum_{i=1}^{p}(x_i\ g_x + y_i\ g_y + \sigma^2(z)\ g_z)$$

wobei

$$\sigma^2(z) = \frac{1}{p}\sum_{i=1}^{p}(z_i - \bar{z})^2$$

mit

p ... Anzahl aktiver PKS-Stationen
x_i ... Anzahl verschiedener Artikel je Station
y_i ... Anzahl verschiedener Touren je Station
z_i ... Kommissioniermenge in GVE je Station
g_x, g_y, g_z ... Gewichtung der Ziele
σ^2 ... Varianz
$\bar{z}$ ... Mittelwert der Kommissioniermengen

(9)

Folgende Nebenbedingungen müssen für alle aktiven Stationen eingehalten werden:

$$
\begin{aligned}
f_i &\leq s \\
z_i &\leq w \\
r_{PKS} - r_{Tol} \leq \frac{k_i}{z_i} &\leq r_{PKS} + r_{Tol}
\end{aligned}
$$

wobei gilt

$$1 \leq i \leq p$$

mit

$f_i \ldots$ Anzahl zu kommissionierender Filialen der i-ten Station
$s \ldots$ Anzahl an Schächten pro PKS-Station
$z_i \ldots$ Kommissionierzeit der i-ten Station
$k_i \ldots$ Kommissioniermenge der i-ten Station
$w \ldots$ Wellendauer in h
$r_{PKS} \ldots$ Nennleistung PKS-Station in GVE/h
$r_{Tol} \ldots$ Toleranz beim Einhaltung der Nennleistung

(10)

Die erste Nebenbedingung muss eingehalten werden. Es dürfen an einer Station (während einer Kommissionierwelle) nicht mehr Filialen kommissioniert werden, als Schächte verfügbar sind. Außerdem darf die rechnerisch benötigte Kommissionierzeit an einer Station nicht länger als die Wellendauer sein und die Nennleistung soll an jeder Station innerhalb einer definierten Toleranz erreicht werden. Eine Toleranz r_{Tol} wird absolut in GVE/h oder prozentual zur Nennleistung vorgegeben. Die letzten beiden Nebenbedingungen sind als schwache Nebenbedingungen anzusehen. Können die Kommissioniermengen der Filialen möglichst gleichmäßig auf die Stationen verteilt werden, so gelten diese beiden Bedingungen für das *Fallbeispiel* als hinreichend erfüllt.

Zusätzlich sollen Kennzahlen wie durchschnittlich benötigte SKU/h oder Pal/h einen festgelegten Wert nicht überschreiten, um einen reibungslosen Ablauf im Lagersystem zu ermöglichen. Diese Kennzahlen können unabhängig vom Verfahren direkt nach der Auftragsverteilung berechnet werden. Bei Überschreiten der Grenzen wird eine Warnung ausgegeben.

Die Zielfunktion beinhaltet alle primären Zielvorgaben aus der Teilsystemoptimierung. Mit Hilfe der Gewichte g_x, g_y, g_z wird jedem Summanden der Zielfunktion ein Faktor zugewiesen. Dies erlaubt eine Priorisierung und/oder Skalierung der einzelnen Teilziele. Die Anzahl verschiedener Artikel x_i wird an jeder Station minimiert, wobei eine vorgegebene Kommissionierzielmenge je Station möglichst genau erreicht werden soll. Die Anzahl verschiedener Artikel die im gesamten PKS-Bereich pro Kommissionierwelle behandelt werden, minimiert bereits die Auftragszeilenverteilung.

Weiters wird die Anzahl verschiedener Touren y_i an jeder Station minimiert und in einer darauf folgenden Heuristik auf die zwei Seiten der Station verteilt. Dies soll einerseits

die Wegzeiten der fertig befüllten Rollcontainer zu den Sammelplätzen (am besten werden diese Sammelplätze tourenrein gehalten) minimieren und andererseits die Leistung des FTFs verbessern.

Um die Kommissioniermengen z_i der p PKS-Stationen auf einem gleichmäßigen Niveau zu halten, wird die Varianz der Kommissioniermengen in die Zielfunktion mit eingebunden. Somit wird eine mengenmäßig gleiche Auslastung der Stationen gegenüber einer ungleichmäßigen in der Zielfunktion bevorteilt.

Die Minimierung der Palettenwechsel und die Maximierung der Leerpaletten sind direkt vom Ziel der Minimierung der Anzahl verschiedener Artikel abhängig und somit nur indirekt in der Zielfunktion vorhanden. Beide werden von einer geringeren Anzahl verschiedener Artikel je Station und dem gesamten PKS-Bereich positiv beeinflusst. Werden beispielsweise im PKS-Bereich weniger verschiedene Artikel (bei gleichbleibender Kommissioniermenge und gleichbleibender Verteilung des Palettenfaktor unter den Artikeln) behandelt, so sind durchschnittlich mehr Leerpaletten und weniger Palettenwechsel zu erwarten, da durchschnittlich pro Palette aus dem HRL gleich viel oder mehr GVE entnommen werden. Da für die Optimierung keine Bestandsdaten verfügbar sind, kann nicht im Voraus bestimmt werden, wann ein Palettenwechsel nötig ist bzw. wann Leerpaletten anfallen und wann nicht.

Die angestrebte Maximierung des Rollcontainerfüllgrades ist in der Zielfunktion nicht abgebildet, da keine Packbildberechnung im PKS-Bereich angedacht ist. Somit ist die Abschätzung des Füllgrades nur näherungsweise möglich und fließt nur als zusätzliche Option der Auftragszeilenverteilung in den finalen Ablauf des *Fallbeispiels* ein.

Die Zielfunktion ist im Hinblick auf Transparenz und Erweiterbarkeit entworfen. Es ist ohne großen Aufwand möglich, die Zielwerte von einzelnen Lösungen in der Datenbank zu berechnen bzw. abzuschätzen, wenn die benötigten Kennzahlen vorliegen. Das Hinzufügen von weiteren Zielen oder ein Umgestalten der Gewichtung ist leicht möglich.

Mit Hilfe dieser Zielfunktion kann die weitere Optimierung beginnen. Es werden in den folgenden Abschnitten einige Lösungsverfahren vorgestellt, die für das *Fallbeispiel* großteils auch umgesetzt wurden. Den Beginn bilden die exakten Algorithmen.

5.3. Exakte Algorithmen

Das Problem, die x Filialen den y Schächten zuzuordnen (wobei im *Fallbeispiel* immer $x <= y$ gilt) ist kombinatorischer Natur. Allgemein wird von einer *Variation ohne Wiederholungen* gesprochen. Der Spezialfall $x = y$, wird als *Permutation* bezeichnet. Dies tritt im *Fallbeispiel* immer dann ein, wenn alle Schächte mit einer Filiale belegt sind (das heißt, die PKS ist in ihrer Schachtkapazität voll ausgelastet). Ist dies im *Fallbeispiel* nicht der Fall, wird angenommen, das einige Schächte unbelegt bleiben. Eine Permutation der x Filialen liefert in beiden Fällen alle Lösungsmöglichkeiten der Filiale-zu-Schacht Zuordnungsaufgabe.

Die Grundprinzipien und Anregungen für eigene Lösungswege sind für diesen Abschnitt aus der Fachliteratur, auf welche im Literaturverzeichnis unter [4], [14], [15], [22], [24] und [26] verwiesen ist, entnommen.

5.3.1. Enumeration

Durch die Permutation können alle möglichen Filiale-zu-Schacht Zuordnungen in endlicher Zeit aufgezählt und bewertet werden. Wird am Ende die beste aller gefundenen Möglichkeiten gewählt, so ist die beste Lösung gefunden. Dieses Vorgehen wird in der Literatur als *Enumeration* oder *erschöpfende Suche* (englisch: exhaustive search) bezeichnet. Da es immer die beste bzw. optimale Lösung liefert, wird es zu den exakten Algorithmen gezählt.

Die Problematik besteht darin, das die Anzahl an Möglichkeiten unter Verwendung der Permutation mit faktoriellem Wachstum ansteigt. In *Landau*-Notation wird dieses Wachstum mit $\mathcal{O}(n!)$ beschrieben. Beispielhaft fallen für 3 Stationen mit jeweils 12 Schächten $(3 \cdot 12)! = 36! = 3,72 \cdot 10^{41}$ Kombinationsmöglichkeiten an. Könnten eine Milliarde Kombinationen in einer Sekunde berechnen und bewerten werden, so würden ca. $1,18 \cdot 10^{16}$ Milliarden Jahre zur Abarbeitung benötigt werden. Somit findet dieser Ansatz keinen praktischen Einsatz im *Fallbeispiel* und ist nur theoretischer Natur.

5.3.2. Beschränkte Enumeration

Der Ansatz der Enumeration kann durch einige Überlegungen wesentlich verbessert werden. Durch Analyse der Problemstellung wurden Regeln erkannt, die es ermöglichen, Teile der vollständigen Enumeration als ungültig zu identifizieren. Diese Kombinationen müssen dann nicht bewertet werden und verringern die benötige Zeit zur Lösungsfindung. Es ist auch von Vorteil, wenn gegebene Symmetrien ausgenutzt werden. Folgende Begriffe werden an dieser Stellen definiert. Eine Filiale-zu-Schacht-Anordnung weist jedem aktiven Schacht jeder aktiven Station eine Filiale zu (Permutation). Eine Filiale-zu-Station-Anordnung weist jeder aktiven Station maximal s Filialen zu. Dabei spielt die Schachtbelegung innerhalb der Stationen keine Rolle.

Unter Beachtung der Problemstellung der Auftragsverteilung und der zugehörigen Zielfunktion werden folgende Regeln erkannt:

1. Wenn Filialen innerhalb einer Station ihre Plätze (Schächte) tauschen, ändert sich der Wert der Zielfunktion nicht.

2. *Kombinationen ohne Wiederholung* beschreiben alle möglichen Filiale-zu-Station-Anordnungen für eine Station.

3. Wenn Filiale-zu-Station-Anordnungen zwischen Stationen[8] getauscht werden, ändert sich der Wert der Zielfunktion nicht.

4. Wenn die Filiale-zu-Station-Anordnung auf einer Station fixiert wird, kann die Zuteilung der verbleibenden Filialen auf die restlichen Stationen als kleineres Teilproblem gelöst werden.

[8] Anmerkung: Alle Stationen im PKS haben in der Theorie exakt die selben Leistungseigenschaften und Konfiguration.

Durch die erste Regel wird ersichtlich, dass durch eine „vollständige“ Permutation zu viele Kombinationen betrachtet werden. Der Wert der Zielfunktion ändert sich nur, wenn eine Filiale von einer Station auf eine andere wechselt. Würde die konkrete Schachtposition der Filialen innerhalb einer Station[9] mit in die Zielfunktion einfließen, würde dies nicht der Fall sein.

Die Regeln 2 bis 4 der obigen Aufzählung dienen dazu, Symmetrien der Problemstellung auszunützen, um die Anzahl zu bewertender Kombinationen weiter zu reduzieren und eine rekursive Abarbeitung zu ermöglichen.

Werden beispielsweise die *Kombinationen ohne Wiederholung* für eine der Stationen des zuvor genannten Beispiels (3 Stationen mit 12 Schächten) berechnet, so sind nur noch $\binom{36}{12} = 1,25 \cdot 10^9$ Kombinationen für diese Station zu betrachten. Jedoch dürfen die restlichen Stationen nicht außer Acht gelassen werden. Auch diese Stationen können nur mit den zuvor berechneten *Kombinationen ohne Wiederholung* belegt werden. Wird die Filiale-zu-Station-Anordnungen auf der ersten Station fixiert, dann fallen für die anderen Stationen alle Kombinationen weg, in denen zumindest eine Filiale der fixierten Zuordnung der ersten Station vorkommt. Danach wird eine Filiale-zu-Station-Anordnung für die zweite Station fixiert. Diese Schritte werden wiederholt bis nur noch eine Kombination für die allerletzte Station übrig bleibt. Somit wird recht einfach eine erste gültige Lösung zur Bewertung erzeugt. Bei der Berechnung der weiteren Lösungen werden „doppelte“ Lösungen, welche durch Punkt 3 der Aufzählung entstehen, ignoriert. Ein sehr einfaches Beispiel ist unter Abschnitt A.3 zu finden. Es zeigt, wie die Kombinationen ohne Wiederholung zu Lösungen kombiniert werden können. Zusätzlich wird die Formel aus Gleichung 11 angewandt und durch das Beispiel leichter verständlich.

Um alle gewünschten Lösungen bzw. Kombinationen zu erhalten, bietet sich die Rekursion als programmiertechnisches Konzept an. Mittels Rekursion werden einerseits die *Kombinationen ohne Wiederholung* für eine Station und anderseits alle gültigen Lösungen über alle Stationen recht einfach implementieren. Das beschriebene Vorgehen hat eine große Ähnlichkeit mit dem in der Literatur zu findenden Ansatz *Teile und Herrsche* oder in Englisch *Divide & Conquer (D&C)*. Dabei wird das gesamte Problem in unabhängige, kleinere Teilprobleme zerlegt. Die Teilprobleme werden rekursiv oder sobald sie klein genug sind, direkt gelöst. Danach werden die Lösungen der Teilprobleme zur Gesamtlösung zusammengefügt. Bekannte Beispiele für D&C sind Sortieralgorithmen wie *Mergesort* und *Quicksort*.

Folgende Formel berechnet die Anzahl an Lösungen, welche bei dieser konkreten Implementierung der beschränkten Enumeration für dieses *Fallbeispiel* zu berechnen und zu bewerten sind:

[9] Anmerkung: Diese Zuteilung erfolgt bei der Sortierung innerhalb der Station und kann nach jedem Auftragsverteilungs-Algorithmus erfolgen.

$$Anzahl\ Lösungen = \prod_{i=p}^{1} \binom{i \cdot s}{s} \cdot \frac{1}{i} \tag{11}$$

mit $p \ldots$ Anzahl aktiver PKS-Stationen

$s \ldots$ Anzahl der Schächte pro PKS-Station

Zur Entstehung dieser Formel ist zu sagen, dass die Formel der *Kombinationen ohne Wiederholung* mit der Idee zur Zerlegung in kleinere Teilprobleme kombiniert wurde. Für eine nicht fixierte Station werden die möglichen Filiale-zu-Station-Zuordnungen unter Berücksichtigung der anderen, noch nicht fixierten Stationen berechnet. Dieser Wert wird durch die Anzahl der noch nicht fixierten Stationen dividiert, da jede dieser Zuordnungen auf genau eine der nicht fixierten Stationen zugewiesen wird. Auf welche der nicht fixierten Stationen spielt hier keine Rolle. Danach wird eine Station fixiert und dieser Vorgang wird so oft mit einer nicht fixierten Station weniger wiederholt, bis nur noch eine Station übrig bleibt. Das Produkt dieser p Werte ergibt die Anzahl der möglichen Lösungen. Wird diese Formel auf das zuvor genannte Beispiel (3 Stationen mit 12 Schächten) angewandt, so bleiben nur noch $\prod_{i=1}^{3} \binom{i \cdot 12}{12} \cdot \frac{1}{i} = 5,64 \cdot 10^{14}$ Lösungskombinationen übrig.

Dieser Wert ist immer noch viel zu groß, um dieses Beispiel in akzeptabler Zeit zu lösen. Könnten eine Milliarde Lösungen in einer Sekunde berechnen und bewerten werden, so würde die Abarbeitung rund eine Woche benötigen.

Das Wachstum bei ansteigender Problemgröße ist immer noch faktoriell. Bei kleinen Problemen ist die Anzahl an möglichen Lösungen jedoch stark reduziert. Wird beispielsweise ein Problem mit nur einer Station betrachtet, dann gibt es immer nur eine Lösung. Alle anderen Permutationen dieser Lösung sind vom Zielfunktionswert her gleichwertig. Somit können kleine Problemstellungen mit zwei bis drei Stationen und weniger als 10 Schächten pro Station relativ schnell exakt gelöst werden. Die Information, die beste Lösung für ein Problem zu kennen, kann zum Vergleich von Heuristiken und deren Bewertung verwendet werden. Jedoch für größere Problemstellungen, wie sie im Fallbeispiel ohne weiteres vorkommen, ist die beschränkte Enumeration nicht geeignet.

Abschließend wird der rekursive Algorithmus der beschränken Enumeration, wie er im *Fallbeispiel* zur Anwendung kommt, angegeben.

In Algorithmus 5 werden in zwei Schritten allen gültigen Lösungen erzeugt. Diese werden dann sequentiell bewertet und mit der bisher besten gefundenen Lösung verglichen. Am Ende ist die beste Lösung gefunden. Die Bedingung $f \leq s_{max}$, wobei $s_{max} = p \cdot s$ ist, muss für eine gültige Lösung eingehalten werden. Dabei steht f für die Anzahl zu kommissionierender Filialen, p für die Anzahl an Stationen, s für die Anzahl an Schächten je Station und $s_{max} = p \cdot s$ für die maximale Anzahl an Schächten im gesamten PKS Bereich.

Wenn $f < s_{max}$ ist, das heißt, mehr Schächte verfügbar sind als Filialen verteilt werden, werden „leere“ Schächte für die Erzeugung der Kombinationen herangezogen. Das selbe Prinzip wird angewandt, wenn Schächte aus anderen Gründen unbelegt bleiben.

Algorithmus 5 Optimale Lösung durch Beschränkte Enumeration

Require: f Filialen, s_{max} Schächte (p Stationen mit s Schächten)

```
1: Initialisiere x[ ][s]                          ▷ als Feld aus Kombinationen
2: Initialisiere y[ ][p][s]                       ▷ als Feld aus Lösungen
3: x ← ErzeugeKombinationen(f, s)                 ▷ Algorithmus 6
4: y ← ErzeugeLösungen(x)                         ▷ Algorithmus 7
5: b ← y[1]                                       ▷ Variable für beste Lösung
6: for all l aus y do
7:     Bewerte Lösung l
8:     if l besser als b then
9:         b ← l
10:    end if
11: end for
12: Die beste Lösung b ist gefunden
```

In Algorithmus 6 werden alle *Kombinationen ohne Wiederholung* von Filialen für eine Station erzeugt. Hierfür wird die rekursive Definition des Binominialkoeffizienten $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ herangezogen (siehe [22, S. 131 ff]).

Die gesuchten Lösungen werden durch das zuvor beschriebene „Fixieren“ einer Station rekursiv erzeugt. Diese Aufgabe übernimmt Algorithmus 7. Es ist von Vorteil, wenn die zuvor erzeugten Kombinationen in anti-lexikografischer Ordnung (das heißt, in lexikografischer Ordnung von rechts nach links) sortiert sind. Dann kann beim Schritt des Einschränkens der Kombinationen für den nächsten Aufruf der Rekursion ein weiterer Teil der Kombinationen weggelassen werden. Warum dies funktioniert, ist am leichtesten am nachfolgendem Beispiel ersichtlich.

Die Vorgangsweise beim Erzeugen der Lösungen wird in Abbildung 11 grafisch anhand eines kleinen Beispiels demonstriert. Die Ausgangslage ist ein PKS-Bereich mit 4 Stationen. Jede Station verfügt über 2 Schächte. Es sollen nun genau 8 Filialen[10] auf die 8 verfügbaren Schachtplätze verteilt werden. Mit Hilfe der beschränkten Enumeration werden alle Lösungen erzeugt. Durch Gleichung 11 ist nachvollziehbar, dass mit $\binom{8}{2}\frac{1}{4} \cdot \binom{6}{2}\frac{1}{3} \cdot \binom{4}{2}\frac{1}{2} \cdot \binom{2}{2}\frac{1}{1} = 7 \cdot 5 \cdot 3 \cdot 1 = 105$ Lösungen zu rechnen ist. Zuvor werden alle möglichen Kombinationen für eine Station erzeugt und in anti-lexikografischer Ordnung sortiert. Diese Kombinationen sind in der Grafik unter der Überschrift „K“ gelistet. Links davon befinden sich die Lösungen nach ihrer Erzeugung mit Nummer gereiht. Der rechte Teil der Abbildung macht anschaulich, wie die Lösungen erzeugt werden. Es wird zuerst die Kombination „1 2“ für die erste Station fixiert. Danach wird die nächste mögliche Kombination in der nicht Filiale „1“ oder „2“ vorkommt gesucht und mit Kombination „3 4“ gefunden. Danach die Nächste in der nicht „1“, „2“, „3“ oder „4“ vorkommt und mit „5 6“ gefunden. Dieser Schritt wird wiederholt bis eine komplette Lösung gefunden ist. Es muss immer nur von der zuletzt gefundenen Kombination weiter gesucht werden.

[10] Anmerkung: Sollen nur 7 Filialen verteilt werden, muss ein Schacht als leerer Schacht behandelt werden. Bei 6 Filialen sind es 2 leere Schächte. Bei 5 oder 6 Filialen ist es jedoch möglich, mit nur 3 aktiven Stationen zu arbeiten.

