

---

# Optimierte Packordnung modularer Boxen

Masterarbeit an der TU Graz  
Institut für Technische Logistik

Von

Thomas Stöhr

Graz, Juli 2014

## **STATUTORY DECLARATION**

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

## **EIDESSTATTLICHE ERKLÄRUNG**

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Graz, Juli 2014

---

## **Widmung**

*„Omnia vestra – vos autem Christi“, 1 KOR 3, 22-23*

## **Danksagung**

In meiner Zeit als stud. Projektmitarbeiter am Institut für Technische Logistik durfte ich bei der Lösung vieler interessanter Problemstellungen mit einem hohen Maß an Gestaltungsfreiheit mitarbeiten. Für dieses entgegengebrachte Vertrauen möchte ich Hr. Univ.-Prof. Dr.-Ing. habil. Dirk Jodin und Hr. Ass.Prof. Dipl.-Ing. Dr.techn. Christian Landschützer danken.

Weiteren Dank gebührt Hr. Dipl.-Ing. Florian Ehrentraut für die Mitbetreuung dieser Arbeit sowie Hr. Russel Meller, PhD für die Unterstützung bei der Darstellung der einzelnen Berechnungsmethodiken.

Weiter möchte ich Hr. Dipl.-Ing. Alexander Ortner-Pichler für die große Hilfsbereitschaft und Geduld bei der programmiertechnischen Umsetzung danken.

Allen Kollegen am ITL für die vielen wertvollen Gespräche, Ideen und Inputs, sei an dieser Stelle auch mein herzlicher Dank ausgesprochen.

Meinen Eltern und meiner Familie danke ich für die ständige, verlässliche Unterstützung und Ermutigung. Ihr glaube an mich und ihr Vorbild war in vielen schwierigen Situationen eine große Motivation.

# Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Forschungsfrage.....	2
2	Problemanalyse.....	3
2.1	Begriffsdefintion.....	3
2.2	Das „Physical Internet“ (PI) und „MODULUSHCA“.....	3
2.3	Zielformulierung aus der Problemstellung heraus.....	8
3	Unit Load Analyse.....	12
3.1	Entwicklung von Maßnahmen.....	12
3.1.1	Zielformulierung.....	12
3.1.2	Rahmenbedingungen.....	12
3.1.3	Methoden.....	14
3.2	Umsetzung.....	18
3.2.1	Allgemeine Übersicht.....	18
3.2.2	Begründung der Vorgehensweise.....	22
3.3	Ergebnisse der Vorgehensweise.....	22
3.4	Bewertung/Beurteilung der Vorgehensweise.....	25
4	Unit Load Packer.....	26
4.1	Entwicklung von Maßnahmen.....	26
4.1.1	Zielformulierung.....	26
4.1.2	Rahmenbedingungen.....	26
4.1.3	Umfeld.....	27
4.1.4	Methoden.....	28
4.2	Umsetzung.....	29
4.2.1	Bin Packing - Algorithmus und Implementierung.....	30
4.2.2	Begründung.....	36
4.3	Ergebnisse der Vorgehensweise.....	36
4.3.1	Prüfdaten.....	36
4.3.2	Ergebnisse.....	41
4.4	Bewertung/Beurteilung der Vorgehensweise.....	43
5	Balanced Trailer Load.....	45
5.1	Entwicklung von Maßnahmen.....	45
5.1.1	Zielformulierung.....	45
5.1.2	Rahmenbedingungen.....	46
5.1.3	Methoden.....	52
5.2	Umsetzung.....	54
5.2.1	Balanced Trailer Load - Algorithmus und Implementierung.....	54
5.2.2	Begründung.....	58
5.3	Ergebnisse der Vorgehensweise.....	59
5.3.1	Ergebnisse.....	59
5.4	Bewertung/Beurteilung der Vorgehensweise.....	64
6	Zusammenfassung.....	66
7	Verzeichnisse.....	67
7.1	Literaturverzeichnis.....	67
7.2	Abbildungsverzeichnis.....	69
7.3	Tabellenverzeichnis.....	70
8	Anhang.....	71