Algorithmus 6 Erzeuge alle x Kombinationen ohne Wiederholung für eine Station

Require: f Filialen und s Schächte

```
1:  procedure ErzeugeKombinationen(f, s)
2:      Initialisiere Kombination c[s]                    ▷ als Feld aus s Filialen
3:      Initialisiere Rückgabewert x[ ][s]                ▷ als Feld aus Kombinationen
4:      Initialisiere Zähler i ← 0
5:      procedure ErzeugeKombinationenRek(f, s)           ▷ rekursive Prozedur
6:          if f > s then
7:              ErzeugeKombinationenRek(f − 1, s)
8:          end if
9:          if f ≥ s − 1 then
10:             c[s] ← f                                  ▷ setze Filiale
11:             if s > 1 then
12:                 ErzeugeKombinationenRek(f − 1, s − 1)
13:             else
14:                 i ← i + 1
15:                 x[i] ← c                              ▷ füge neue Kombination hinzu
16:             end if
17:         end if
18:     end procedure
19:     return x
20: end procedure
```

Algorithmus 7 Erzeuge alle y Lösungen der Filiale-zu-Station-Anordnung

Require: $x[\,][s]$ Kombinationen (Feld von Kombinationen $c[s]$)

```
 1: procedure ErzeugeLösungen(x)
 2:     Initialisiere Lösung l[p][s]                    ▷ als Feld von p Kombinationen
 3:     Initialisiere Rückgabewert y[ ][p][s]                ▷ als Feld von Lösungen
 4:     Initialisiere Hilfsvariable z[ ][s]              ▷ als Feld von Kombinationen
 5:     Initialisiere Zähler i ← 0
 6:     Initialisiere Zähler j ← 1
 7:     procedure ErzeugeLösungenRek(x,j)                          ▷ rekursive Prozedur
 8:         if j ≥ p then
 9:             l[j] ← x[1]                              ▷ füge letzte Kombination hinzu
10:             i ← i + 1
11:             y[i] ← l                                      ▷ füge neue Lösung hinzu
12:         else
13:             for all Kombination c[s] aus x do
14:                 l[j] ← c                             ▷ füge eine Kombination hinzu
15:                 z ← jene c aus x, deren aller Filialen nicht in l sind
16:                 ErzeugeLösungenRek(z,j+1)                      ▷ rekursiver Aufruf
17:             end for
18:         end if
19:     end procedure
20:     return y
21: end procedure
```

Zusätzlich ist das rekursive Vorgehen zu erkennen. Es werden zuerst alle 15 möglichen Lösungen, in denen die Kombination „1 2“ vorkommt, erzeugt. Danach werden alle Lösungen mit Kombination „1 3“ usw. erzeugt. Die Zahlen in den gefärbten Quadraten stehen dafür, wie oft eine Kombination bereits in einer Lösung Verwendung gefunden hat. Das vollständige Beispiel ist im Anhang unter Abschnitt A.4 zu finden.

Abschließend ist zu sagen, dass dieser Ansatz nur für die exakte Lösung von sehr kleinen Problemstellungen des *Fallbeispiels* gut geeignet ist.

5.3.3. Branch & Bound und Dynamische Programmierung

Die aus der Informatik stammenden Ansätze des *Branch & Bound (B&B)* sowie des *Dynamic Programming (DP)* sind weitere Möglichkeiten um eine optimale Lösung zu erhalten. Beide gehören thematisch zum Gebiet der partiellen Enumeration und werden im Rahmen dieser Arbeit auf ihre Anwendbarkeit für das *Fallbeispiel* untersucht. Keiner der beiden Ansätze ist als Lösungsverfahren für das *Fallbeispiel* implementiert.

Die Grundidee beim B&B (übersetzt *Verzweigung und Schranke*) ist, nicht alle Lösungsmöglichkeiten abarbeiten zu müssen, sondern schon vorher Zweige des Lösungsbaumes wegzulassen. Die Grundbestandteile sind Initialisierung, Verzweigung und Terminierung. Bei einem Minimierungsproblem (Maximierungsproblem) dient eine obere (untere) Schranke zum Vergleich. Diese bezieht sich auf die bisher beste gefundene Lösung. Zu Beginn wird die obere (untere) Schranke mit einem sehr großen (kleinen) Wert initialisiert. Oft wird dieser Wert auch mit einer „schnellen“ Heuristik festgelegt. Die *Verzweigung* definiert, wie der Lösungsbaum in seine Knoten zerteilt wird und bestimmt im späteren Algorithmus die Auswahl des nächsten Knotens. Dabei können unterschiedlichste Regeln definiert werden. Ist die Abarbeitung des Lösungsbaumes an einem Knotenpunkt K_i angelangt, so muss eine untere (obere) Schranke für die aktuelle (noch nicht vollständige) Lösung L_{K_i} bestimmt werden, indem die restlichen Kind-Knoten heranzogen oder diese vom optimal Zielfunktionswert ableitet werden. Ist die aktuelle untere Schranke von L_{K_i} größer als die obere Schranke, so kann der Zweig von K_i getrost weggelassen werden. Dieser Schritt wird *Terminierung* genannt. Je genauer die *Initialisierung* der oberen Schranke und Berechnung der unteren Schranken erfolgt, desto mehr Zweige können vernachlässigt werden. Jedoch sollen diese Operationen aus Gründen der Rechenzeit nicht zu aufwändig ausfallen. Wird eine bessere Lösung als die bisher beste gefunden, so wird die obere Schranke angepasst. Das B&B-Verfahren kann durch diverse Überlegungen erweitert werden. In der Literatur sind viele verschiedene Ansätze zu finden. Es kann auch mit anderen Verfahren kombiniert werden. Beispielsweise ist *Branch & Cut* die Kombination von B&B und dem Schnittebenenverfahren. Die Grundlagen und weiterführende Informationen zu B&B sind unter den Literaturangaben [26, S. 252 ff], [14, S. 596 ff] und [3] nachzulesen.

Das Abschätzen einer guten unteren Schranke von noch nicht vollständigen Lösungen, stellt das Hauptproblem für eine Realisierung im *Fallbeispiel* dar. Das Anstreben einer möglichst gleichmäßigen Verteilung der Kommissioniermengen auf die Stationen ist problematisch. Auch für die anderen Ziele kann diese Abschätzung nur mit relativ hohem Aufwand erreicht werden. Dies liegt einerseits an der zugrunde liegenden Datenstruktur

L-Nr/S	1	2	3	4
1	01 02	03 04	05 06	07 08
2	01 02	03 04	05 07	06 08
3	01 02	03 04	06 07	05 08
4	01 02	03 05	04 06	07 08
5	01 02	03 05	04 07	06 08
6	01 02	03 05	06 07	04 08
7	01 02	04 05	03 06	07 08
8	01 02	04 05	03 07	06 08
9	01 02	04 05	06 07	03 08
10	01 02	03 06	04 07	05 08
11	01 02	03 06	05 07	04 08
12	01 02	04 06	03 07	05 08
13	01 02	04 06	05 07	03 08
14	01 02	05 06	03 07	04 08
15	01 02	05 06	04 07	03 08
16	01 03	02 04	05 06	07 08
17	01 03	02 04	05 07	06 08
18	01 03	02 04	06 07	05 08
19	01 03	02 05	04 06	07 08
20	01 03	02 05	04 07	06 08
21	01 03	02 05	06 07	04 08
22	01 03	04 05	02 06	07 08
23	01 03	04 05	02 07	06 08
24	01 03	04 05	06 07	02 08
25	01 03	02 06	04 07	05 08
26	01 03	02 06	05 07	04 08
27	01 03	04 06	02 07	05 08
28	01 03	04 06	05 07	02 08
29	01 03	05 06	02 07	04 08
30	01 03	05 06	04 07	02 08
31	02 03	01 04	05 06	07 08
32	02 03	01 04	05 07	06 08
33	02 03	01 04	06 07	05 08
34	02 03	01 05	04 06	07 08
35	02 03	01 05	04 07	06 08
36	02 03	01 05	06 07	04 08
37	02 03	04 05	01 06	07 08

K/L-Nr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
1 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																						
1 3																1	2	3	4	5	6	7	8	9	10	11	12	13	14	15							
2 3																															1	2	3	4	5	6	7
1 4																															1	2	3				
2 4																1	2	3																			
3 4	1	2	3																																		
1 5																																		1	2	3	
2 5																			1	2	3																
3 5				1	2	3																															
4 5							1	2	3													4	5	6													7
1 6																																					1
2 6																						1			2	3											
3 6							1			2	3																										
4 6				1								2	3						4								5	6						7		
5 6	1													2	3	4													5	6	7					
1 7																																					
2 7																							1				2		3								
3 7								1			2		3																								
4 7					1					2				3							4					5				6					7	
5 7		1									2		3				4										5	6			7					
6 7			1			2			3									4				5			6								7			8	
1 8																																					
2 8																								1				2		3							
3 8									1				2		3																						
4 8						1					2			3								4					5			6					7	
5 8			1							2		3						4									5	6				7				
6 8		1			2			3									4				5			6								7			8	
7 8	1			2			3									4			5			6										7			8		9

Legende:

L-Nr ... Lösungsnummer
S ... Station
K ... Kombination

Abbildung 11: Beispiel zum Erzeugen von Lösungen

und andererseits an der definierten Zielfunktion. Da erste Überlegungen und Versuche zu keinem befriedigenden Ergebnis führen, wird der Ansatz für diese Arbeit nicht weiter verfolgt.

Beim DP (Dynamic Programming) wird versucht, ähnlich wie beim D&C (Divide & Conquer), zuerst Teilprobleme zu lösen. Die Teilprobleme sind jedoch voneinander nicht unabhängig, sie werden nur einmal gelöst und daraus resultierende Informationen für die spätere Verwendung abgespeichert. Sind diese Teilprobleme optimal gelöst, werden sie zu einem größeren Problem zusammengesetzt. Dies wird wiederholt, bis das eigentliche Problem gelöst ist. DP folgt dem *Optimalitätsprinzip von Bellman*. Im *Fallbeispiel* verhindert bzw. erschwert die gleichmäßige Verteilung der Kommissioniermengen auf die Stationen den Einsatz von DP. Weitere Informationen zur Dynamischen Programmierung sind unter [26, S. 233 ff], [24, S. 323 ff], [23] und [9] zu finden.

5.4. Heuristische Algorithmen

Mit heuristischen[11] Algorithmen wird ein Kompromiss zwischen Laufzeit und Lösungsgüte eingegangen. In der Regel garantiert ein heuristisches Verfahren keine optimale Lösung. Sie überzeugen meist durch kurze Rechenzeit und geringen Speicherplatzbedarf. Häufig ist ihr Ziel, in möglichst kurzer Zeit, eine möglichst nahe am Optimum liegende Lösung zu finden. Heuristiken sind in der Regel von der Struktur der Problemstellung abhängig und an diese angepasst. Eine einheitliche Definition für heuristische Verfahren konnte in der recherchierten Literatur nicht gefunden werden. Jedoch werden sie recht gut anhand folgender Eigenschaften beschrieben:

- Es wird nicht der gesamte Lösungsraum untersucht.
- Es besteht keine Garantie auf die optimale Lösung.
- Es wird nach Regeln und nicht willkürlich oder zufällig vorgegangen.
- Bei fehlender Konvergenz wird gemäß vorgegebener Stoppregeln abgebrochen.
- Durch einfache Änderung kann das Verfahren an die Problemstruktur angepasst werden.

Grundsätzlich können heuristische Algorithmen nach deren Zielsetzung und Prinzipien gliedert werden. Unterschieden werden *konstruktive* und *verbessernde Verfahren*. Konstruktive Verfahren haben das Ziel, möglichst stabil und zuverlässig eine gültige Lösung zu erreichen. Bei verbessernden Verfahren wird von einer bereits verfügbaren Lösung ausgegangen und versucht, diese in neue, bessere, gültige Lösungen überzuführen. Die wichtigsten Prinzipien sind *Dekomposition*, *Induktives Vorgehen*, *Analogschlüsse*, *Inkrementanalyse*, *Stufenweise Verfeinerung* und *Modellmanipulation*. Die Prinzipien können untereinander kombiniert werden.

[11] Anmerkung: Das Adjektiv heuristisch stammt vom griechischen *heuriskein* = finden, entdecken.

In *Fallbeispiel* finden Heuristiken Verwendung, da die bisher genannten exakten Algorithmen für größere Problemstellungen nicht zufriedenstellend schnell konvergieren. Heuristiken dienen auch dazu, Startlösungen für exakte Verfahren zu generieren, Teilaufgaben von exakten Verfahren abzuarbeiten oder ein besseres Verständnis über das Problem und dessen Lösung zu vermitteln. [26, S. 271 ff]

5.4.1. Rundlaufverfahren

Für diese Heuristik wird die Zielfunktion des *Fallbeispiels* manipuliert. Es wird nur noch die gleichmäßige Verteilung der Kommissioniermengen auf die Stationen verfolgt. Die anderen Ziele werden vorerst vernachlässigt. Der in diesem Abschnitt behandelte heuristische Algorithmus orientiert sich am *Rundlaufprinzip* bzw. engl. *Round-Robin (RR) Principle*. Dieses Prinzip stammt aus dem Bereich des *Scheduling*. Es kann u.a. dafür verwendet werden, eine gleichmäßige *Lastverteilung* (engl. *Load Balancing*) zu erreichen. Beim klassischen Lastverteilungsproblem müssen x Aufgaben unterschiedlichen Umfangs (z.B. Zeitbedarf) auf y Ressourcen verteilt werden. Dabei spielt es keine Rolle, wie viele Aufgaben jede Ressource zugeteilt bekommt. Eine Aufgabe darf immer nur auf einer Ressource bearbeitet werden. Die Arbeitslast (z.B. Arbeitszeit) soll möglichst gleich auf die Ressourcen verteilt werden.

Im *Fallbeispiel* soll zusätzlich dafür gesorgt werden, das auf allen Stationen (Ressourcen) gleich viele Filialen (Aufgaben) kommissioniert werden. Der grundsätzliche Ablauf wird mit dem Ansatz *Längste Operationszeit (LOZ)* bzw. engl. *Longest Processing Time (LPT)* verbunden, um eine akzeptable Heuristik zu erhalten. Diese Heuristik wird im weiteren Verlauf dieser Arbeit *Rundlaufverfahren* genannt. [2] [10] [18] [17]

Beim Rundlaufverfahren werden die f Filialen nach ihren Kommissioniermengen sortiert und in einer Liste verwaltet. Es werden dafür die Bestellmengen aller Auftragszeilen einer Filiale aufsummiert. Das heißt, den ersten Platz der Liste belegt die Filiale mit der größten Bestellmenge. Danach folgt die Filiale mit der zweitgrößten Bestellmenge usw..

Jeder der p aktiven Station wird reihum die jeweils erste Filiale der Liste zugeteilt. Diese Filiale wird, sobald sie einer Station zugeteilt wurde, von der sortierten Liste entfernt. Bei der letzten aktiven Station angelangt, beginnt die Zuteilung der auf der Liste verbliebenen Filialen. Begonnen wird jetzt von der Station, welche die bisher letzte Filiale zugeteilt bekommen hat, das heißt, in rückläufiger Richtung.[12] Dies wird wiederholt, bis die sortierte Liste leer ist. Danach hat jede Station maximal eine Filiale mehr oder weniger zugeteilt als alle anderen. Auch die Kommissioniermengen sind relativ gut auf die Stationen verteilt. Algorithmus 8 zeigt den Ablauf in kompakter Form.

Das Rundlaufverfahren dient hauptsächlich als Eröffnungsverfahren für andere Heuristiken. Eine Laufzeitabschätzung liefert ein Verhalten von $\mathcal{O}(f\ log\ f) + \mathcal{O}(f)$. Der logarithmische Teil steht für den Aufwand des Sortierens der Filialen und der lineare für das Zuweisen der Filialen und den damit verbundenen Operationen.

[12] Anmerkung: Bei drei Stationen „S1“ bis „S3“ und neun nach Bestellmenge durchnummerierten Filialen „F1“ bis „F9“ wurde die Zuteilung wie folgt aussehen: S1-F1, S2-F2, S3-F3, S3-F4, S2-F5, S1-F6, S1-F7, S2-F8, S3-F9.

Algorithmus 8 Rundlaufverfahren mit sortierter Liste

Require: f Filialen mit aufsummierter Kommissioniermenge aller Auftragszeilen, p aktive Stationen

Verwalte aktive Stationen in Feld $a[\,]$ mit Länge p
Sortiere die f Filialen nach ihrer Kommissioniermenge
Verwalte die f Filialen in einer sortierten Liste l mit Länge f
Initialisiere einen Zähler $i \leftarrow 1$
Initialisiere eine Richtung $r \leftarrow +1$
while Liste l nicht leer **do**
    erste Filiale in l wird Station $a[i]$ zugeteilt
    Entferne erste Filiale aus l
    $i \leftarrow i + r$
    if $i < 1$ **then**
        $r \leftarrow +1$
        $i \leftarrow i + r$
    else if $i > p$ **then**
        $r \leftarrow -1$
        $i \leftarrow i + r$
    end if
end while

Es wurde beim Testen des Rundlaufverfahrens festgestellt, dass dieses bei bestimmten Situationen erhebliche Schwächen aufweist. Ist beispielsweise die kumulierte Kommissioniermenge der stärksten Filiale signifikant größer als die aller anderen Filialen, dann kann dies der Algorithmus nur schwer ausgleichen. Ein *verbessertes Rundlaufverfahren* löst diese Situation besser. Dabei werden die auf die Stationen zugewiesenen Kommissioniermengen der Filialen aufsummiert. Wenn eine Filiale auf eine Station zugewiesen werden soll, wird zuvor geprüft, ob die aktuelle kumulierte Kommissioniermenge der Station größer als ein definierter Schwellwert ist. Ist dies der Fall, wird die Filiale vom Ende der Liste gewählt. Diese Filiale hat die kleinste verfügbare Kommissioniermenge in der Liste. Ansonsten wird wie gewohnt die erste Filiale gewählt. Algorithmus 9 zeigt den abgeänderten Ablauf.

Das folgende Beispiel „Rundlaufverfahren“ zeigt den Vorteil des verbesserten Rundlaufverfahren. Angenommen, es sollen die in Tabelle 9 gelisteten Filialen (Nummer 1-9) auf 3 verschiedene Stationen mit je 3 Schächten verteilt werden. Jede Filiale hat eine zugehörige Menge die kommissioniert werden muss. Die Filialen sind bereits nach ihrer Menge absteigend sortiert in der Tabelle geführt. Beim einfachen Rundlaufverfahren (in der Tabelle 9 kurz *RR* genannt) ist die Reihenfolge in der die Filialen gewählt werden von 1 bis 9 durchnummeriert. Außerdem ist ersichtlich, welcher Station jede Filiale zugewiesen wird. Diese Informationen sind für das verbesserte Rundlaufverfahren (kurz *VRR*) dargestellt. Dabei kann erkannt werden, dass sich die Reihenfolge und Stationen für die letzte Hälfte (abgerundet) der Filialen ändert. Die Schwelle ist für dieses Beispiel das Minimum (kurz *Min*) der kumulierten Kommissioniermengen der Stationen

Algorithmus 9 Verbessertes Rundlaufverfahren

Require: f Filialen mit ihrer Bestell- bzw. Kommissioniermenge, p aktive Stationen

Verwalte die p aktive Stationen in Feld $a[p]$
Verwalte die Kommissioniermengen der Stationen in Feld $m[p]$
Initialisiere $m[\,] \leftarrow 0$
Sortiere die f Filialen nach ihrer Kommissioniermenge
Verwalte die Filialen in einer sortierten Liste l mit Länge f
Initialisiere einen Zähler $i \leftarrow 1$
Initialisiere eine Richtung $r \leftarrow +1$
Definiere *Schwelle* ▷ z.B. kleinste Kommissioniermenge in $m[\,]$ mal Faktor
while Liste l nicht leer **do**
    Aktualisiere *Schwelle*
    if $m[i] \leq Schwelle$ **then**
        erste Filiale in l wird Station $a[i]$ zugeteilt
        $m[i] \leftarrow m[i]+$ Kommissioniermenge der ersten Filiale
        Entferne die erste Filiale aus l
    else
        letzte Filiale in l wird Station $a[i]$ zugeteilt
        $m[i] \leftarrow m[i]+$ Kommissioniermenge der letzten Filiale
        Entferne die letzte Filiale aus l
    end if
    $i \leftarrow i + r$
    if $i < 1$ **then**
        $r \leftarrow +1$
        $i \leftarrow i + r$
    else if $i > p$ **then**
        $r \leftarrow -1$
        $i \leftarrow i + r$
    end if
end while

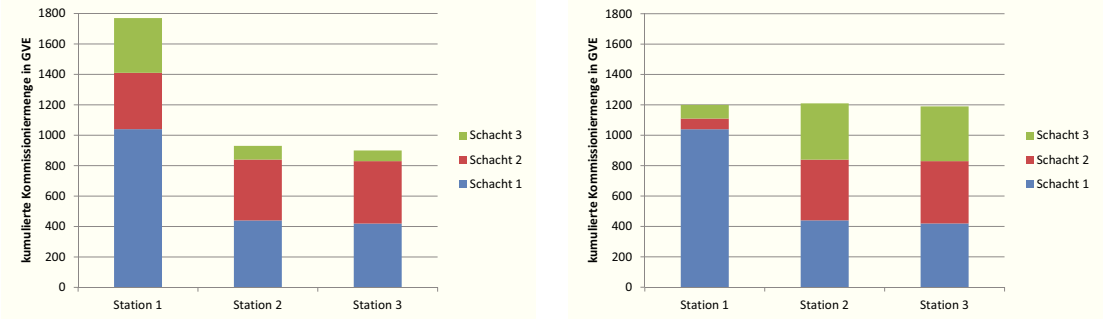
mal einem Faktor von 1,2. Für die Berechnung der Schwelle werden immer nur die schon zugewiesenen Filialen betrachtet und somit muss die Schwelle nach jeder Zuweisung neu berechnet werden. Die Prüfung kontrolliert, ob die kumulierte Kommissioniermenge der aktuellen Station kleiner gleich oder größer der Schwelle ist. Ist sie größer, so wird die Filiale vom Ende der Liste zugewiesen (siehe Filiale 9 und 8 bei VRR). Dieser Umstand verändert die Reihenfolge.