---

8.1	Anhang 1 – veröffentlichter Kurztext .....	71
8.1.1	A1 Deutsch.....	71
8.1.2	A2 Englisch.....	71
8.2	Anhang 2 – Zusatzinformationen.....	72
8.2.1	Legenden.....	72
8.2.2	Programmcodes.....	72

# 1 Einleitung

Die Warendistribution steht vor einem Dilemma. Einerseits müssen Kosten, CO2-Footprint und Lieferzeiten minimiert werden und andererseits fordert der gesellschaftliche Wandel hin zu Onlineversandhandel, Urbanisierung und Individualisierung immer flexibleren und zeitnäheren Versand. Diese Entwicklungen legen nahe, dass das derzeit etablierte Distributionssystem dieser ständig an Umfang und Komplexität wachsenden Aufgabe in Zukunft nicht mehr gewachsen sein wird. In Zahlen ausgedrückt lässt sich dieses Dilemma wie folgt darstellen (vgl. [PII14]):

- Zur Zeit wird hauptsächlich Luft und Verpackungsmaterial versandt
- 25% aller gefahrenen Kilometer sind leer
- beladene LKW sind nur zu 56,8% gefüllt
- 42,6% durchschnittliche Kapazitätsauslastung

Basierend auf diesen Tatsachen wurde das Konzept des „Physical Internet (PI)“ geboren. Die grundlegende Änderung im PI ist, dass Waren nicht von einem Spediteur von Ursprungs- zu Zielort gebracht werden, sondern in standardisierten, modularen Behältnissen, von Distributionszentren (PI-Hubs) zu Distributionszentrum bis zum Endkunden (vgl. [PII14]).

Ein wesentliches Kernelement zur Etablierung des PI ist die Entwicklung des standardisierten, modularen Behältnisses. Die konstruktionsbestimmenden Einflussfaktoren sind ob der Menge an Anforderungen mannigfaltig und von vielen verschiedenen Teilbereichen bestimmt. So haben die Prozesse in der Supply Chain genauso Einfluss auf die Konstruktion wie technische Anforderungen und Normierungen.

Um die Visionen des PI in der Distributionslogistik umsetzen zu können und einen ersten geeigneten Prototyp der Transportbox zu entwickeln, wurde das MODULUSHCA-Projekt initiiert. Dieses Entwicklungsprojekt (im Rahmen des 7. Europäischen Rahmenprogramms gefördert) hat sich zum Ziel gesetzt mittels modularer, intelligenter Boxen das PI erstmals in Realität umzusetzen. (vgl. [MOD14])

Um dieses Entwicklungsprojekt erfolgreich zu absolvieren, bedarf es einiger Werkzeuge, welche zum einen die Entwicklung der modularen Boxen (im folgenden M-Boxen genannt) unterstützen und zum anderen das zu realisierende MODULUSHCA-Szenario simulieren, um es mit dem heutigen Zustand vergleichen zu können. Des Weiteren sollten diese Werkzeuge die wesentlichen Einflussfaktoren auf die Konstruktion M-Box darstellen damit dadurch Anforderungen und Funktionen identifiziert werden können.

## 1.1 Motivation

Die Motivation zu dieser Arbeit ruht auf zwei Säulen. Zum einen ist es das große Interesse des Autors an der Identifizierung und Abbildung logistischer Prozesse und zum anderen die Arbeit im Rahmen des ersten Realisierungsprojektes des PI in der Logistik. Dadurch besteht die Möglichkeit an der Realisierung eines Pilotprojektes mitzuarbeiten, das bei erfolgreichem Gelingen, weitreichende Auswirkung auf sämtliche Distributionsvorgänge haben kann.

## **1.2 Forschungsfrage**

Die Forschungsfragen, welche in dieser Arbeit behandelt werden, definieren sich aus der umfassenden Darstellung des PI-Konzeptes heraus. Nach der Darstellung jener Prozesse, die für die Realisierung des PI als relevant identifiziert werden, sollen diese unter dem Aspekt der technischen Rahmenbedingungen auf deren Realisierbarkeit überprüft werden.

Zuerst müssen, basierend auf dem PI Szenario, wesentliche Einflussfaktoren auf die einzelnen Prozesse erfasst werden. Diese Prozesse müssen in Form von Algorithmen und Berechnungsprogrammen unter Berücksichtigung der Einflussfaktoren abgebildet werden.