Tabelle 9: Tabelle zum Beispiel Rundlaufverfahren

Filiale		Reihenfolge		Station		Schwelle	
Nr.	Menge	RR	VRR	RR	VRR	Min	Prüfung
1	1040	1	1	1	1	0	$0 \leq 0$
2	440	2	2	2	2	0	$0 \leq 0$
3	420	3	3	3	3	0	$0 \leq 0$
4	410	4	4	3	3	420	$420 \leq 504$
5	400	5	5	2	2	440	$440 \leq 528$
6	370	6	8	1	2	830	$840 \leq 996$
7	360	7	9	1	3	830	$830 \leq 996$
8	90	8	7	2	1	830	$1110 > 996$
9	70	9	6	3	1	830	$1040 > 996$

Abbildung 12 zeigt die Verteilung der Filialen bzw. die kumulierten Kommissioniermengen des Beispiels in Form von Balkendiagrammen. Das verbesserte Rundlaufverfahren schneidet im Vergleich besser ab. Es ist speziell für solche Fälle konzipiert. Das Einstellen des Faktors für die Schwellenberechnung ist dabei entscheidend und konkret von der Problemstellung abhängig. Wird der Faktor für dieses Beispiel von 1,2 auf 1 verändert, so würden die Stationen exakt gleich hohe Kommissioniermengen zugewiesen bekommen.[13]

Das Hauptproblem bei der Anwendung der Rundlaufverfahren im *Fallbeispiel* ist, dass nur die Verteilung der Kommissioniermengen und keines der anderen Ziele verfolgt wird.



(a) Rundlaufverfahren (b) verbessertes Rundlaufverfahren

Abbildung 12: Diagramme zum Beispiel Rundlaufverfahren

[13] Anmerkung: Filiale 6 und 7 würden ihre Station und Reihenfolge für VRR tauschen.

5.4.2. Lokale Optimierung

Es wurde hier auf die Idee der *lokalen Suche* zurückgegriffen, wie sie in der Literatur unter [14, S. 580 ff] zu finden ist. Es handelt sich dabei um eine verbessernde Heuristik.

Einer der bekanntesten Vertreter der lokalen Suche ist die *k-Opt-Heuristik* zur Lösung des Traveling Salesman Problem (TSP). Bei TSP muss ein Handelsreisender n Städte in Form einer Rundreise besuchen. Die Wegstrecke, welche der Handelsreisende dabei zurücklegen muss, soll minimal sein. In einem Graphen werden die Städte als Knoten und die Wegstrecken zwischen den Städten als Kanten modelliert. Die *k-Opt-Heuristik*, ermittelt in jeder Iteration die bestmögliche Rundreise in der k-Nachbarschaft einer zuvor gefundenen Rundreise. Das heißt, es werden alle Möglichkeiten k Kanten dieser Rundreise zu Tauschen betrachtet und es wird mit der besten neuen Rundreise fortgesetzt. Wird das Vertauschen solange wiederholt, bis keine Verbesserung mehr erreicht wird, konvergiert der Algorithmus zu einem lokalen Optimum der Nachbarschaft. Das heißt, durch das Vertauschen von zwei Kanten kann dieses nicht mehr verlassen werden.

Die *2-Opt-Heuristik* des TSP wurde für das *Fallbeispiel* angepasst. Einfach gesagt, werden anstatt zweier Kanten einer Rundreise, zwei Filialen einer bereits gültigen Filiale-zu-Schacht-Anordnung zwischen zwei Stationen getauscht. Als lokale Änderung ist somit das Tauschen zweier Filialen verschiedener Stationen festgelegt. Dieser Schritt wird im *Fallbeispiel* als „Vertauschen“ bezeichnet. Er definiert die Nachbarschaft der lokalen Suche. Änderungen werden sofort übernommen, sobald die neue Lösung eine Verbesserung der Zielfunktion darstellt. Das heißt, es wird nicht die komplette Nachbarschaft durchsucht, sondern sofort mit der besseren Lösung von neuem begonnen. Sobald durch diese endliche Verkettung von 2-Vertauschungen keine Verbesserung mehr erreicht wird, stoppt die Suche in einem lokalem Optimum. Der Ablauf ist in Algorithmus 10 abgebildet.

Algorithmus 10 Lokale Optimierung - Vertauschen

Require: Startlösung, Zielfunktion $f(Lösung)$
1: Lösung $b \leftarrow$ Startlösung
2: **while** Verbesserung **do**
3: **for all** Paare: Filiale - Filiale (verschiedener Stationen) **do**
4: Lösung $n \leftarrow$ Vertausche die zwei Filialen
5: **if** $f(n) < f(b)$ **then** ▷ bei Minimierungsproblem
6: $b \leftarrow n$ ▷ eine Verbesserung
7: **end if**
8: **end for**
9: **end while**

Als Alternative zu Algorithmus 10 kann zuerst die gesamte Nachbarschaft durchsucht werden, bevor die aktuelle Lösung (bei der ersten Iteration ist die aktuelle Lösung die Startlösung) abgeändert wird. Ist die beste durch „Vertauschen“ entstandene Lösung der Nachbarschaft besser als die aktuelle Lösung, so wird diese als neue aktuelle Lösung für die nächste Iteration genommen. Ansonsten wird der Algorithmus beendet.

Dieses Vorgehen liefert gute Ergebnisse, wenn alle Schächte mit Filialen belegt sind. Bleiben Schächte unbelegt, so kann der Algorithmus nicht den gewünschten Erfolg erzielen. Um bessere Ergebnisse zu ermöglichen, müssen mit Filialen belegte Schächte zusätzlich mit leeren Schächten anderer Stationen getauscht werden können. Somit wird der Algorithmus in seiner verbesserten Version, um eine zweite lokale Änderung, welche in dieser Arbeit „Verschieben“ genannt wird, erweitert. Diese Änderung ist in Algorithmus 11 zu sehen.

Algorithmus 11 Lokale Optimierung - Verschieben und Vertauschen

```
Require: Startlösung, Zielfunktion f(Lösung)
 1: Lösung b ← Startlösung
 2: while Verbesserung do
 3:     for all Paare: Filiale - unbelegter Schacht (verschiedener Stationen) do
 4:         Lösung n ← Verschiebe Filiale auf andere Station
 5:         if f(n) < f(b) then                          ▷ bei Minimierungsproblem
 6:             b ← n                                          ▷ eine Verbesserung
 7:         end if
 8:     end for
 9:     for all Paare: Filiale - Filiale (verschiedener Stationen) do
10:         Lösung n ← Vertausche die zwei Filialen
11:         if f(n) < f(b) then                          ▷ bei Minimierungsproblem
12:             b ← n                                          ▷ eine Verbesserung
13:         end if
14:     end for
15: end while
```

Die gezeigten Algorithmen sind in vereinfachter Form niedergeschrieben. Es sind aus Gründen der Übersichtlichkeit, nicht alle Abfragen und Prüfungen inkludiert. Beispielweise ist es nur sinnvoll Algorithmus 11 auszuführen, wenn mehr als eine Station aktiv ist. Es wurden insgesamt fünf verschiedene Verfahren der lokalen Optimierung nach Vorbild der *2-Opt-Heuristik* für das *Fallbeispiel* implementiert, getestet und bewertet. Die Ergebnisse dazu sind unter Abschnitt 5.7.4 zu finden.

Eine Erweiterung der Nachbarschaft nach dem Vorbild der *k-Opt-Heuristik* mit $k > 2$ wäre durchaus denkbar. Diese wurde jedoch im *Fallbeispiel* nicht konkret umgesetzt. Auch die Ansätze des *Tabu Search*, könnten mit dem beschriebenen Vorgehen gut kombiniert werden, um lokale Minima wieder zu verlassen und unerforschte Lösungsräume zu durchforsten.

Die Algorithmen der lokalen Suche sind für das *Fallbeispiel* aus zweierlei Gründen von großer Bedeutung. Sie konvergieren für übliche Größen der Problemstellen relativ schnell und liefern außerdem für die Praxis ausreichend gute Ergebnisse. Es werden alle Zielgrößen der Problemstellung mit einbezogen. Zum Erzeugen der Startlösung werden die schnellen Rundlaufverfahren herangezogen.

Zusätzlich kann der Ablauf der lokalen Suche für eine Ausprägung der Problemstellung mit unterschiedlichen Startlösungen wiederholt ausgeführt werden. Dadurch kann ein

größerer Bereich des Lösungsraumes abgesucht werden. Abschließend wird die beste gefundene Lösung gewählt.

5.5. Genetische Algorithmen

Genetische Algorithmen (GA) versuchen, den Selektionsprozess der Natur nachzubilden. In der Natur überleben die Individuen, welche sich durch Kreuzung und Mutation von den anderen positiv abheben und im weiteren Verlauf durchsetzen können. So wie sich die Individuen in der Natur den Umweltgegebenheiten anpassen müssen, müssen sich die Individuen in einem genetischen Algorithmus den Vorgaben der Problemstellung anpassen. Ein Individuum wird im genetischen Algorithmus durch eine gültige Lösung dargestellt. Diese Lösung und somit das Individuum kann mit der Zielfunktion bewertet werden. Gewisse Ausprägungen, Merkmale oder Muster innerhalb der Lösung werden als *„Gen“* bezeichnet und durch Kreuzung[14] von den Vorfahren an die Nachkommen weitergegeben. Bei genetischen Algorithmen wird parallel mit mehreren Individuen gearbeitet. Alle aktiven bzw. „lebenden“ Individuen stellen die Population dar. Die Individuen haben nur bei gewissen Varianten von GAs eine bestimmte Lebensdauer. Die Fitnessfunktion bewertet ein Individuum innerhalb einer Population. Damit wird die Qualität bzw. der Grad der Zielerfüllung des Individuums ermittelt. Das heißt, dass alle Individuen einer Population über diese Fitnessfunktion eingestuft werden können. Durch sie werden Selektion und Kreuzung gesteuert. Individuen mit besserer Fitness werden mit größerer Wahrscheinlichkeit zur „Fortpflanzung“ auserkoren. Mittels Kreuzung und Mutation werden neue Individuen erzeugt, welche eine neue Generation darstellen. Der Übergang von einer alten in eine neue Population wird durch den Algorithmus nachgebildet, indem alte Individuen „sterben“ und die der neuen Generation hinzukommen. Bei zu hohem Selektionsdruck setzen sich gute Individuen zu früh durch und viele mögliche Lösungen werden gar nicht betrachtet. Bei zu geringem Druck konvergiert der Algorithmus sehr langsam oder gar nicht mehr. Die Diversität innerhalb einer Population kann darüber Aufschluss geben, ab wann der Algorithmus von Neuem gestartet werden soll.

Es gibt eine Vielzahl von Möglichkeiten, genetische Algorithmen zu gestalten und zu erweitern. Der prinzipielle Ablauf und dessen Grundbestandteile sind jedoch meist die selben. Zu den Grundbestandteilen eines genetischen Algorithmus gehören:

Initialisierung: Durch die Initialisierung wird die allererste Population festgelegt. Diese Population kann aus zufällig generierten Individuen bestehen oder aus bereits bekannten Lösungen erzeugt werden.

Fitness: Durch die Fitnessfunktion kann jedem Individuum einer Population ein Wert zugewiesen werden. Die Fitness eines Individuums bestimmt bei der Selektion, wie gut das Individuum im Vergleich zu den anderen ist.

Selektion: Durch die Selektion wird entschieden, welche Individuen der Population für die Kreuzung herangezogen werden. Die Auswahl erfolgt i.d.R über die Fitness

[14] Anmerkung: Die Kreuzung (engl. *Crossover*) wird im Kontext der GA oft als Rekombination bezeichnet.

der Individuen. Die Bestimmung der Fitness kann proportional oder rangbasiert erfolgen und bestimmt u.a. den Selektionsdruck. Die Auswahl kann durch diverse Verfahren wie beispielsweise die Roulette-, Elite- oder Turnierselektion erfolgen.

Kreuzung: Durch die Kreuzung werden aus selektierten Eltern-Individuen neue Kind-Individuen erzeugt. Die Struktur der Individuen (Lösungen der Problemstellung) bestimmt die konkrete Umsetzung dieses Schrittes. Es wird i.d.R versucht *„Gene“* der Eltern-Individuen an die Kind-Individuen so weiterzugeben, dass gute Eigenschaft erhalten bleiben.

Mutation: Durch Mutation wird versucht, ein gewisses Maß an künstlicher Diversität innerhalb der neuen Individuen-Population zu erzeugen. Diese zufälligen, kleinen Abänderungen der Gene erfolgen relativ selten und stellen neues Genmaterial bereit. Das Verfahren der Mutation ist von seiner Art her an die Problemstellung angepasst.

Wiedereinfügen: Durch das Wiedereinfügen wird bestimmt, wie der Übergang von der alten in die neue Population erfolgt. Der einfachste Fall ist, dass alle Kinder-Individuen die neue Population darstellen und die alte Generation ersetzen. Andere Vorgangsweisen ermöglichen beispielsweise, dass einige Eltern-Individuen beibehalten werden, oder dass die Anzahl an Individuen unterschiedlicher Generationen[15] unterschiedlich sein können. [16]

Terminierung: Der genetische Algorithmus wird nach Erreichen eines Abbruchkriteriums beendet. Hierfür kann beispielsweise eine zuvor definierte Rechenzeit, eine bestimmte Anzahl an Iterationen (neuer Generationen), das Fehlen von Verbesserung oder die Diversität in der aktuellen Population herangezogen werden.

Wie die einzelnen Teile konkret implementiert und zu einem Ganzen zusammengesetzt werden, ist hauptsächlich von der Problemstellung abhängig. In jedem Fall muss eine Vielzahl von Parametern eingestellt werden. Dazu gehören beispielsweise die Größe der Population, Mutationsrate und Anzahl an Iterationen. Für einige sind empfohlene Wertebereiche oder Faustregeln in der Literatur zu finden. Die Feineinstellung dieser Parameter ist für jedes Problem neu vorzunehmen.

Ein einfacher genetischer Algorithmus könnte wie folgt aussehen (in Anlehnung an die Fachliteratur [16] und [20]):

Dabei steht $P(i)$ für die aktuelle Population der Iteration i. Die Parameter sind für dieses einfache Beispiel die Populationsgröße, Mutationsrate und maximale Anzahl an Iterationen. Die Population hat immer eine bestimmte Größe (z.B. 100 Individuen).

[15] Anmerkung: Eine Generation sind alle Kinder-Individuen die aus der Kreuzung einer Population entstehen.

[16] Anmerkung: Die GA und Evolutionsstrategien (ES) zählen beide zu den evolutionäre Algorithmen (EA) und können heutzutage in einigen Bereichen nur mehr schwer voneinander getrennt werden. Beispielsweise sind die (μ,λ) und (μ+λ)-Selektion auch bei modernen GA im Einsatz. Siehe dazu unter [1, S. 3].

Algorithmus 12 Genetischer Algorithmus

Require: Zielfunktion $f(Lösung/Individuum)$, Angabe der Parameter

1: Iterationszähler $i \leftarrow 0$
2: Population $P(i) \leftarrow$ Erzeuge Individuen (zufällig) ▷ Anfangspopulation
3: Bewerte Fitness von $P(i)$ ▷ mittels $f(Individuum)$
4: **repeat**
5: $i \leftarrow i + 1$
6: Eltern $E(i) \leftarrow$ Selektiere Individuen von $P(i-1)$
7: Kinder $K(i) \leftarrow$ Kreuze Individuen von $E(i)$
8: $P(i) \leftarrow$ Mutiere Individuen von $K(i)$ ▷ neue Population
9: Bewerte Fitness von $P(i)$
10: **until** Abbruchbedingung

Diese werden in der Initialisierungsphase zufällig erzeugt und über die Fitnessfunktion bewertet. $P(0)$ stellt die Anfangspopulation dar. Bis die Abbruchbedingung erreicht ist, werden pro Iteration neue Populationen von Individuen durch Selektion, Kreuzung und Mutation erzeugt. Die Kinder stellen die neue Population dar und ersetzen die alte. Dabei werden in jeder Iteration genau so viele Kinder durch Kreuzung erzeugt, dass die Populationsgröße immer konstant bleibt.

Für die Ausarbeitung dieses Abschnittes diente die Fachliteratur, welche unter den Verweisen [1], [8], [16], [19], [20] und [21] gefunden werden kann.

Für das *Fallbeispiel* wurde dieser einfache Ansatz erweitert und problemspezifisch umgesetzt. In der erweiterten Version kann das Verhalten des Algorithmus und das der Erweiterungen durch Parameterwahl gesteuert werden. Es werden nun alle Erweiterungen und spezifischen Umsetzungsmerkmale im Detail beschrieben.

Die Gene eines Individuum sind die Filialen auf den Stationen. Diese Zuordnung wird für die Startpopulation zufällig generiert. Die Größe der Population ist ein Parameter und bleibt während des Ablaufs konstant. Die Reihung der Individuen nach ihrer Fitness kann wahlweise proportional oder rangbasiert erfolgen. Mittels Roulette-Selektion werden die Eltern für die Kreuzung bestimmt. Ein Individuum darf sich nicht mit sich selbst kreuzen. Bei jeder Kreuzung werden standardmäßig aus zwei Eltern vier Kinder erzeugt. Dabei werden die Eltern als Vater und Mutter unterschieden. Zwei der Kinder erhalten prinzipiell eine zufällige Gen-Sequenz der Vaters unverändert vererbt. Gene die Mutter- und Vater-Individuen gemeinsam teilen, werden auch weitervererbt. Sollte noch zusätzliches Gen-Material für eine gültige Lösung notwendig sein, wird es zufällig gewählt. Selbiges gilt für die anderen beiden Kinder, nur dass diese jeweils eine zufällige Sequenzen der Mutter unverändert vererbt bekommen. Das Vorgehen bei der Kreuzung ist in Abbildung 13 grafisch dargestellt. Bei dieser Darstellung wird nicht eine zufällige Gen-Sequenz gewählt, sondern immer eine Hälfte der Gen-Sequenz eines der Elternteile (vier verschiedenen Hälften entsprechen den vier Nachkommen) an die Kinder direkt vererbt.

Danach werden alle Kinder einer zufälligen Mutation unterzogen. Mit dem Parameter $\lambda_{\%}$ kann angeben werden wie groß der Anteil an Kindern in der neuen Population ausfal-

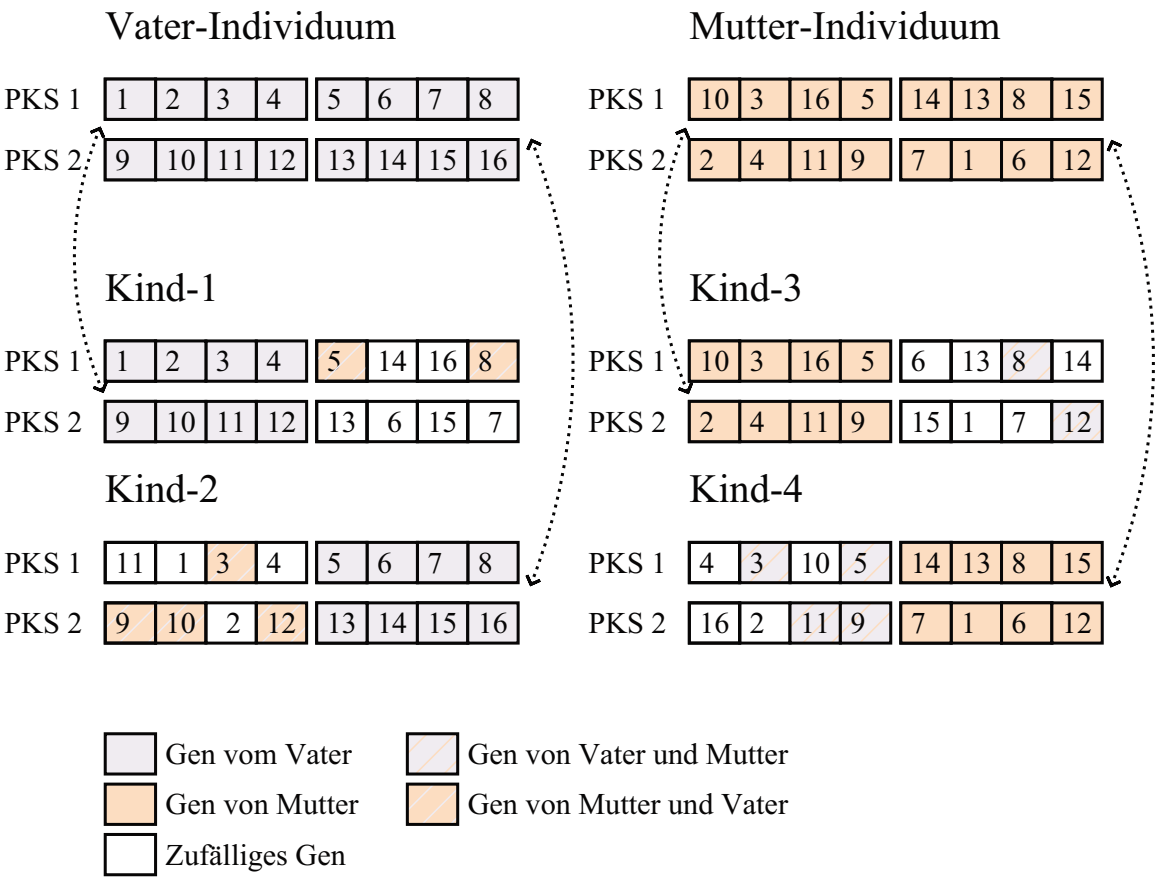


Abbildung 13: Kreuzung

len soll. Wird dieser mit 100% gewählt, so entspricht dies einer vollständigen Ersetzung (wie in Algorithmus 12). Somit kann das Verhältnis von μ Individuen der Elterngeneration zu λ Individuen der Kindergeneration für die neue Population eingestellt werden. Es werden jeweils die besten Eltern und Kinder gewählt und somit ein elitärer Ansatz verfolgt. Der Abbruch der Optimierung erfolgt entweder nach einer definierten Zeit, Anzahl an absolvierten Iterationen (neuen Populationen), oder einer Kombination von beidem. Zusätzlich wird über die Zielfunktionswerte der einzelnen Individuen in der aktuellen Population eine Diversität berechnet. Unterschreitet die Vielfältigkeit einen einstellbaren Wert, so wird ein „Neustart“ durchgeführt. Das heißt, es wird eine neue Startpopulation zufällig erzeugt und mit dieser fortgesetzt. Die beste gefundene Lösung wird separat gespeichert und aktualisiert, damit kein Fortschritt verloren geht.

Algorithmus 13 zeigt den angepassten Ablauf für das *Fallbeispiel*. In dieser Version wurde er schlussendlich implementiert und getestet. Bei den GA gibt es viele verschiedene Varianten und Erweiterungen. Einige davon wurden während der Entwicklung ausprobiert und wieder verworfen und nicht erfolgversprechende wurden erst gar nicht weiter untersucht. Beispielsweise setzte sich in den Tests die rangbasierte Bestimmung der Fitness gegenüber der fitnessproportionalen durch. Es wurden jedoch beide Verfahren implementiert. Bei der Kreuzung wurden einige Varianten untersucht, erweitert und teilweise auch wieder verworfen. Schlussendlich wurde die bereits beschriebene Variante mit den vier Nachkommen gewählt. Es besteht die Möglichkeit einen fixen Prozentsatz der besten (Eltern-)Individuen der alten Population mit in die neue aufzunehmen. Die Größe der Population ändert sich jedoch nicht zwischen den Generationen. Ein kritischer Faktor ist eine passende Parameterwahl, um mit der Suche den Lösungsraum möglichst gut abzudecken und eine Lösung nahe am Optimum zu finden.

Algorithmus 13 Genetischer Algorithmus des *Fallbeispiels*

Require: Zielfunktion $f(Lösung/Individuum)$, Angabe der Parameter

Individuum b ▷ bestes gefundene Individuum
Iterationszähler $i \leftarrow 0$
Population $P(i) \leftarrow$ Erzeuge Individuen (zufällig) ▷ Anfangspopulation
Bewerte Fitness von $P(i)$ ▷ proportional oder rangbasiert
repeat
    $i \leftarrow i + 1$
    if Diversität zu gering **then** ▷ Neustart
        Population $P(i) \leftarrow$ Erzeuge Individuen (zufällig)
        Bewerte Fitness von $P(i)$
    end if
    Initialisiere Kinder $K(i)$
    repeat
        Vater $V \leftarrow$ Roulette-selektiere ein Individuum aus $P(i-1)$
        Mutter $M \leftarrow$ Roulette-selektiere ein Individuum aus $P(i-1)$
        Kinder $K(i) \overset{\text{add}}{\leftarrow}$ Kreuze(V, M) ▷ erzeuge 4 Kinder und füge sie hinzu
    until genug Kinder
    $G(i) \leftarrow$ Mutiere Individuen von $K(i)$ ▷ neue Generation
    Bewerte Fitness von $G(i)$
    if $\lambda_{\%} \geq 100$ **then**
        $P(i) \leftarrow G(i)$
    else
        $P(i) \overset{\text{add}}{\leftarrow}$ besten $\frac{\lambda_{\%} \cdot \text{Populationsgröße}}{100}$ Individuen von $G(i)$
        $P(i) \overset{\text{add}}{\leftarrow}$ besten $\frac{(100-\lambda_{\%}) \cdot \text{Populationsgröße}}{100}$ Individuen von $P(i-1)$
    end if
    if b schlechter als bestes Individuum aus $P(i)$ **then**
        $b \leftarrow$ bestes Individuum aus $P(i)$
    end if
until $i \geq$ Maximale Anzahl Iterationen $\vee$ Maximale Rechenzeit erreicht

5.6. Sortierung innerhalb der Station

Die endgültige Schachtbelegung der Filialen innerhalb der Stationen ist als separater Schritt umgesetzt. Sie kann im Anschluss an alle zuvor vorgestellten Verfahren (wie beschränkte Enumeration, Rundlaufverfahren, lokale Optimierung oder genetischer Algorithmus) durchgeführt werden. In die Zielfunktion der Teilsystemoptimierung fließt die Position einer Filiale innerhalb einer Station nicht ein und somit wird diese bei den zuvor vorgestellten Verfahren nicht gewertet. Nachdem die Filiale-zu-Station-Zuordnung festgelegt wurde, wird innerhalb jeder Station noch einmal optimiert und die Filiale-zu-Schacht-Zuordnung festgelegt.

Als Ziel wird definiert, möglichst alle Filialen einer Tour in benachbarten Schächten einer Seite der Station für die Kommissionierung zu platzieren. Dabei wird eine PKS

Station in zwei Seiten (links bzw. rechts) unterteilt. Die Schächte sind gleichmäßig auf die beiden Seiten verteilt (z.B. bei 12 Schächten befinden sich auf jeder Seite jeweils 6 Schächte). Dies soll den Abtransport, der fertig kommissionierten Rollcontainer, für die Mitarbeiter erleichtern, da diese die Rollcontainer nach Touren getrennt für das FTF bereitstellen müssen. Die Position der Filialen innerhalb der Station ist also nicht für die Auftragsverteilung oder Kommissionierung, sondern für die nachfolgenden Prozesse relevant. Durch Auftragszeilenverteilung zwischen TKS und PKS und Auftragsverteilung im PKS wird sichergestellt, dass nie mehr Filialen einer Station zugewiesen werden, als Schächte auf der Station verfügbar sind.

Um eine möglichst gute Filiale-zu-Schacht-Zuordnung innerhalb einer Station zu finden, wurde folgender Ablauf entwickelt, der für jede aktive PKS Station einzeln abgearbeitet wird:

1. Sortiere die Touren (einer Station) absteigend nach ihrer Filialen-Anzahl (auf dieser Station).
2. Verwalte die Touren in einer sortierten Liste.
3. Weise den Seiten (l . . . links, r . . . rechts) alternierend nach Rundlaufverfahren (das heißt, l-r-r-l-l-r-r-l usw.) die Filialen der ersten Tour der Liste zu und lösche diese Tour danach aus der Liste.
4. Sollte auf der einen Seite kein freier Schacht mehr zur Verfügung stehen, muss die Filiale bzw. müssen die Filialen auf die andere Seite platziert werden.
5. Wiederhole Punkt 3 und Punkt 4 solange bis die Liste leer geworden ist.

Folgendes Beispiel soll den Ablauf veranschaulichen. Es sollen zwölf Filialen auf einer PKS Station mit zwölf Schächten verteilt werden. Die zwölf Filialen gehören zu fünf verschiedenen Touren. Jede Tour besitzt ein bestimmte Anzahl an zugehörigen Filialen. Die Touren sind in Tabelle 10 nach ihrer Anzahl an Filialen aufsteigend gelistet. Es sind zusätzlich die nach dem Rundlaufprinzip ermittelten Seiten und die nach dem Algorithmus tatsächlich zugeteilten Seiten angeführt.

Tabelle 10: Beispiel Sortierung innerhalb der PKS-Station

Tour	**Anzahl Filialen**	**Seite nach RR**	**Seite zugeteilt**
2	4	links	links
1	3	rechts	rechts
3	2	rechts	rechts
4	2	links	links
5	1	links	rechts

Abbildung 14 zeigt die Platzierung der Filialen anhand des zuvor beschriebenen Vorgehens. In der Abbildung wird nur die Tournummer, zu welcher die Filiale gehört dargestellt. Die Filialen werden nicht näher identifiziert und im Schacht zu dem die Filiale

zugeteilt wird, steht daher nur die Nummer der Tour. Die eine Filiale der Tour Nummer 5 sollte eigentlich auf der linken Seite der Station platziert werden. Da jedoch dort kein freier Schacht mehr verfügbar ist, wird die Filiale auf die rechte Seite der Station verlegt. Die Filialen können zusätzlich nach einem definierten Kriterium (z.B. GVE-Kommissioniermenge) innerhalb der Tour sortiert und dementsprechend platziert werden. Dies würde es beispielsweise ermöglichen, dass Filialen, für welche viele Rollcontainer kommissioniert werden, näher zu den Rollcontainer-Sammelplätzen positioniert werden.

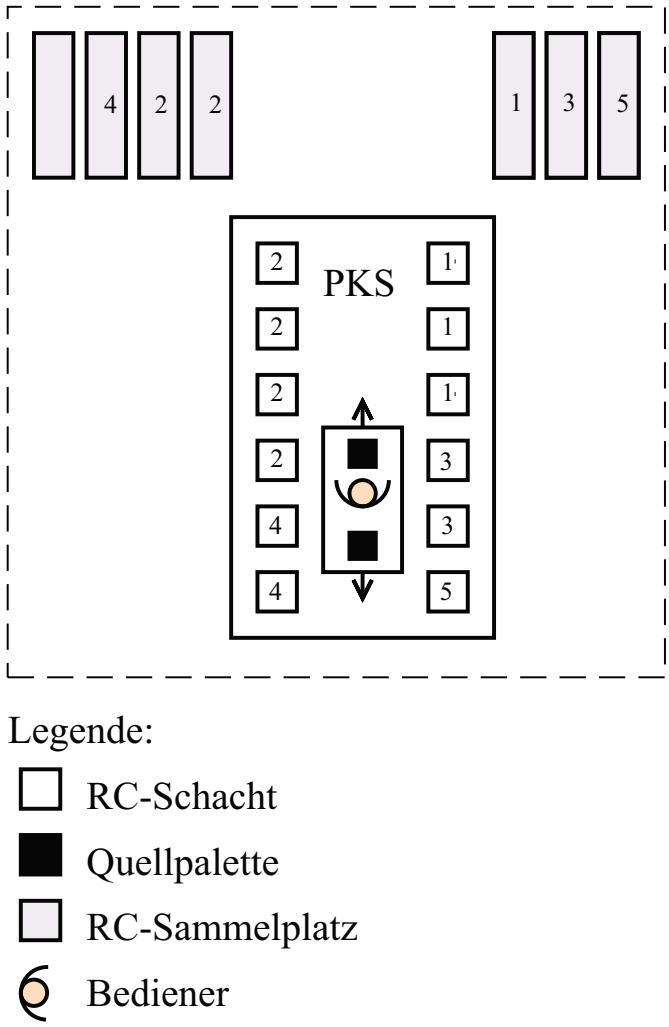


Abbildung 14: Beispiel Sortierung innerhalb der PKS-Station

Der Ablauf ermöglicht es, Filialen einer Tour innerhalb einer Station zusammenzuhalten, solange es von den Schächten her möglich ist. Hat eine Tour mehr Filialen als die Hälfte der verfügbaren Schächte oder gibt es z.B. drei Touren mit jeweils vier Filialen bei zwölf Schächten, so müssen in jedem Fall die Filialen einer Tour auf beiden Seiten platziert werden.

5.7. Bewertung der realisierten Algorithmen

Die Bewertung der realisierten Algorithmen erfolgt anhand der *Testdaten* des *Fallbeispiels*.

Zur Bewertung der Algorithmen werden der benötigte Rechenaufwand und die gefundene Lösungsgüte herangezogen. Beide Bereiche werden mit Werten aus den Berechnungen der Testdaten hinterlegt. Es werden keine Laufzeitabschätzungen durchgeführt. Es wird die durchschnittlich benötigte Rechenzeit für die Berechnung der Testdaten zum Vergleich der Algorithmen herangezogen. Die Lösungsgüte wird auf Grund der erreichten Werte der Zielfunktion der Lösung ermittelt. Es wird jeweils der erreichte Wert mit dem Wert des Optimums oder, falls dieses aus zeitlichen Gründen nicht berechnet werden konnte, der besten gefundenen Lösung verglichen.

Zur Kennzahl der Rechenzeit ist zu sagen, dass die Berechnungen auf einem Laptop *Dell Latitude* E6520 (mit folgenden Leistungseigenschaften: *Intel ® Core™* i5-2410M mit 2,3 GHz und 4 GB RAM) durchgeführt wurden.

Weitere Bewertungsmöglichkeiten wie Grad der Allgemeinheit und Benutzerfreundlichkeit der Algorithmen werden nicht untersucht.

5.7.1. Vergleich mit der exakten Lösung

Zuerst wird eine aussagekräftige Kommissionierwelle des Bereichs UKB des Durchschnittstages herangezogen, um mit allen umgesetzten Verfahren gelöst zu werden. Diese Kommissionierwelle wird mit zwei PKS Stationen mit jeweils acht Schächten und vier TKS Stationen kommissioniert. Somit ist die Problemstellung ausreichend klein, dass sie auch exakt gelöst werden kann. Die Auftragszeilenverteilung liefert die in Tabelle 11 zu sehenden Kennzahlen.

Tabelle 11: Ergebnisse nach Auftragszeilenverteilung

Kennzahl	**Wert**
Abweichung GVE-Ziel	0,232%
Abweichung Wellendauer	-0,143%
SKU	139
Filialen	16
Rollcontainer	130
Füllgrad	80,18%
GVE/RC	51,786
GVE/SKU	21,781
SKU/h	25,309

Im Anschluss werden alle implementierten Algorithmen für die Teilsystemoptimierung auf diese Ausprägung der Problemstellung angewandt. Die Ergebnisse sind in Tabelle 12 zu sehen. Der Zielfunktionswert (ZFW) der besten gefundenen Lösungen jedes Verfahrens wird absolut und in Prozent zur optimalen Lösung angegeben. Neben der Rechenzeit (RZ) in Sekunden werden auch noch die durchschnittlichen GVE pro Palette (GVE/Pal)

und die SKU pro Station angeben. Je mehr GVE durchschnittlich von einer Palette bei der Kommissionierung entnommen werden, desto besser ist das für die Kommissionierung im PKS. Sinkt der Wert der SKU pro Station, so müssen durchschnittlich weniger Palettenwechsel zwischen den Stationen durchgeführt werden, da einige Artikel nur auf einer PKS Station kommissioniert werden.

Tabelle 12: Ergebnisse des Vergleichs der Algorithmen

Verfahren	**ZFW absolut**	**Abw. ZFW in %**	**RZ [s]**	**SKU pro Station**	**GVE/Pal**
Exakter Algorithmus	27104,9	0,00%	2837,0	135,00	21,94
Rundlaufverfahren (RR)	47092,1	73,74%	0,218	138,50	21,47
verbessertes RR (RR imp)	32104,9	18,45%	0,234	138,50	21,62
Lokale Optimierung (LO)	27422,5	1,17%	6,574	136,50	21,79
Genetischer Alg. 10 sec	27580,4	1,75%	13,615	137,00	21,70
Genetischer Alg. 30 sec	27510,8	1,50%	32,693	136,83	21,70
Genetischer Alg. 60 sec	27104,9	0,00%	65,809	135,00	21,94

Auffällig gute Ergebnisse erreichen die lokale Optimierung und der genetische Algorithmus. Dabei ist anzumerken, dass die lokale Optimierung (Suche) nur mit einer einzigen Startlösung durchgeführt wurde. Der genetische Algorithmus wurde in seiner Rechenzeit verschieden stark eingegrenzt. Es wurden mehrere Rechenläufe durchgeführt und der Mittelwert gebildet.

Die Rundlaufverfahren können für diese Kommissionierwelle nur mäßig gute Ergebnisse erreichen, da sie nur die gleichmäßige Verteilung der Kommissioniermenge zwischen den Stationen verfolgen. Insbesondere das einfache Verfahren scheint an dieser Aufgabe zu scheitern.

Wie gut die Verfahren im Durchschnitt auf größeren Ausprägungen der Problemstellung und mehreren Kommissionierwellen der beiden Bereiche der Testdaten arbeiten, behandelt der folgende Abschnitt.

Abschließend zum Vergleich zur exakten Lösung wird noch ein Diagramm (siehe Abbildung 15) gezeigt, welches alle 6.435 gültigen Lösungsmöglichkeiten der Problemstellung aufsteigend sortiert mit den zugehörigen ZFW darstellt. Die besten gefundenen Lösungen der Verfahren sind als Punkte eingezeichnet.

In Abbildung 16 ist zu sehen, dass sehr oft mehrere Lösungen den gleichen ZFW liefern. Im diesem Ausschnitt der besten 100 Lösungsmöglichkeiten ist zu erkennen, wie nahe die besten Verfahren für diese Ausprägung der Problemstellung beieinander liegen.

Abbildung 15: Diagramm - Vergleich zu Exakt

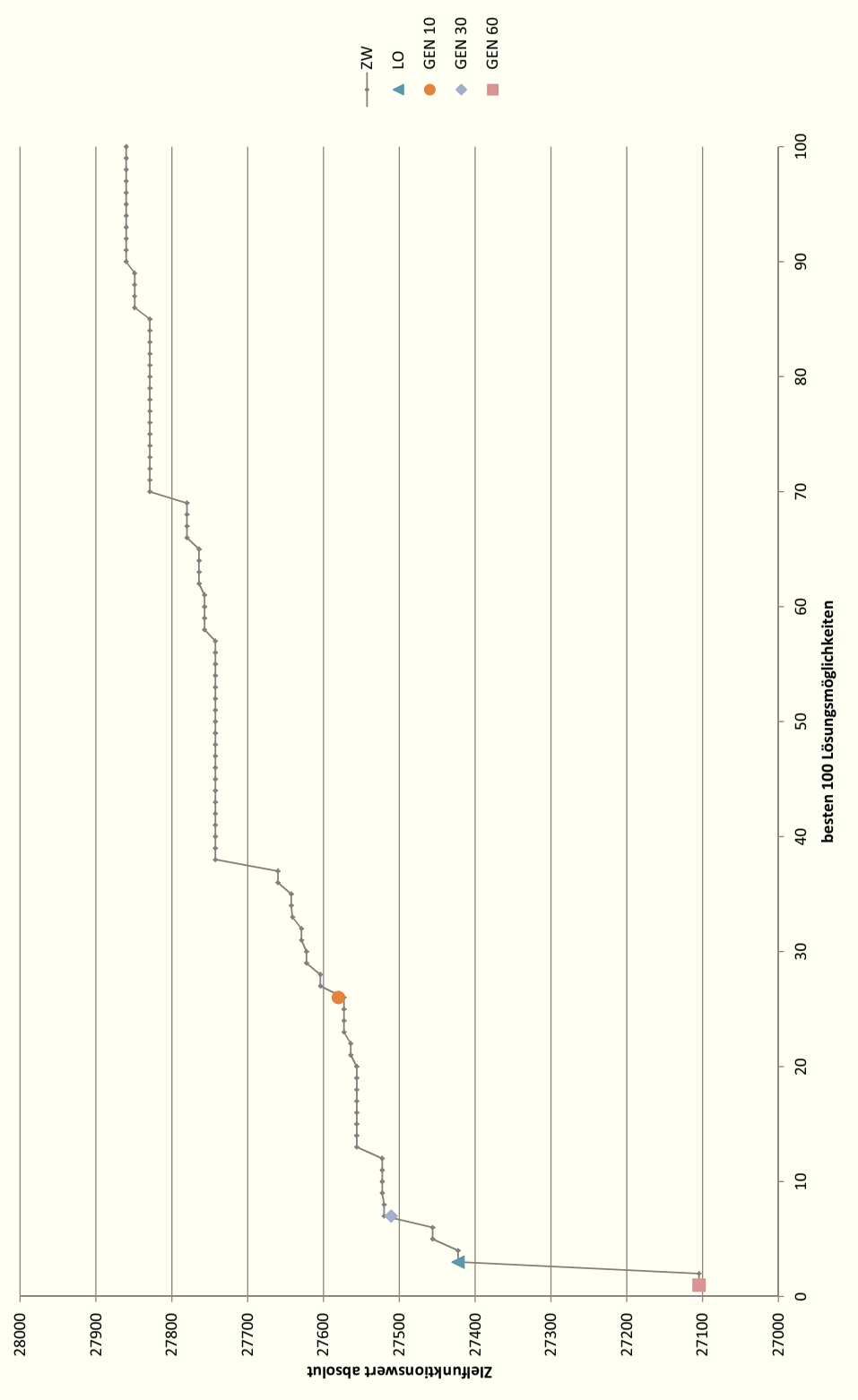


Abbildung 16: Diagramm - Vergleich zu Exakt

5.7.2. Vergleich über alle Tage der Testdaten

Für das *Fallbeispiel* ist es von Bedeutung, wie gut sich die Verfahren bei realistischen Problemgrößen über verschiedene Tage in beiden Bereichen verhalten. Es werden dafür über hundert, an Hand der Auftragsdaten und -struktur unterschiedliche Kommissionierwellen, untersucht. Dabei werden nur Szenarien mit mehr als zwei PKS-Stationen betrachtet, da die anderen keine wirkliche Aufgabe für die Teilsystemoptimierung im PKS darstellen.