Anschließend soll mittels vielseitiger Variantenbildung und Tests das Bestehen des MODULUSHCA Szenarios unter realen Bedingungen verifiziert werden. Des Weiteren soll die virtuelle Abbildung wesentlicher Prozesse des PI Szenarios dazu dienen, das zukünftige MODULUSHCA Szenario mit der jetzigen Situation vergleichen zu können.

Als ergänzendes Ergebnis sollen Funktionen eines zu entwickelnden Prototypen der modularen Transportbox aus den Anforderungen, resultierend aus den Prozessen des PI, abgeleitet werden.

An dieser Stelle sei angemerkt, dass die Betrachtung die technischen Aspekte wie Stabilität, Strukturmechanik, Schiffbarkeit und gesetzliche Vorschriften beinhaltet und nicht etwaige Kommunikationstechnologien, die zur Realisierung eines PI ebenfalls von immanenter Wichtigkeit sind.

## 2 Problemanalyse

Dieses Kapitel erörtert zunächst die Rahmenbedingungen und Begrifflichkeiten dieser Arbeit und dient der Schaffung von Übersicht.

### 2.1 Begriffsdefinition

#### *Unit Loads (UL)*

Unit Loads sind zu einem Verbund gestapelte M-Boxen und werden in dieser Arbeit mit UL abgekürzt. Die Maße einer UL betragen 1200x800x2400mm (Länge-Breite-Höhe) und entsprechen in der Grundfläche dem Euro-Palettenmaß. Im MODULUSHCA Szenario, das die Rahmenbedingungen dieser Arbeit definiert, werden UL auf einer Euro-Palette gestapelt und manipuliert. In einer späteren Etablierung ist die Verwendung von speziellen PI-Staplern vorgesehen, sodass für die M-Boxen kein sekundäres Ladehilfsmittel verwendet werden muss. Durch die Normung in [DIN04] ist somit das maximale Eigengewicht einer UL auf 2.000kg beschränkt.

#### *M-Box*

Als M-Box werden im Folgenden jene Transportbehältnisse bezeichnet, die im Rahmen des MODULUSHCA Projektes am Institut für Technische Logistik (ITL) entwickelt werden. Wesentliches Merkmal der M-Box ist, dass sich die Maße ihrer Längen, Breiten und Höhen jeweils auf das Maß einer Unit-Load ergänzen und somit „modular“ sind. Des Weiteren sind sie mit Identifikations- Kommunikation- und Prozessortechnologie ausgestattet und besitzen somit die Fähigkeit mit anderen M-Boxen zu kommunizieren.

#### *FMCG*

Abkürzung für „Fast moving consumer goods“. FMCG sind pharmazeutische Produkte, Verbraucherelektronik, Konsumartikel für den Haushaltsbedarf, Körperpflegeprodukte, Alkoholika und Tabak. Die meisten dieser Produkte werden regelmäßig verbraucht und hauptsächlich in den entwickelten Ländern stark nachgefragt (vgl. [LJE14]).

### 2.2 Das „Physical Internet“ (PI) und „MODULUSHCA“

Um das Konzept des „Physical Internet“ beschreiben zu können, ist eine vorhergehende Darstellung des „Internet der Dinge“ notwendig. Das „Internet der Dinge“ ist eine Vision, die durch die fortschreitende Entwicklung und Verbreitung der Kommunikationstechnologien, wie RFID-Tags, geboren wurde. Ziel des „Internets der Dinge“ ist, die zentrale Aufbereitung und Vernetzung von Daten in Computern durch Dezentralisierung mittels „intelligenter“ Subsysteme zu ersetzen. Diese Subsysteme kennen ihre jeweilige Aufgabe und Funktion, vernetzen sich selbständig mit anderen Systemen und erfüllen die ihnen angedachte Leistung. Anders als bei zentral koordinierten Systemen müssen im „Internet der Dinge“ folglich sämtliche Subsysteme mit Identifizierungs-, Kommunikations- und Prozessortechnologie ausgestattet sein. Die technologischen Rahmenbedingungen hierfür wurden in den letzten Jahren, wesentlich beeinflusst durch die Entwicklung des Smartphones, geschaffen (vgl. [FM05]).