Bei mehr als zwei PKS Stationen oder mehr als zehn Schächten ist es mit dem umgesetzten exakten Verfahren nur noch in unakzeptabler Rechenzeit möglich, die beste Lösung zu berechnen. Somit wird nun für den Vergleich der Zielfunktionswerte die beste gefundene Lösung herangezogen.

Außerdem soll gezeigt werden, wie sich die Rechenzeit der Verfahren im Vergleich zueinander bei größeren Problemstellungen verändert. Tabelle 13 zeigt die Abweichung zum ZFW der besten gefundenen Lösung in Prozent und die Rechenzeit die im Mittel für die Berechnung einer Kommissionierwelle benötigt wird. Bei der Abweichung wird der Mittelwert und der Median über alle Abweichungen, der 107 Kommissionierwellen mit insgesamt 233 aktiven Stationen, gebildet.

Tabelle 13: Ergebnisse des Vergleichs der Algorithmen über alle Tage

Kennzahl	**RR**	**RR imp**	**LO**	**Gen 10**	**Gen 60**
Median der Abweichung	4,327%	3,224%	0,025%	0,519%	0,000%
Mittelwert der Abweichung	34,935%	5,989%	0,575%	1,988%	0,831%
Standardabweichung d. Abw.	87,446%	8,605%	1,287%	7,606%	3,239%
Mittelwert d. Rechenzeit [s]	1,192	1,425	7,235	12,848	65,044
Standardabweichung d. RZ	0,432	0,432	11,609	2,654	3,034

Das einfache Rundlaufverfahren (kurz: RR) liegt im Mittel sehr weit vom besten Zielfunktionswert entfernt. Der Median und die Standardabweichung als Maß für die Streuung[17] lassen darauf schließen, dass einige sehr schlechte Ausreißer den gemittelten Wert so schlecht machen. Das verbesserte Rundlaufverfahren (kurz: RR imp) liefert etwas bessere, aber wesentlich konstantere Zielfunktionswerte. Dafür benötigt es im Schnitt nur etwas mehr an Rechenzeit. Die lokale Suche (kurz: LO) liefert gemittelt die besten Werte in Bezug auf den besten gefundenen ZFW. Auch der Rechenaufwand ist im Vergleich zu den genetischen Algorithmen relativ gering, jedoch mit wesentlich höherer Streuung. Der Median und die kleine Standardabweichung lassen eine konstante Lösungsgüte vermuten. Der genetische Algorithmus benötigt mehr Rechenzeit, um eine wirklich sehr gute Lösung zu finden. Wenn er diese Zeit bekommt, findet er in den meisten Fällen auch eine bessere Lösung als seine Konkurrenz. Jedoch liegt er in manchen Fällen deutlich weiter vom Optimum entfernt als die lokale Suche und daher ist der gemittelte Wert

[17] Anmerkung: Das heißt, die Stichprobenstandardabweichung σ^* als Quadratwurzel der nicht korrigierten Stichprobenvarianz. $\sigma^* = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$

auch etwas schlechter, trotz deutlich längerer Laufzeit. In diesem Test geht die lokale Optimierung deutlich als beste Alternative hervor.

Tabelle 14: Ergebnisse des Vergleichs der Algorithmen über alle Tage nach Bereich

Ber.	Kennzahl	RR	RR imp	LO	Gen 10	Gen 60
GKB	Median der Abw.	2,154%	1,799%	0,000%	0,325%	0,016%
	Mittelwert der Abw.	17,960%	3,701%	0,159%	1,932%	0,853%
	Standardabw. d. Abw.	68,534%	5,812%	0,305%	10,072%	3,778%
	Mittelwert RZ [s]	1,344	1,652	10,224	12,949	64,892
	Standardabw. d. RZ	0,453	0,373	14,586	2,844	3,031
UKB	Median der Abw.	8,637%	6,503%	0,315%	1,501%	0,000%
	Mittelwert der Abw.	54,678%	8,638%	1,069%	2,015%	0,789%
	Standardabw. d. Abw.	101,635%	10,405%	1,748%	2,066%	2,397%
	Mittelwert RZ [s]	1,004	1,146	3,561	12,724	65,231
	Standardabw. d. RZ	0,316	0,321	3,801	2,394	3,027

Tabelle 14 zeigt die selben Kennzahlen nochmals nach Bereich getrennt. Im UKB sind durchschnittlich weniger Aufträge und somit auch weniger aktive Stationen und Schächte, sowie zu kommissionierende Artikel zu erwarten. Das heißt, die Größe des Problems ist im Durchschnitt kleiner als im GKB. Die Rechenzeiten der nicht zeitlich begrenzen Verfahren (RR, RR imp und LO) sind daher wesentlich kürzer. Die Rundlaufverfahren haben Probleme, die Filialen gleichmäßig unter den Schächten zu verteilen. Der genetische Algorithmus profitiert am stärksten vom kleineren Lösungsraum. Bei den größeren Ausprägungen der Problemstellung werden die Stärken der lokalen Suche deutlich.

5.7.3. Vergleich DST zu SPT

Abschließend noch ein Vergleich der Verfahren getestet am Durchschnittstag und am Spitzentag. Der SPT benötigt im Schnitt mehr verschiedene Artikel, Filialen sowie aktive Stationen im Bereich PKS, als der DST, um die geforderte Kommissionierleistung zu erfüllen. Die Auftragsstruktur ist bei diesen beiden Tagen vergleichbar. Die Verfahren werden mit jeweils den selben Parametern und Ausgangslösungen gestartet und die Ergebnisse sind in Abbildung 17 dargestellt.

Es ist zu sehen, dass am DST der genetische Algorithmus (mit maximaler Laufzeit von 600 Sekunden, kurz: GEN 600) die „beste“ Lösung, über alle Verfahren hinweg, ermitteln kann. Am SPT ist es die lokale Optimierung (bzw. Suche, kurz: LO), welche die „beste“ Lösung findet.

Die Rundlaufverfahren finden relativ gute Lösungen, die jedoch gemessen an der Lösungsgüte nicht mit den anderen Verfahren mithalten können. Sie überzeugen durch ihre konstant kurze Rechenzeit und sind als Startlösungen für die anderen Verfahren prädestiniert.

Der genetische Algorithmus benötigt bei größeren Problemstellungen mehr Rechenzeit, um den Lösungsraum ausreichend gut abzudecken. Neben dem größeren Lösungs-

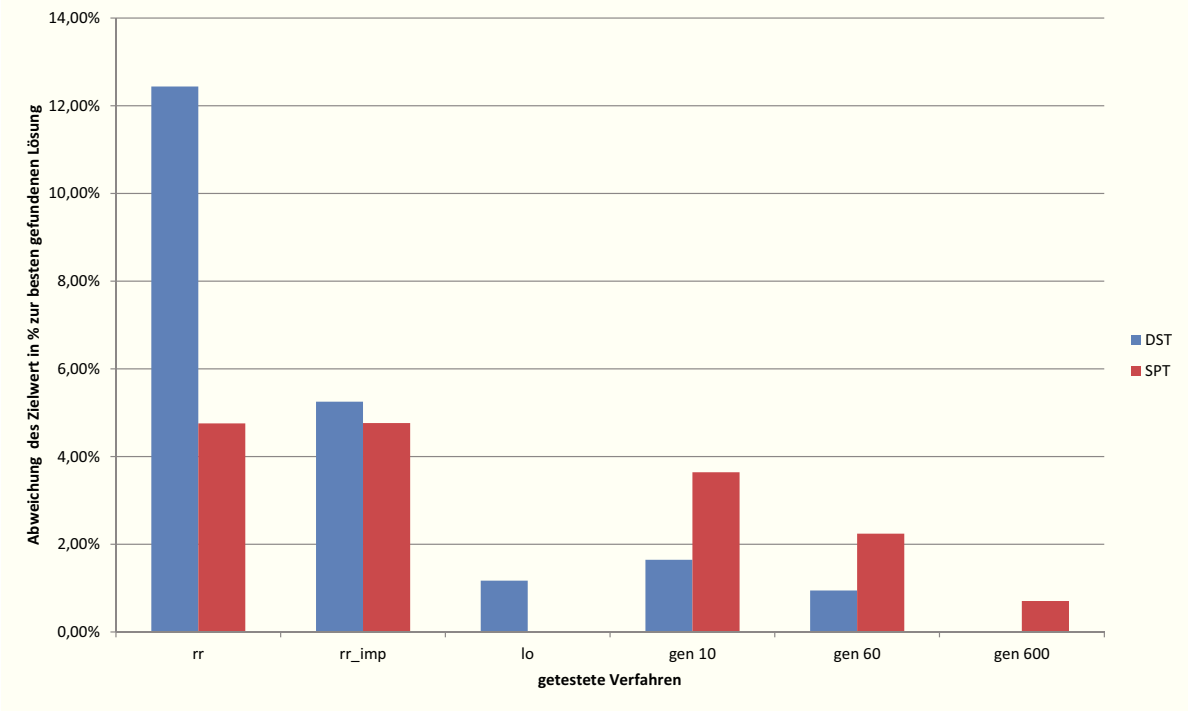


Abbildung 17: Diagramm - Vergleich DST zu SPT

raum sind die rechenzeitintensiveren Grundoperationen (wie Kreuzung, Bewertung der Fitness) bei steigender Problemgröße dafür verantwortlich.

5.7.4. Vergleich der lokalen Optimierung

Eine Woche an Testdaten wird verwendet, um die umgesetzten Verfahren der lokalen Optimierung nach Vorbild der *2-Opt-Heuristik* zu vergleichen. Jede Kommissionierwelle durchläuft zuvor die Auftragszeilenverteilung. Das verbesserte Rundlaufverfahren liefert die selbe Startlösung für jede Kommissionierwelle. Es werden folgende fünf Verfahren untereinander verglichen:

Verfahren	**Beschreibung**
LO_1	Änderung: „Vertauschen“, sofortige Änderung, ein Durchlauf
LO_2	Änderung: „Vertauschen“, sofortige Änderung, ein erweiterter Durchlauf
LO_3	Änderung: „Verschieben“ und „Vertauschen“, sofortige Änderung, ein erweiterter Durchlauf
LO_4	Änderung: „Verschieben“ und „Vertauschen“, beste Änderung, solange Verbesserung
LO_5	Änderung: „Verschieben“ und „Vertauschen“, sofortige Änderung, solange Verbesserung

Abbildung 18 zeigt die Ergebnisse. Auf der linken Achse wird die Lösungsgüte als Abweichung in Prozent zur besten gefunden Lösung der fünf Verfahren aufgetragen. Die Lösungsgüte wird durch den Zielwert definiert. Die Balken symbolisieren den Mittelwert und die Standardabweichung über alle Kommissionierwellen der betrachteten Woche. Auf der rechten Achse ist die benötigte Rechenzeit in Sekunden aufgetragen. Die Linien stehen für den Mittelwert und die Standardabweichung der Rechenzeit über alle Kommissionierwellen der betrachteten Woche.

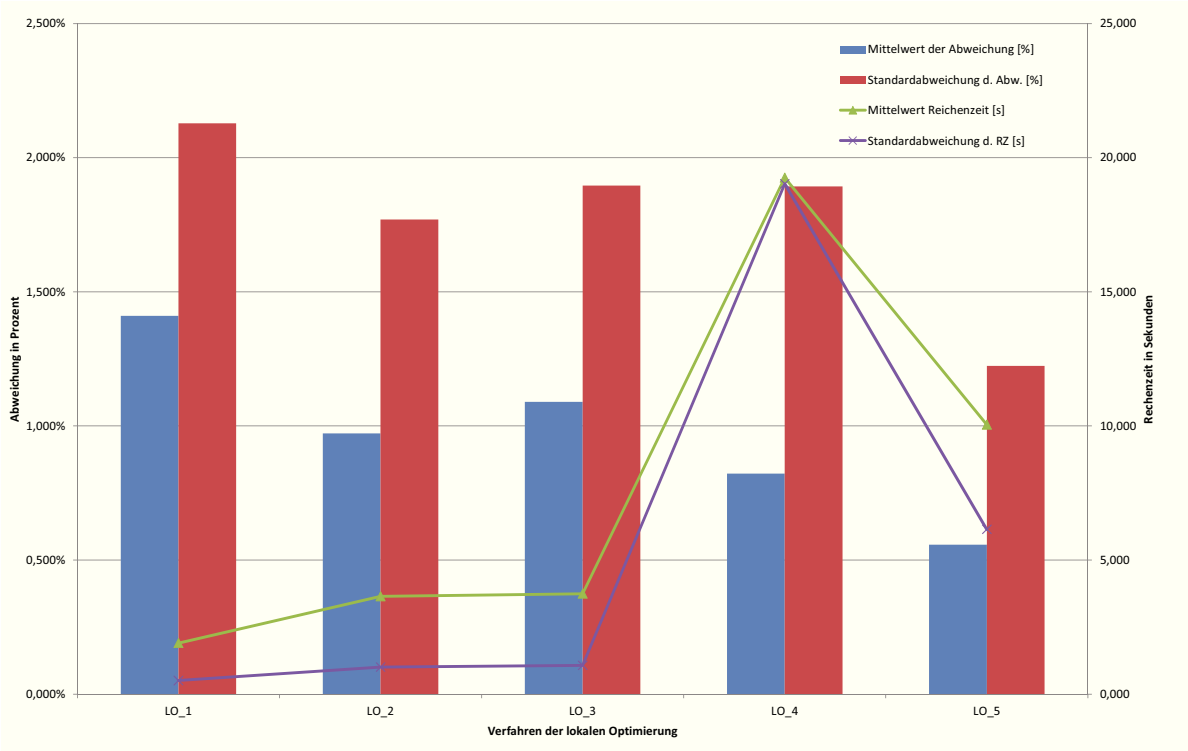


Abbildung 18: Diagramm - Vergleich der lokalen Optimierung

Die einzelnen Verfahren liegen anhand der Mittelwerte der Lösungsgüte nicht einmal eine Prozent auseinander. Das Verfahren ohne „Verschieben“ (LO_1) erzielt die schlechtesten Zielfunktionswerte, da ihm eine eingeschränktere Nachbarschaft zu Suche zur Verfügung steht. Aufgrund seiner Einfachheit punktet das Verfahren durch die geringste benötigte Rechenzeit. Die Verfahren mit sofortiger Änderung und nur einem kompletten Durchlauf (LO_1, LO_2 und LO_3), lassen aufgrund ihrer guten Ergebnisse darauf schließen, das für die betrachteten Testdaten bereits nach einer Iteration ein Ergebnis nahe am lokalen Optimum erreicht wird. Ihre Rechenzeit liegt bei circa 1 Sekunde im Durchschnitt. Die Verfahren, welche solange abgearbeitet werden, bis keine Verbesserung in der Nachbarschaft mehr möglich ist, (LO_4 und LO_5) benötigen ein Vielfaches der Zeit, um im Schnitt etwas bessere Lösungen zu erreichen. Insbesondere der klassische Ansatz (LO_4), indem vor einer Änderung die gesamte Nachbarschaft abgesucht wird, kann unter Anbetracht des Verhältnisses mittlerer Zielfunktionswert zu benötigter Rechenzeit nicht überzeugen.

6. Ergebnisse und Erkenntnisse

Dieser Abschnitt der Arbeit soll die Ergebnisse und Erkenntnisse in kompakter Form zusammenfassen. Zuerst wird noch einmal auf die umgesetzten Verfahren und ihre möglichen Potenziale eingegangen. Danach werden Erkenntnisse und Erfahrungen des Autors festgehalten, welche sich auf die Herausforderungen beziehen, welche beim Zusammenführen von Theorie und Praxis auftreten können. Zum Abschluss werde einige Ergebnisse der Simulation gezeigt.

6.1. Die Verfahren

Alle implementierten Verfahren konnten entweder durch ihren geringen Rechenaufwand oder die Lösungsgüte überzeugen. Welches der Verfahren im praktischen Umfeld zum Einsatz kommen wird, hängt von der verfügbaren Rechenzeit und der benötigten Qualität der Lösung ab. Diese beiden Richtwerte sind zum aktuellen Zeitpunkt noch nicht ausreichend genau definiert. In jedem Fall muss das Verfahren konstant gute Lösungen in einer bestimmbaren Zeit liefern können. Auch eine Kombination der Verfahren in gewissen Bereichen ist nicht unmöglich.

In allen umgesetzten Verfahren stecken mit Gewissheit noch Potenziale bzgl. Laufzeitoptimierung (z.B. durch leistungsfähigere Datenstrukturen) und Stabilität. Zusätzlich könnten bei Bedarf noch weitere Verfahren oder bereits erwähnte wie B&B oder DP auf ihre Anwendung für das *Fallbeispiel* untersucht und umgesetzt werden. Beim Teilproblem der Auftragszeilenverteilung wäre die Umsetzung eines exakten Lösungsverfahren wünschenswert.

Für den praktischen Einsatz von Verfahren ist es von Vorteil, wenn diese (von allen Beteiligten) leicht nachvollzogen und somit auch leichter umgesetzt werden können. Wenn ein Verfahren bei jeder Berechnung mit gleicher Eingabe, die selben Ergebnisse liefert, ist dies für Außenstehende leichter zu akzeptieren, als wenn immer andere Ergebnisse mit gleicher oder im Schnitt sogar besserer Lösungsgüte gefunden werden. Dies ist jedoch nur von Bedeutung, wenn ein Verfahren wiederholt auf die selben Daten angewandt werden muss oder für das Verfahren Transparenz gefordert wird.

In Anbetracht der verfügbaren Informationen, dem für die finale Umsetzung im Projekt entstehenden Aufwand und der Ergebnisse der Bewertung ist für die Auftragszeilenverteilung die Variante „die stärksten Artikel und deren Filialen (2-stufig)“ und für die Auftragsverteilung im PKS die lokale Optimierung (adaptierter *k-Opt*) am besten geeignet. Sollte die verfügbare Rechenzeit zur Lösungsfindung unerwartet vermindert werden, bzw. die voraussichtliche Problemgröße ansteigen, sollten die „schnelleren“ Heuristiken benutzt werden.

6.2. Erkenntnisse der Arbeit

Für die Analyse des Prozesses der Auftragsverteilung ist es notwendig, den gesamten Ablauf in kleinere Teilprozesse und kleinere Problemstellungen aufzuteilen. Das gesamte Problem in einem mathematischen Modell zu beschreiben und in akzeptabler Zeit zu

lösen, war dem Autor dieser Arbeit nicht möglich. Es ist einfacher für übersichtlichere Problemstellungen, passende Konzepte in der Fachliteratur zu finden. Es wurden zwei der Teilprozesse (Auftragszeilenverteilung und Teilsystemoptimierung PKS) vom Systemanbieter für die Optimierung in dieser Arbeit bestimmt. Diese können relativ unabhängig voneinander verbessert und angepasst werden. Zusätzlich können die implementieren Verfahren zum Lösen eines Teilproblems ohne erwähnenswerte Einschränkungen ausgetauscht werden.

Eine übergreifende Optimierung ist nicht möglich, da bestimmt Teilprozesse nicht verändert oder von anderen System optimiert werden. Dies schränkt die Möglichkeiten (neben den vielen einzuhaltenden Bedingung und Vorgaben) stark ein. Die Abläufe sind daher teilweise komplizierter als nötig gestaltet.

Durch die Optimierung der beiden Teilprobleme könnten messbare Verbesserungen im Vergleich zum zuvor geplanten Vorgehen erzielt werden. Wie stark die Verbesserungen ausfallen, hängt hauptsächlich von der Problemgröße und der Auftragsstruktur ab. Neben dem ansatzweise vorhandenen, geplanten heuristischen Vorgehen des Systemanbieters wurden passende Vorgangsweisen aus der Literatur untersucht und umgesetzt. Dabei stellte sich das Identifizieren und Priorisieren von konkreten Zielen und harten oder schwachen Nebenbedingungen als nicht triviale Aufgabe heraus.

Durch das Auswerten der *Testdaten* wurde in dieser Arbeit festgestellt, dass die vorher durchgeführte Wellen- und Toureneinteilung möglicherweise einen sehr großen Einfluss auf die Lösungsqualität der in dieser Arbeit behandelten Teilprobleme hat. Es wird daher in Betracht gezogen, im nächsten Schritt der Optimierung, welche nicht Inhalt dieser Arbeit ist, zumindest die Welleneinteilung mit in eine etwas umfassendere Optimierung aufzunehmen. Außerdem wird die Verbesserung des Rollcontainerfüllgrades in der finalen Umsetzung für das Projekt nicht von der Auftragszeilenverteilung durchgeführt. Dieser Schritt wird bei Bedarf in veränderter Form von einem der Subsysteme im Lagersystem übernommen.

6.3. Ergebnisse der Simulation

Durch die Simulation des Systemanbieters können die, durch die hier vorgestellten Verfahren ermittelten Lösungen, auf ihren praktischen Einsatz geprüft werden, obwohl das konkrete Lagersystem physikalisch noch nicht errichtet ist. Die Simulation bildet alle wichtigen Prozesse und Bestandteile eines Lagers in einer Softwareumgebung ab. Somit kann bestimmt werden, ob das Lagersystem die verplanten Aufträge mit der geforderten Nennleistung und in der definierten Zeit abarbeiten kann. Außerdem kann eine Vielzahl an Kennzahlen und Nebenbedingungen überprüft werden und wie sich deren Verletzung auf den Lagerbetrieb auswirkt.

Es werden an dieser Stelle Ausschnitte der Auswertung eines Testfalles des Durchschnittstages im ungekühlten Bereich präsentiert. Es wird mit zwei PKS Stationen gearbeitet und es sollen rund 18.000 GVE in maximal 16 Stunden kommissioniert werden. Die Aufträge des Tages werden durch die beschriebenen Verfahren auf die PKS-Stationen und Schächte verteilt.

Die Simulation ergibt, dass die Kommissioniermenge im PKS an diesem Tag nach

etwas mehr als 15 Stunden erfolgreich abgearbeitet wurde. Es wurde mit einer durchschnittlichen Leistung von 594 GVE/h gearbeitet. Somit wird die geforderte Nennleistung von 550 GVE/h erreicht. Die folgenden Diagramme zeigen einige durch die Simulation erhaltenen Informationen.

Abbildung 19 zeigt einerseits die ausgelagerten Paletten über die Kommissionierzeit, sowie im rechten Diagramm die Leistung des Kommissionierers über die Kommissionierzeit. Gearbeitet wird von 6:00 bis 21:00 Uhr und in den Diagrammen sind die Übergänge zwischen den Kommissionierwellen an den Einbrüchen der Funktionen um 11:00, 13:00 und kurz vor 18:00 Uhr zu erkennen. Die gesamte Kommissionierdauer ist in diesem Fall nicht gleichmäßig unter den Wellen verteilt. Die zweite Welle benötigt nur zwei Stunden für die Kommissionierung im Gegensatz zur ersten Welle, die ungefähr fünf Stunden benötigt. Diese Einbrüche sind vom Ablauf der PKS-Kommissionierung her unumgänglich.

Ansonsten sind die Verläufe der Kurven relativ konstant auf einem wünschenswerten Niveau. Um 11:00 Uhr erreicht die Kurve der Palettenauslagerungen kurzzeitig einen kritischen Wert von ungefähr 80 Paletten/h. Dies ist darauf zurückzuführen, dass zu Beginn einer Welle viele Paletten in kurzer Zeit aus dem HRL ausgelagert werden müssen.

Abbildung 20 visualisiert im linken Diagramm die Auslastung des Kommissionierers und rechts die mittlere Entnahmemenge in GVE pro Palette pro Station über die Kommissionierzeit. Die Kurvenverläufe zeigen, dass der Kommissionierer beinahe vollständig mit dem Kommissionieren von Ware beschäftigt ist. Eine verständliche Ausnahme sind die Übergange zwischen den Kommissionierwellen. Auch die Entnahmemengen pro Palette sind relativ konstant, jedoch könnte ein Miteinbeziehen des Volumens der Artikel in die Berechnungen und Verfahren, die am stärksten schwankende Kurve in diesem Diagramm, positiv beeinflussen.

Im Großen und Ganzen zeigen die Ergebnisse der Simulation die erwarteten Ergebnisse. Werden bereits bei der Verteilung der Aufträge kritische Werte festgestellt, so kann die Simulation diese durch einen Einbruch der Kommissionierleistung bestätigen. Im Betrieb des Lagersystems gilt es, diese Situationen so früh wie möglich zu erkennen und bei Möglichkeit durch automatisches oder manuelles Eingreifen zu verhindern. Sollten keine Korrekturmaßnahmen durchführbar sein, kann zumindest darauf hingewiesen werden, dass mit Leistungseinbußen und Terminverzug zu rechnen ist.

6.4. Der Vergleich

Als Abschluss wird ein Vergleich zwischen der Auftragsverteilung zu Beginn dieser Arbeit und der Auftragsverteilung zum Zeitpunkt des Abschlusses dieser Arbeit durchgeführt. Am Beginn dieser Arbeit existierte eine teilweise automatisierte Auftragsverteilung der Planungsabteilung des Systemanbieters. Diese hat in ihrer letzten Version die Möglichkeit innerhalb einer Kommissionierwelle Aktions- und Normalaufträge in zwei getrennten kleineren Wellen innerhalb der Kommissionierwelle abzuarbeiten. Somit hat diese Version einen Vorteil gegenüber der in dieser Arbeit behandelten Verfahren, da diese innerhalb einer Kommissionierwelle bis zu doppelt so viele Filialen einer PKS-Station zuweisen kann. Die verfügbare Zeit der Kommissionierwelle ist dabei unverändert.

Am besten vergleichbar ist die Auftragsverteilung der Planungsabteilung mit einer

Auftragsverteilung dieser Arbeit, in welcher keine Trennung zwischen Aktions- und Normalartikel stattfindet. In dieser Version kann die Verteilung zumindest die Aktions- und Normalaufträge einer Filiale auf einen Schacht zuweisen. Es ist nur nicht möglich die Aufträge verschiedener Filialen auf einen Schacht zuzuweisen.

Zum Vergleich wird eine Woche an Auftragsdaten herangezogen, für welche die Planungsabteilung die Auftragsverteilung durchgeführt hat. Zur Bewertung des Vergleichs wird die Anzahl unterschiedlicher SKU pro Kommissionierwelle ermittelt. Eine Minimierung der SKU-Anzahl wird in allen Verfahren verfolgt. Auf die Ergebnisse der Filiale-zu-Station-Zuordnung innerhalb des PKS wird in diesem Vergleich nicht näher eingegangen, da die Version der Planungsabteilung für diesen Schritt nur ein einfaches Rundlaufverfahren angewandt hat.

Abbildung 21 vergleicht die SKU-Anzahl der Kommissionierwellen zwischen der Auftragszeilenverteilung der Planungsabteilung und der Auftragszeilenverteilung dieser Arbeit mit und ohne Trennung nach Aktions- und Normalaufträgen. Die Kommissionierwellen wurden im Diagramm absteigend nach der besten gefundenen SKU-Anzahl sortiert. Das automatisierte Verfahren ohne Trennung nach Aktion- und Normalaufträgen (in Tabelle kurz: Auto A+N, grüne Linie) liefert für jeder Kommissionierwelle dieser Woche das beste Ergebnis, gemessen an der SKU-Anzahl. Das automatisierte Verfahren mit Trennung (in Tabelle kurz: Auto, rote Linie) liefert im Durchschnitt bessere Ergebnisse als das Verfahren der Planungsabteilung (in Tabelle kurz: Plan, blaue Linie). Bei Kommissionierwellen mit vielen Kunden und großen zu kommissionierenden Ziel-GVE-Mengen kann das Verfahren mit Trennung nicht mit den anderen beiden konkurrieren, da es nur halb so viele Filialenaufträge den Stationen zuweisen kann.

Der Vergleich zeigt, dass die neu entwickelten Verfahren dieser Arbeit für diese Woche an Auftragsdaten trotz strengerer Vorgaben bessere Ergebnisse erzielen, als das alte Verfahren der Planungsabteilung. Dabei wird die vollständige Automatisierung und der nachfolgenden Schritt der Teilsystemoptimierung im PKS der neuen Verfahren, welche weitere Vorteile darstellen, in diesen Vergleich nicht miteinbezogen.

Warum die neuen Verfahren im Vergleich besser sind, wird in den Diagrammen in Abbildung 22 und Abbildung 23 gezeigt. Es wird eine aussagekräftige Kommissionierwelle der Woche an Auftragsdaten herangezogen. Die Kennzahlen GVE pro Artikel und Filialen pro Artikel werden gegenüber der ausgewählten Artikel des neue Verfahrens ohne Trennung (kurz: Auto, rote Balken) und des alten Verfahrens der Planungsabteilung (kurz: Plan, blaue Balken) visualisiert. Das neue Verfahren kann gegenüber dem alten für einen Artikel sehr oft mehr Filialen und GVE-Mengen finden. Somit kann es mit weit weniger verschiedenen Artikeln die gleiche Ziel-GVE-Kommissioniermenge erreichen, ohne dabei die Vorgaben zu verletzten. [18]

[18] Anmerkung: Der Artikel mit der Artikelnummer „0284493“ darf im neuen Verfahren nicht gewählt werden, da er für die Sonderkommissionierung markiert ist und daher nicht am PKS kommissioniert werden darf. Dies ist eine weitere zusätzliche Einschränkung, welche nur in den neuen Verfahren berücksichtigt wird.

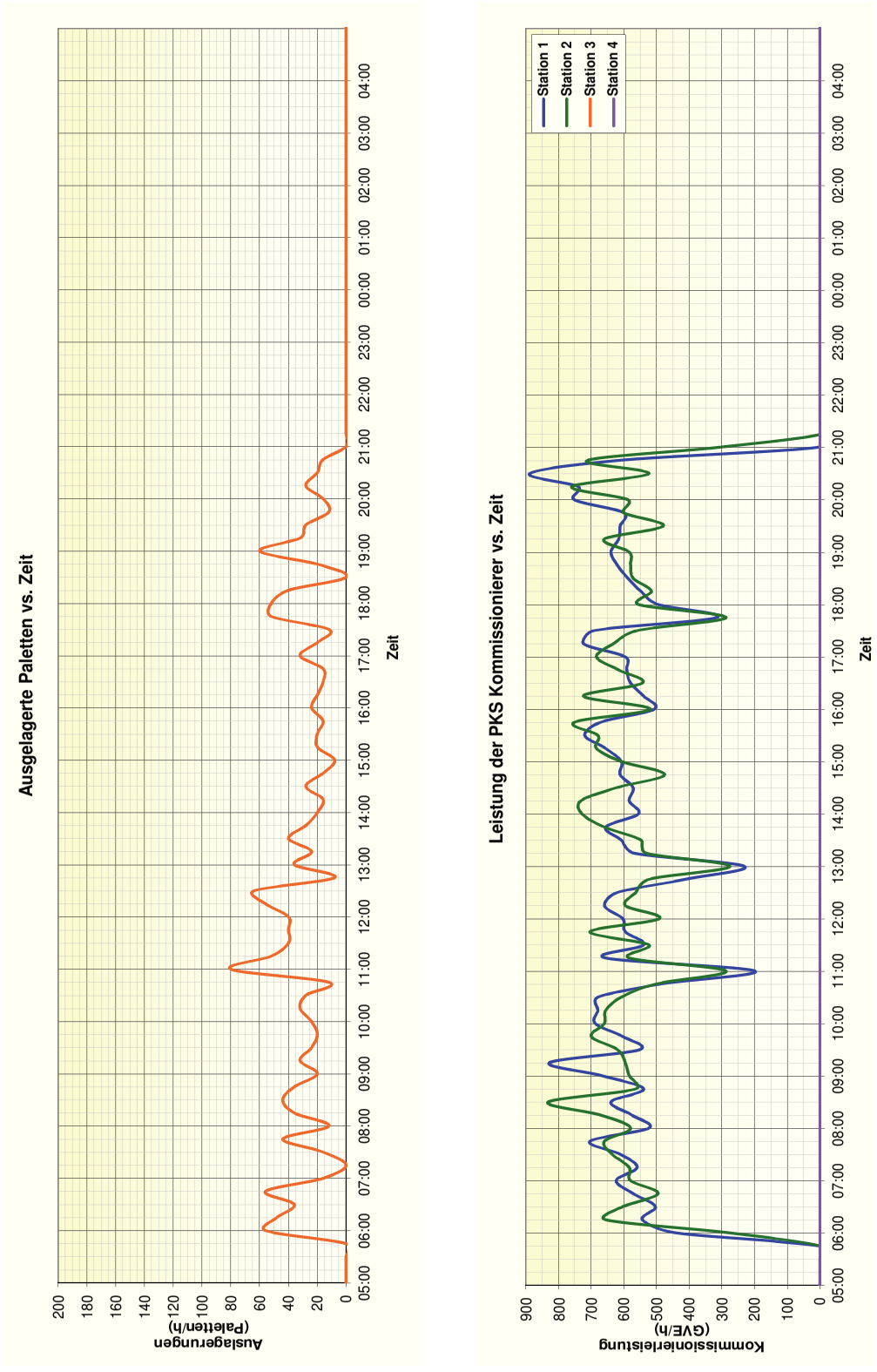


Abbildung 19: Simulation - Palettenauslagerungen und Leistung (Quelle:[11])

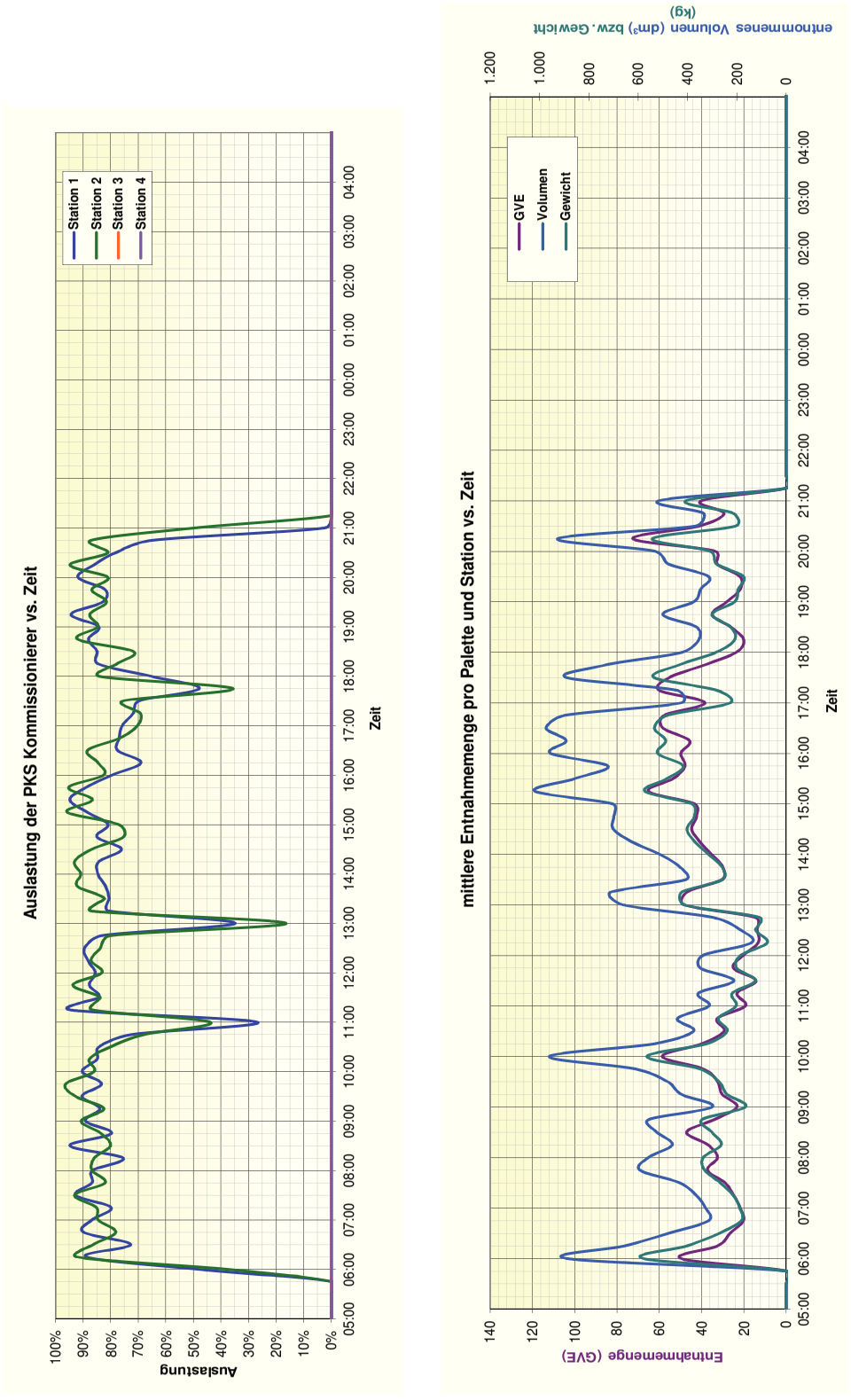


Abbildung 20: Simulation - Auslastung und Entnahmemenge (Quelle:[11])

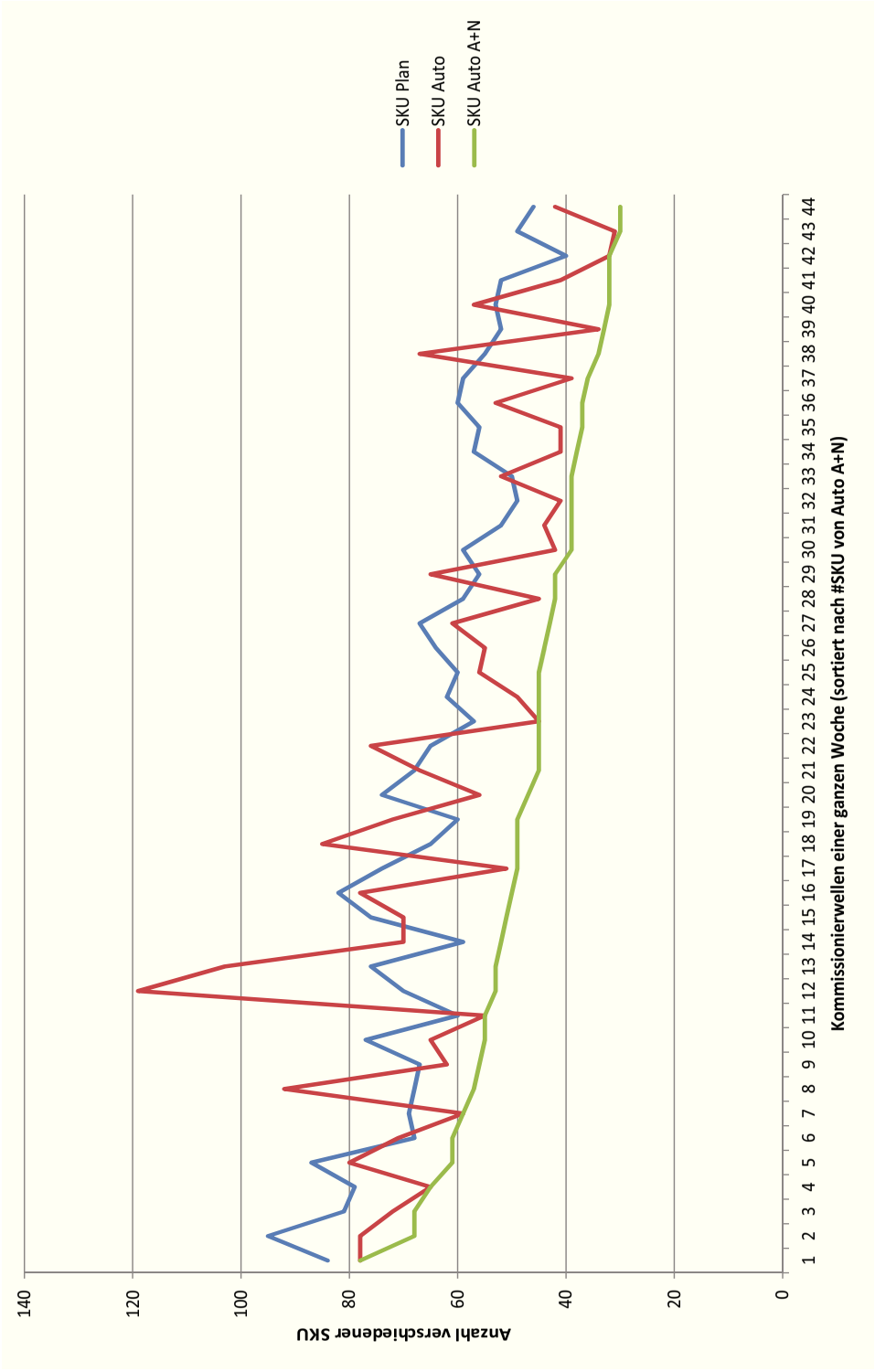


Abbildung 21: Diagramm - Vergleich der Auftragsverteilung (vorher - nachher)

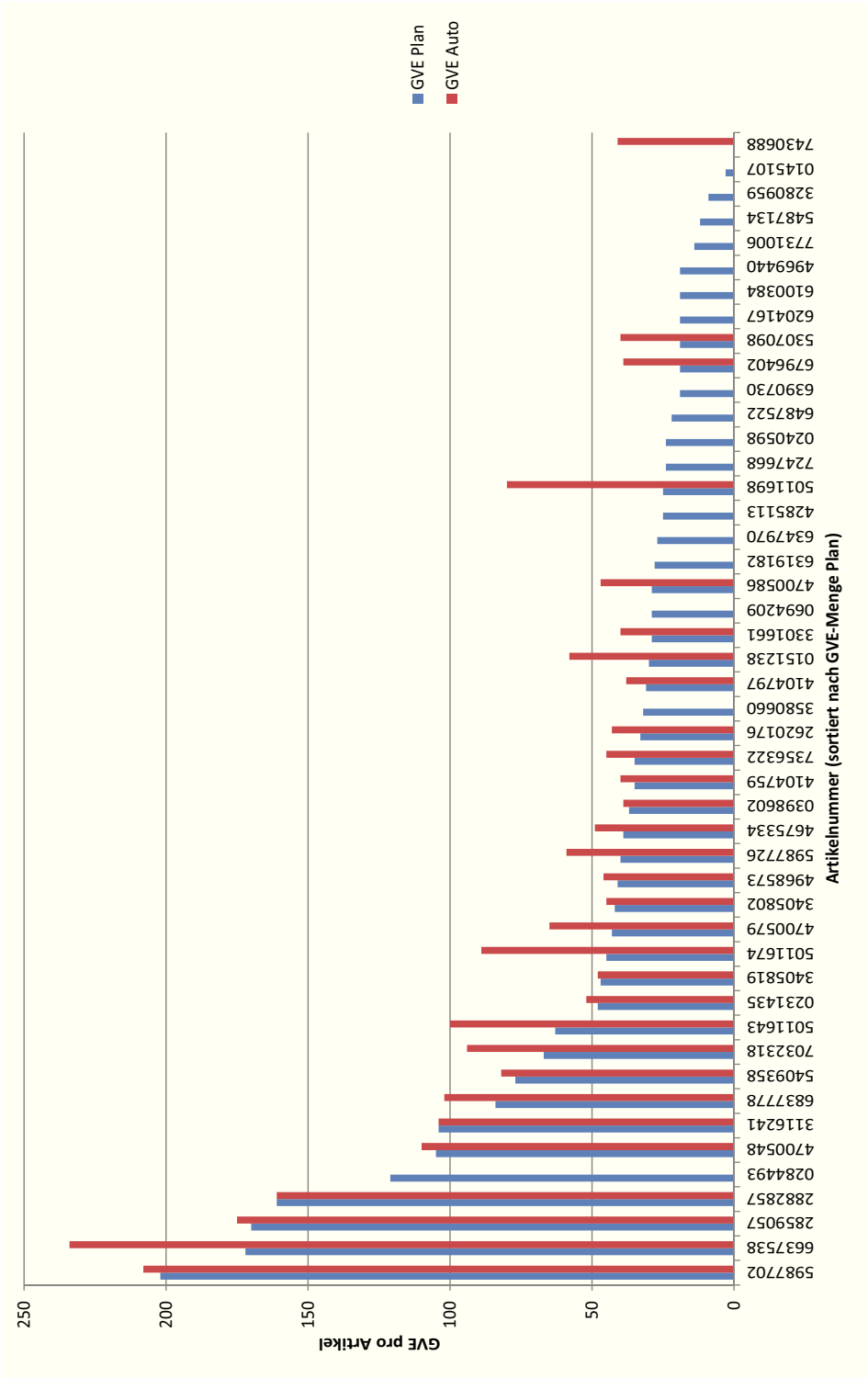


Abbildung 22: Diagramm - Vergleich der GVE-Menge pro Artikel

Abbildung 23: Diagramm- Vergleich der Anzahl Filialen pro Artikel

Anhang

A. Beispiele

A.1. Beispiel 1: Auftragszeilenverteilung, sFuA

Dieses Beispiel zeigt, warum der *sFuA*-Algorithmus nicht immer die minimale Anzahl an SKU (verschiedenen Artikeln) auswählt. Es wurde extra für diesen Zweck entworfen und ist keine übliche Auftragsstruktur des *Fallbeispiels*.

Es soll ein PKS-Ziel von 200 GVE erreicht werden. Für die Kommissionierung steht eine PKS-Station mit zwei Schächten zur Verfügen. Es können somit maximal zwei Filialen gewählt werden. Tabelle 15 zeigt die Filialen mit ihren bestellten GVE-Mengen an Artikeln. Es wird zusätzlich die GVE-Summen je Filiale über alle Artikel in der letzten Spalte dargestellt. Nach dieser Reihung wählt der Algorithmus die Filialen aus.

Tabelle 15: Tabelle der Auftragsstruktur für Beispiel 1

Filiale	**Artikel**					**Filial-**
	A 1	**A 2**	**A 3**	**A 4**	**A 5**	**summe**
F 1	40	40	30	10	20	140
F 2	50	30	10	25	15	130
F 3	5	5	5	5	105	125
F 4	5	5	5	5	95	115
Summe	100	80	50	45	235	510

Der *sFuA*-Algorithmus würde die Filialen F1 und F2 aufgrund ihrer Summe auswählen. Um das PKS-Ziel von 200 GVE zu erreichen, müssen die drei Artikel A1, A2 und A3 kommissioniert werden. Die gewählten Filialen und Artikel (in der Tabelle rot markiert) kommen auf eine GVE-Summe von 200. Es wäre jedoch für das PKS besser, Filiale F3 und F4 mit Artikel A5 zu wählen, da so nur ein Artikel kommissioniert werden muss. Der *sAuf*-Algorithmus würde diese bessere Wahl (in der Tabelle grün markiert) treffen, da er Artikel A5 als ersten wählen würde, da für diesen die größte GVE-Menge bestellt wird.

A.2. Beispiel 2: Auftragszeilenverteilung, sAuF

Dieses Beispiel zeigt, warum der *sAuF*-Algorithmus nicht immer die minimale Anzahl an SKU (verschiedenen Artikeln) auswählt. Es wurde extra für diesen Zweck entworfen und ist keine übliche Auftragsstruktur des *Fallbeispiels*.

Es soll ein PKS-Ziel von 200 GVE erreicht werden. Für die Kommissionierung steht eine PKS-Station mit zwei Schächten zur Verfügen. Es können somit maximal zwei Filialen gewählt werden. Tabelle 16 zeigt die Filialen mit ihren bestellten GVE-Mengen an Artikeln. Es werden zusätzlich die GVE-Summen der Artikel über alle Filialen und über die besten zwei Filialen je Artikel in den letzten beiden Spalten dargestellt. Nach letzterer sortiert der Algorithmus die Artikel und fügt je Iteration einen zur Lösung hinzu.

Tabelle 16: Tabelle der Auftragsstruktur für Beispiel 2

Artikel	**Filialen**				**Artikel-summe**	**Artikelsumme der besten 2 Filialen**
	F 1	**F 2**	**F 3**	**F 4**		
A 1	52	52	1	5	110	104
A 2	1	52	51	5	109	103
A 3	51	1	51	5	108	102
A 4	51	1	1	50	103	101
A 5	49	1	1	50	101	99
Summe	204	107	105	115	531	509

Der *sAuF*-Algorithmus würde die Artikel A1, A2 und A3 aufgrund ihrer GVE-Summe auswählen. Um das PKS-Ziel von 200 GVE zu erreichen, müssen zwei Filialen, beispielsweise F1 und F2, kommissioniert werden. Die gewählten Filialen und Artikel (in der Tabelle rot markiert) kommen auf eine GVE-Summe von 209. Es wäre jedoch für das PKS besser, Filiale F1 und F4 mit Artikel A4 und A5 zu wählen, da so nur zwei Artikel anstatt von drei kommissioniert werden müssen. Der *sFuA*-Algorithmus würde in diesem Beispiel die bessere Wahl (in der Tabelle grün markiert) treffen, da er Filiale F1 und F4 wählen würde, da diese beiden Filialen die größten GVE-Mengen bestellen.

Wird das Beispiel, wie in Tabelle 17 gezeigt, abgeändert, so finden beide Verfahren die schlechtere (rote) Lösung mit drei Artikeln. Der *sFuA*-Algorithmus wählt nun auch die Filialen F1 und F2 aus, da diese beiden Filialen die meisten GVE-Mengen bestellen.

Tabelle 17: Tabelle der Auftragsstruktur für Beispiel 2 (verändert)

Artikel	**Filialen**				**Artikel-summe**	**Artikelsumme der besten 2 Filialen**
	F 1	**F 2**	**F 3**	**F 4**		
A 1	52	52	1	1	106	104
A 2	1	52	51	1	105	103
A 3	51	1	51	1	104	102
A 4	51	1	1	50	103	101
A 5	49	1	1	50	101	99
Summe	204	107	105	103	519	509

A.3. Beispiel 3: beschränkte Enumeration, 3 Stationen

Beim kleineren Beispiel handelt es sich um einen PKS-Bereich mit 3 Stationen. Jede Station verfügt über 2 Schächte. Es werden genau 6 Filialen auf die verfügbaren Plätze verteilt.

$\binom{6}{2} = 15$ mögliche Filiale-zu-Station-Zuordnungen von je zwei unterschiedlichen Filialen werden erzeugt, wenn die Kombinationen ohne Wiederholungen für eine Station berechnet werden. Werden die Filialen von 1 bis 6 durchnummeriert, so sind es folgende Filiale-zu-Station-Zuordnungen: 1 2, 1 3, 2 3, 1 4, 2 4, 3 4, 1 5, 2 5, 3 5, 4 5, 1 6, 2 6,

3 6, 4 6 und 5 6. Aus diesen Kombinationen werden $\binom{6}{2}\frac{1}{3} \cdot \binom{4}{2}\frac{1}{2} \cdot \binom{2}{2}\frac{1}{1} = \frac{15}{3} \cdot \frac{6}{2} \cdot 1 = 15$ unterschiedliche[19] Lösungen erzeugt. Diese Lösungen sind in Tabelle 18 angeführt.

Tabelle 18: Tabelle der möglichen Lösungen

	Station				**Station**				**Station**		
Lösung	1	2	3	**Lösung**	1	2	3	**Lösung**	1	2	3
1	1 2	3 4	5 6	d.L.	1 4	2 3	5 6	d.L.	1 5	2 3	4 6
2	1 2	3 5	4 6	**10**	1 4	2 5	3 6	d.L.	1 5	2 4	3 6
3	1 2	4 5	3 6	**11**	1 4	3 5	2 6	d.L.	1 5	3 4	2 6
d.L.	1 2	3 6	4 5	d.L.	1 4	2 6	3 5	d.L.	1 5	2 6	3 4
d.L.	1 2	4 6	3 5	d.L.	1 4	3 6	2 5	d.L.	1 5	3 6	2 4
d.L.	1 2	5 6	3 4	d.L.	1 4	5 6	2 3	d.L.	1 5	4 6	2 3
4	1 3	2 4	5 6	d.L.	2 4	1 3	5 6	d.L.	2 5	1 3	4 6
5	1 3	2 5	4 6	**12**	2 4	1 5	3 6	d.L.	2 5	1 4	3 6
6	1 3	4 5	2 6	**13**	2 4	3 5	1 6	d.L.	2 5	3 4	1 6
d.L.	1 3	2 6	4 5	d.L.	2 4	1 6	3 5	d.L.	2 5	1 6	3 4
d.L.	1 3	4 6	2 5	d.L.	2 4	3 6	1 5	d.L.	2 5	3 6	1 4
d.L.	1 3	5 6	2 4	d.L.	2 4	5 6	1 3	d.L.	2 5	4 6	1 3
7	2 3	1 4	5 6	d.L.	3 4	1 2	5 6				
8	2 3	1 5	4 6	**14**	3 4	1 5	2 6				
9	2 3	4 5	1 6	**15**	3 4	2 5	1 6		u.s.w.		
d.L.	2 3	1 6	4 5	d.L.	3 4	1 6	2 5				
d.L.	2 3	4 6	1 5	d.L.	3 4	2 6	1 5				
d.L.	2 3	5 6	1 4	d.L.	3 4	5 6	1 2				

In der Spalte „Lösung“ steht eine Zahl, wenn eine Lösung zuvor noch nicht vorgekommen ist, und diese ist zugleich die Nummer der gefundenen Lösung. Die Abkürzung d.L. steht für doppelte Lösung und zeigt an, dass bereits zuvor eine Lösung gefunden wurde, welche mit Sicherheit den selben Zielfunktionswert liefert. Diese doppelten Lösungen können verworfen werden. Zum Beispiel die Lösung 1 2, 3 6, 4 5 liefert den selben Zielfunktionswert wie Lösung Nummer 3. Es sind lediglich die Filiale-zu-Station-Zuordnungen von Station 2 und 3 vertauscht.

Die horizontalen Linien nach jeder sechsten Lösung markiert einen Wechsel der Filiale-zu-Station-Zuordnung auf der ersten Station. Werden die Zuordnungen von Station 2 und 3 bei den ersten sechs Lösungen betrachtet, so kann erkannt werden, dass die zweite Hälfte der Lösungen immer doppelte Lösungen sind. Es werden nur die Zuordnungen zwischen diesen beiden Stationen getauscht. Dies gilt auch für die restlichen Sechser-Gruppen und ist ein Erklärung für den Faktor $\frac{6}{2}$ bei der Berechnung der Anzahl unterschiedlicher Lösungen. Der Faktor $\frac{15}{3}$ ist etwas schwieriger zu erklären. Im Beispiel kann

[19] Anmerkung: Diese Lösungen unterscheiden sich sehr wahrscheinlich vom Zielfunktionswert, obwohl gleiche Zielfunktionswerte immer noch möglich sind. Es werden lediglich jene doppelten Lösungen ausgeschlossen, welche mit Sicherheit den gleichen Zielfunktionswert liefern.

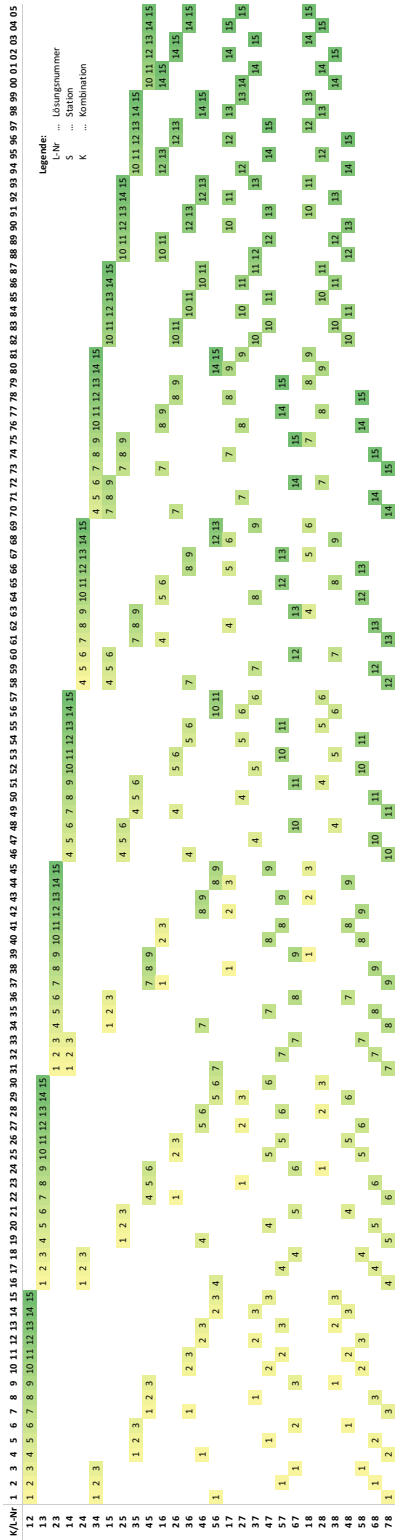
von 15 möglichen Sechser-Gruppen mit jeweils 3 verbliebenen, relevanten Lösungen ausgegangen werden. Jede Gruppe liefert im Schnitt nur eine neue Lösung und die restlichen zwei sind doppelte Lösungen. Das Beispiel zeigt jedoch, dass die ersten Sechser-Gruppen zwei bis drei neue Lösungen und die Gruppen zum Schluss nur noch doppelte Lösungen finden. Die Anzahl der berechneten Lösungen stimmt aber mit der Anzahl der erzeugten Lösungen überein.

A.4. Beispiel 4: beschränkte Enumeration, 4 Stationen

Das gesamte Beispiel zur beschränkten Enumeration ist in Abbildung 24 zu sehen. Es handelt sich um einen PKS-Bereich mit 4 Stationen. Jede Station verfügt über 2 Schächte. Es werden genau 8 Filialen auf die verfügbaren Plätze verteilt.

L-Nr/S	1	2	3	4
1	01	02	03 04	05 06 07 08
2	01	02	03 04	05 07 06 08
3	01	02	03 04	06 07 05 08
4	01	02	03 05	04 06 07 08
5	01	02	03 05	04 07 06 08
6	01	02	03 05	06 07 04 08
7	01	02	04 05	03 06 07 08
8	01	02	04 05	03 07 06 08
9	01	02	04 05	06 07 03 08
10	01	02	03 06	04 07 05 08
11	01	02	03 06	05 07 04 08
12	01	02	04 06	03 07 05 08
13	01	02	04 06	05 07 03 08
14	01	02	05 06	03 07 04 08
15	01	02	05 06	04 07 03 08
16	01	03	02 04	05 06 07 08
17	01	03	02 04	05 07 06 08
18	01	03	02 04	06 07 05 08
19	01	03	02 05	04 06 07 08
20	01	03	02 05	04 07 06 08
21	01	03	02 05	06 07 04 08
22	01	03	04 05	02 06 07 08
23	01	03	04 05	02 07 06 08
24	01	03	04 05	06 07 02 08
25	01	03	02 06	04 07 05 08
26	01	03	02 06	05 07 04 08
27	01	03	04 06	02 07 05 08
28	01	03	04 06	05 07 02 08
29	01	03	05 06	02 07 04 08
30	01	03	05 06	04 07 02 08
31	02	03	01 04	05 06 07 08
32	02	03	01 04	05 07 06 08
33	02	03	01 04	06 07 05 08
34	02	03	01 05	04 06 07 08
35	02	03	01 05	04 07 06 08
36	02	03	01 05	06 07 04 08
37	02	03	04 05	01 06 07 08
38	02	03	04 05	01 07 06 08
39	02	03	04 05	06 07 01 08
40	02	03	01 06	04 07 05 08
41	02	03	01 06	05 07 04 08
42	02	03	04 06	01 07 05 08
43	02	03	04 06	05 07 01 08
44	02	03	05 06	01 07 04 08
45	02	03	05 06	04 07 01 08
46	01	04	02 05	03 06 07 08
47	01	04	02 05	03 07 06 08
48	01	04	02 05	06 07 03 08
49	01	04	03 05	02 06 07 08
50	01	04	03 05	02 07 06 08
51	01	04	03 05	06 07 02 08
52	01	04	02 06	03 07 05 08
53	01	04	02 06	05 07 03 08
54	01	04	03 06	02 07 05 08
55	01	04	03 06	05 07 02 08
56	01	04	05 06	02 07 03 08
57	01	04	05 06	03 07 02 08
58	02	04	01 05	03 06 07 08
59	02	04	01 05	03 07 06 08
60	02	04	01 05	06 07 03 08
61	02	04	03 05	01 06 07 08
62	02	04	03 05	01 07 06 08
63	02	04	03 05	06 07 01 08
64	02	04	01 06	03 07 05 08
65	02	04	01 06	05 07 03 08
66	02	04	03 06	01 07 05 08
67	02	04	03 06	05 07 01 08
68	02	04	05 06	01 07 03 08
69	02	04	05 06	03 07 01 08
70	03	04	01 05	02 06 07 08
71	03	04	01 05	02 07 06 08
72	03	04	01 05	06 07 02 08
73	03	04	02 05	01 06 07 08
74	03	04	02 05	01 07 06 08
75	03	04	02 05	06 07 01 08
76	03	04	01 06	02 07 05 08
77	03	04	01 06	05 07 02 08
78	03	04	02 06	01 07 05 08
79	03	04	02 06	05 07 01 08
80	03	04	05 06	01 07 02 08
81	03	04	05 06	02 07 01 08
82	01	05	02 06	03 07 04 08
83	01	05	02 06	04 07 03 08
84	01	05	03 06	02 07 04 08
85	01	05	03 06	04 07 02 08
86	01	05	04 06	02 07 03 08
87	01	05	04 06	03 07 02 08
88	02	05	01 06	03 07 04 08
89	02	05	01 06	04 07 03 08
90	02	05	03 06	01 07 04 08
91	02	05	03 06	04 07 01 08
92	02	05	04 06	01 07 03 08
93	02	05	04 06	03 07 01 08
94	03	05	01 06	02 07 04 08
95	03	05	01 06	04 07 02 08
96	03	05	02 06	01 07 04 08
97	03	05	02 06	04 07 01 08
98	03	05	04 06	01 07 02 08
99	03	05	04 06	02 07 01 08
100	04	05	01 06	02 07 03 08
101	04	05	01 06	03 07 02 08
102	04	05	02 06	01 07 03 08
103	04	05	02 06	03 07 01 08
104	04	05	03 06	01 07 02 08
105	04	05	03 06	02 07 01 08

(a) tabellarisch



(b) grafisch

Abbildung 24: Gesamtes Beispiel zum Erzeugen von Lösungen

B. QAP

Weiterführende Literatur zum Quadratischen Zuordnungsproblem ist unter [5], [6] und [7] zu finden.

Die mathematische Formulierung des Problems der Standortbestimmung von Fabriken als QAP kann wie folgt aussehen. Die folgende Problembeschreibung ist aus [5] entnommen und wurde vom Englischen ins Deutsche übersetzt.

Angenommen, es wird eine Menge von Fabriken auf eine Menge von Standorten verteilt und es muss dabei jedem Standort genau eine Fabrik zugeordnet werden. Zu Minimieren sind die Kosten des Warenflusses (das heißt, die Anzahl der Waren mal der Entfernung) zwischen den Fabriken und die Kosten für das Errichten der Fabriken. Die Problemgröße n sei die Anzahl an Fabriken und Standorten und definiert als $N = \{1, 2, \ldots, n\}$. Als Eingabewert werden die drei $n \times n$ Matrizen $F = (f_{ij}), D = (d_{kl}), B = (b_{ik})$ benötigt. Dabei ist f_{ij} der Fluss an Waren zwischen Fabrik i und Fabrik j, d_{kl} die Entfernung zwischen Standort k und Standort l und b_{ik} gibt die Kosten an, die beim Errichten der Fabrik i am Standort k entstehen. Das Minimierungsproblem sei definiert wie folgt:

$$\min_{\pi \in \mathcal{S}_n} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\pi(i)\pi(j)} + \sum_{i=1}^{n} b_{i\pi(i)} \tag{12}$$

wobei $\mathcal{S}_n$ die Menge aller Permuationen $\pi : N \to N$ ist.

Jedes der Produkte $f_{ij} d_{\pi(i)\pi(j)}$ steht für die Kosten des Warenflusses, wenn Fabrik i auf Standort $\pi(i)$ und Fabrik j auf Standort $\pi(j)$ errichtet wird. Die Summe über $b_{i\pi(i)}$ steht für die gesamten Kosten zum Errichten der Fabriken. Bei der Standortbestimmung von Fabriken sind die Matrizen F und D symmetrisch und besitzen Nullen in den Hauptdiagonalen. Alle Matrizen sind nicht negativ. Der lineare Teil des QAP kann bei anderen Problemstellungen wegfallen (d.h $B = 0$).

Laut der Fachliteratur kann das QAP derzeit nur für relativ kleine Problemstellungen in akzeptabler Rechenzeit exakt gelöst werden. Zum Einsatz kommen dafür zumeist Ansätze des B&B und des Schnittebenen-Verfahren. Um schneller an Lösungen zu gelangen bzw. um größere Problemstellungen zu lösen, bieten sich Heuristiken und Metaheuristiken wie Lokale Suche, Tabu Search, Simulated Annealing, Genetische Algorithmen und Ameisenalgorithmen an.

Die Auftragsverteilung im PKS verfolgt nicht die selben Ziele wie ein QAP. Folgender Ansatz kann die Auftragsverteilung im PKS zumindest teilweise in ein QAP überführen. Es soll versucht werden, die Anzahl verschiedener Artikel auf einer PKS Station so klein wie möglich zu halten. Als Entfernung dient die räumliche Position der Schächte. Das heißt, Schächte einer Station haben eine sehr kleine bzw. keine Entfernung zueinander. Schächte unterschiedlicher Stationen haben ein sehr große Entfernung zueinander. Statt dem Warenfluss zwischen den Fabriken, wird die Anzahl gleicher Artikel zwischen zwei Filialen (=Aufträgen) herangezogen. Somit erreicht die Minimierung, dass Filialen mit vielen gleichen Artikel auf den Schächten einer Station platziert werden.

Dieser Ansatz würde jedoch in dieser Arbeit nicht weiter verfolgt, da die gleichmäßige Verteilung der Kommissioniermengen unter den Stationen nicht berücksichtigt wird.

C. Datenauszüge

Dieser Abschnitt des Anhangs beinhaltet einen Datenauszug zu den Auswertungen aus Abschnitt 4.5.2. Folgende Daten sind in den Tabellen geführt:

Kurzzeichen	**Erklärung**
ID	Identifikation des Datensatzes, Prefix (P - Phase, F - sFuA) und Nummer
ZEIT	Benötigte Kommissionierzeit in Stunden
ZIEL	Kommissionierziel in GVE über beide Bereiche
ZIELPKS	Kommissionierziel in GVE des Bereichs PKS
ZIELTKS	Kommissionierzile in GVE des Bereichs TKS
TYP	Lagerbereich (FD - GKB, TS - UKB)
TAG	Kommissioniertag
BATCH	Kommissionierwelle
ANZPKS	Anzahl an aktiven PKS-Stationen
ANZTKS	Anzahl an aktiven TKS-Stationen
ANZMA	Anzahl an benötigten Mitarbeitern
GVE PKS	Kommissioniermenge in GVE im Bereich PKS
SKU PKS	Verschiedenen SKU im Bereich PKS
FILIALEN	Verschiedenen Filialen im Bereich PKS
RC	Anzahl an vorberechneten Rollcontainern im PKS
FG	Durchschnittlicher Füllgrad der vorberechneten Rollcontainer im PKS
GPRC	Durchschnittliche GVE-Menge pro Rollcontainer
SKU TKS	Verschiedenen SKU im Bereich TKS
GPS	Durchschnittliche GVE-Menge pro SKU im PKS
SKUPH	Durchschnittliche SKU pro Stunde im PKS
RECHENZEIT	Rechenzeit in Sekunden

Tabelle 19: Datenauszug Teil 1 - Phase und sFuA

ID	ZEIT	ZIEL	ZIELPKS	ZIELTKS	TYP	TAG	BATCH	ANZPKS	ANZTKS	ANZMA
P 1	3,027	22399	6659	15740	FD	DST5	1	4	8	12
P 2	2,771	20509	6097	14412	FD	DST5	2	4	8	12
P 3	2,625	19426	5775	13651	FD	DST5	3	4	8	12
P 4	3,085	22831	6788	16043	FD	DST5	4	4	8	12
P 5	2,202	16292	4844	11448	FD	8	1	4	8	12
P 6	2,012	14889	4426	10463	FD	8	2	4	8	12
P 7	1,736	12845	3819	9026	FD	8	3	4	8	12
P 8	1,486	10997	3269	7728	FD	8	4	4	8	12
P 9	2,322	9868	3831	6037	TS	8	1	3	4	7
P 10	1,626	6909	2682	4227	TS	8	2	3	4	7
P 11	2,593	11019	4278	6741	TS	8	3	3	4	7
P 12	1,864	7923	3076	4847	TS	8	4	3	4	7
ID	**ZEIT**	**ZIEL**	**ZIELPKS**	**ZIELTKS**	**TYP**	**TAG**	**BATCH**	**ANZPKS**	**ANZTKS**	**ANZMA**
F 1	3,027	22399	6659	15740	FD	DST5	1	4	8	12
F 2	2,771	20509	6097	14412	FD	DST5	2	4	8	12
F 3	2,625	19426	5775	13651	FD	DST5	3	4	8	12
F 4	3,085	22831	6788	16043	FD	DST5	4	4	8	12
F 5	2,202	16292	4844	11448	FD	8	1	4	8	12
F 6	2,012	14889	4426	10463	FD	8	2	4	8	12
F 7	1,736	12845	3819	9026	FD	8	3	4	8	12
F 8	1,486	10997	3269	7728	FD	8	4	4	8	12
F 9	2,322	9868	3831	6037	TS	8	1	3	4	7
F 10	1,626	6909	2682	4227	TS	8	2	3	4	7
F 11	2,593	11019	4278	6741	TS	8	3	3	4	7
F 12	1,864	7923	3076	4847	TS	8	4	3	4	7

Tabelle 20: Datenauszug Teil 2 - Phase und sFuA

ID	GVE PKS	SKU PKS	FILIALEN	RC	FG	GPRC	SKU TKS	GPS	SKUPH	RZ
P 1	6660	112	48	140	79,383%	63,788	1698	14,866	37,002	63,951
P 2	6132	75	48	119	75,935%	70,013	1682	20,44	27,061	4,118
P 3	5808	99	46	134	77,855%	59,46	1718	14,667	37,712	5,392
P 4	6808	126	48	148	78,945%	61,203	1730	13,508	40,839	63,155
P 5	4852	86	36	138	78,964%	46,091	1702	14,105	39,062	4,354
P 6	4436	83	25	122	79,863%	47,692	1705	13,361	41,252	4,207
P 7	3834	79	25	113	78,640%	44,85	1686	12,133	45,512	4,316
P 8	3284	84	34	94	75,845%	44,778	1579	9,774	56,525	4,511
P 9	3833	85	26	84	75,668%	48,635	1600	15,031	36,608	3,773
P 10	2689	69	17	57	73,190%	48,663	1430	12,99	42,445	3,569
P 11	4278	88	26	96	75,872%	49,008	1656	16,205	33,941	3,565
P 12	3091	56	21	65	73,166%	50,336	1432	18,399	30,039	3,529
ID	**GVE PKS**	**SKU PKS**	**FILIALEN**	**RC**	**FG**	**GPRC**	**SKU TKS**	**GPS**	**SKUPH**	**RZ**
F 1	6676	116	48	147	78,559%	60,991	1697	14,388	38,323	11,895
F 2	6125	76	46	117	76,722%	71,245	1718	20,148	27,422	9,73
F 3	5780	101	43	130	78,354%	60,273	1745	14,307	38,474	8,621
F 4	6797	128	48	151	78,877%	58,979	1723	13,275	41,487	10,046
F 5	4852	86	36	138	78,964%	46,091	1702	14,105	39,062	8,302
F 6	4491	83	25	124	79,832%	47,716	1690	13,527	41,252	7,971
F 7	3834	79	25	113	78,640%	44,85	1686	12,133	45,512	7,767
F 8	3284	84	34	94	75,890%	44,564	1579	9,774	56,525	7,972
F 9	3863	85	26	85	75,964%	49,317	1597	15,149	36,608	8,136
F 10	2689	69	17	57	73,190%	48,663	1430	12,99	42,445	7,171
F 11	4278	88	26	94	76,136%	49,232	1656	16,205	33,941	8,301
F 12	3091	56	21	65	73,166%	50,336	1432	18,399	30,039	6,737

Glossar

Distributionszentrum
: Ein Distributionszentrum stellt oft den zentralen Knoten innerhalb der Wertschöpfungskette eines Distributionsnetzes dar. Neben den klassischen Lagerfunktionen kann es auch für den Umschlag verantwortlich sein. [21]

Diversität
: (lat.: *diversitas*) Vielfältigkeit [25]

Enumeration
: (lat.: *enumeratio*) Aufzählung [25]

Hochregallager
: Sehr hohe Lager die in ihrer Bauform an herkömmliche Regale erinnern. Es können diverse Ladungsträger wie Paletten, Kartons oder Kleinteilebehälter gelagert werden. Die Lagerprozesse werden meist von automatisierten Regalbediengeräten oder speziellen Staplern erledigt. [21]

Palettenkommissioniersystem
: Das Palettenkommissioniersystem ist eine Ware-zur-Person-Kommissionierung, bei welcher der Mitarbeiter die geforderte Menge an Artikeln direkt von einer Quell-Palette entnimmt und auf einen Ziel-Rollcontainer kommissioniert. Die Quell-Paletten werden entweder direkt aus dem Wareneingang oder einem HRL zugeführt. Es ist keine Depalettierung und Zwischenlagerung im AS/RS notwendig. [12]

Permutation
: (lat.: *permutatio*) Umstellung [25]

Traykommissioniersystem
: Das Traykommissioniersystem besteht aus Ware-zur-Person-Arbeitsplätzen, welche die Kommissionierung der Artikel durchführen. Die Artikel werden in Trays (flache, kistenförmige Ladungsträger, welche für die Lagerung im AS/RS und den Transport auf der Fördertechnik geeignet sind) in einem AS/RS gelagert und den Arbeitsplätzen bei Bedarf in diesen Trays zugeführt. In der Depalettierung werden die Artikel von einer (WE-)Palette auf die Trays umgelagert. Der Mitarbeiter am Arbeitsplatz kommissioniert die geforderte Menge an Artikeln aus den Quell-Trays in einen Ziel-Rollcontainer oder auf eine Ziel-Palette. [13]

Abkürzungsverzeichnis

AS/RS	Automated Storage and Retrieval System
B&B	Branch & Bound
D&C	Divide & Conquer
DC	Distribution Center
DP	Dynamic Programming
DST	Durchschnittstag
DZ	Distributionszentrum
EA	evolutionäre Algorithmen
ERM	Entity-Relationship-Modell
ES	Evolutionsstrategien
FG	Füllgrad
FTF	fahrerloses Transportfahrzeug
GA	Genetische Algorithmen
GKB	gekühlter Bereich
GVE	Großverbrauchereinheit oder Großhandels-Verkaufs-Einheit
GVE/Pal	GVE pro Palette
GVE/RC	GVE pro Rollcontainer
GVE/SKU	GVE pro SKU
HRL	Hochregallager
LKW	Lastkraftwagen
LOZ	Längste Operationszeit
LP	Leerpalette
LPT	Longest Processing Time
MHD	Mindesthaltbarkeitsdatum
Pal/h	Paletten pro Stunde
PK	Palettenkommissionierung
PKS	Palettenkommissioniersystem
QAP	Quadratic Assignment Problem
RC	Rollcontainer

RDC	Retail Distribution Center
RR	Round-Robin
RZ	Rechenzeit
SKU	Stock Keeping Unit
SKU/h	SKU pro Stunde
SPT	Spitzentag
TK	Traykommissionierung
TKS	Traykommissioniersystem
TSP	Traveling Salesman Problem
UKB	ungekühlter Bereich
WA	Warenausgang
WD	Wellendauer
WE	Wareneingang
ZFW	Zielfunktionswert

Symbolverzeichnis

Δz	Differenz von Wellendauer zu benötigter Kommissionierzeit
$\mathcal{A}$	Menge aller Artikel
$\mathcal{F}$	Menge aller Filialen
$\mathcal{Z}$	Menge aller Auftragszeilen
f	Anzahl an Filialen
k	Kommissioniermenge in GVE
k_{PKS}	Ziel-Kommissioniermenge im PKS in GVE
k_{TKS}	Ziel-Kommissioniermenge im TKS in GVE
p	Anzahl aktiver PKS-Stationen
p_{max}	Maximale Anzahl an PKS-Stationen
r_{PKS}	Nennleistung PKS-Station in GVE/h
r_{TKS}	Nennleistung TKS-Station in GVE/h
r_{Tol}	Toleranz beim Einhaltung der Nennleistung
s	Anzahl der Schächte pro PKS-Station
s_{max}	Maximale Anzahl an Schächten im PKS
sph_{max}	Maximale Anzahl an SKU pro Stunde
t	Anzahl aktiver TKS-Stationen
t_{max}	Maximale Anzahl an TKS-Stationen
w	Wellendauer in h
z	Kommissionierzeit in h

Literatur

[1] Beyer, H.G.: *The Theory of Evolution Strategies* (Natural Computing Series). – ISBN 978-3-5406-7297-5

[2] Brucker, P.: *Scheduling Algorithms*. – ISBN 978-3-5406-9515-8

[3] Brusco, M.J. ; Stahl, S.: *Branch-And-Bound Applications in Combinatorial Data Analysis* (Statistics And Computing). – ISBN 978-0-3872-5037-3

[4] Burg, Klemens ; Haf, Herbert ; Wille, Friedrich: *Höhere Mathematik für Ingenieur*. 8., überarbeitete Auflage. – ISBN 978-3-8351-0255-2

[5] Burkard, R.E. ; Çela, E. ; Pardalos, P.M. ; Pitsoulis, L.S. ; Du, D. (Hrsg.) ; Pardalos, P.M. (Hrsg.): *The Quadratic Assignment Problem* (Handbook of Combinatorial Optimization). – ISBN 978-0-7923-5019-4

[6] Burkard, R.E. ; Dell'Amico, M. ; Martello, S.: *Assignment Problems*. – ISBN 978-0-8987-1663-4

[7] Çela, E.: *The Quadratic Assignment Problem: Theory and Algorithms* (Combinatorial Optimization). – ISBN 978-0-7923-4878-8

[8] Goldberg, D.E.: *Genetic algorithms in search, optimization, and machine learning* (Artificial Intelligence). – ISBN 978-0-2011-5767-3

[9] Hubert, L.J. ; Arabie, P. ; Meulman, J.: *Combinatorial Data Analysis: Optimization by Dynamic Programming* (Siam Monographs on Discrete Mathematics and Applications). – ISBN 978-0-8987-1478-4

[10] Kleinberg, J. ; Tardos, É.: *Algorithm Design* (Alternative Etext Formats). – ISBN 978-0-3212-9535-4

[11] Knapp AG, Dokument: *Simulationsergebnisse*

[12] Knapp AG, Homepage: *Pick-it-Easy Move*. Knapp AG, 2012. – URL `http://knapp.com/glossary?char=P&iD=23`. – Abfragedatum: 15. August. 2012

[13] Knapp AG, Homepage: *Pick-it-Easy Tray*. Knapp AG, 2012. – URL `http://knapp.com/glossary?char=P&iD=24`. – Abfragedatum: 15. August. 2012

[14] Korte, Bernhard ; Vygen, Jens: *Kombinatorische Optimierung*. Springer Berlin Heidelberg. – ISBN 978-3-540-76919-4

[15] Lippe, Peter von der (Hrsg.): *Induktive Statistik*. Oldenbourg Verlag. – ISBN 978-3-486-20009-6

[16] Michalewicz, Z.: *Genetic Algorithms + Data Structures* (Artificial intelligence). – ISBN 978-3-5406-0676-5

[17] Pinedo, M.: *Planning and Scheduling in Manufacturing and Services.* – ISBN 978-1-4419-0911-4

[18] Pinedo, M.: *Scheduling: Theory, Algorithms, and Systems.* – ISBN 978-0-3877-8934-7

[19] Pohlheim, H.: *Evolutionäre Algorithmen: Verfahren, Operatoren und Hinweise Für Die Praxis* (Vdi-buch). – ISBN 978-3-5406-6413-0

[20] Reeves, C.R. ; Rowe, J.E.: *Genetic Algorithms: Principles and Perspectives : a Guide to GA Theory* (Operations Research/Computer Science Interfaces Series). – ISBN 978-1-4020-7240-6

[21] Roberts, Laura: *Gabler Wirtschaftslexikon.* 17. Auflage. Gabler Verlag, 2010. – ISBN 978-3-8349-0152-1

[22] Skiena, S.S. ; Revilla, M.A.: *Programming Challenges: The Programming Contest Training Manual* (Texts in Computer Science). – ISBN 978-0-387-00163-0

[23] Sniedovich, M.: *Dynamic Programming: Foundations and Principles* (Pure and Applied Mathematics). – ISBN 978-0-8247-4099-3

[24] Stein, C. ; Cormen, T.H. ; Rivest, R.L. ; Leiserson, C.E.: *Introduction To Algorithms.* – ISBN 978-0-262-03293-3

[25] Wermke, Matthias (Hrsg.): *Der Duden in zwölf Bänden.* Dudenverlag. – ISBN 978-3-411-04014-8, 978-3-411-04059-9 u.w.

[26] Zimmermann, Hans-Jürgen: *Operations Research.* 2. Auflage. Vieweg+Teubner Verlag. – ISBN 978-3-8348-9461-8