Das „Physical Internet“ versucht die durch das „Internet der Dinge“ bereit gestellten Kommunikationsstrukturen optimal zu nutzen, um so eine Dezentralisierung der Steuerung von Distributionsprozessen zu schaffen. Es folgt den gleichen Prinzipien wie das digitale



Internet: Die Schaffung eines offenen, globalen Systems, basierend auf physischer, digitaler und operativer Vernetzung durch standardisierte Austauschsysteme (Boxen), Schnittstellen (PI-Hubs) und Protokolle (Kommunikationsstandards). Diese Prinzip-bedingte Synergien lassen sich wie in Abbildung 2-1 darstellen.

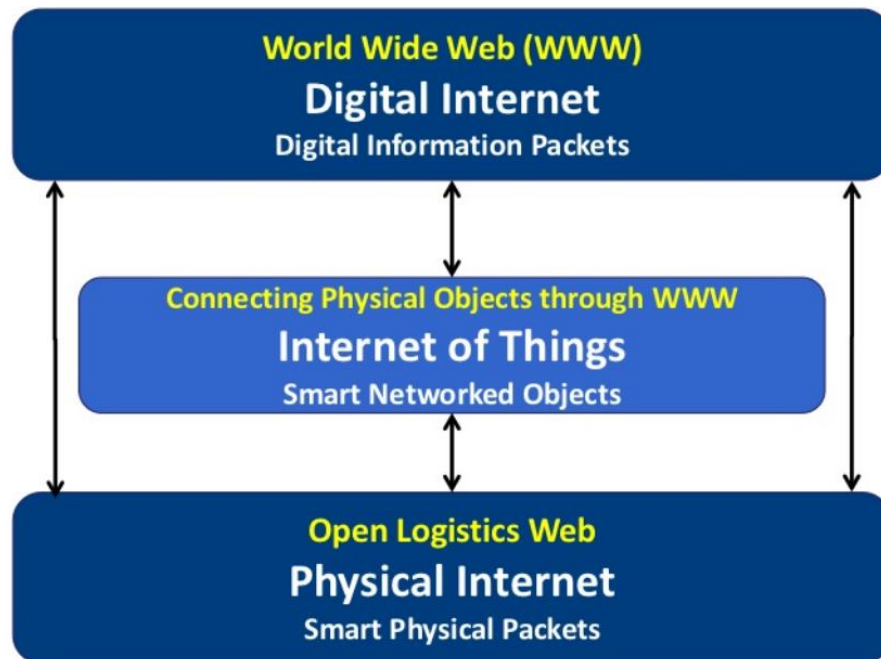


Abbildung 2-1: Synergien zwischen „Digital Internet“ - "Internet der Dinge" - "Physical Internet" [PII14]

Aus der Abbildung geht hervor, dass das digitale Internet die Vernetzung und die notwendige Datenverarbeitung ermöglicht, diese im „Internet der Dinge“ umgesetzt wird und beide Prinzipien im „Physical Internet“ vereint werden. Konkret kann zusammengefasst werden, dass im PI ein offenes, physisches System geschaffen wird, das vernetzt durch das digitale Internet, physische Größen weltweit verfügbar macht.

Ähnlich dem virtuellen Internet, in dem Daten in standardisierten Formaten ausgetauscht werden, wird im PI der Austausch von Waren in standardisierten Transportboxen realisiert. Diese haben Kommunikations- und Ortungssysteme sowie eine Recheneinheit integriert, sodass die Transportbox über deren Inhalt, Ziel, aktuellen Standort und Route zum Zielpunkt Bescheid weiß. Durch dieses Prinzip der standardisierten und intelligenten Transportboxen ist eine grundlegende technische Revolution im Bereich der Warendistribution möglich. Einer der wesentlichen Schlüsselfaktoren zur Umsetzung dieses Konzeptes ist neben dem Informations- und Kommunikationssystem, die Entwicklung und Gestaltung der Transportbox.

Um die Vision des PI erstmalig in die Realität umzusetzen wurde das Projekt namens MODULUSHCA (Abkürzung für „Modular logistics units in shared co-modal networks“) initiiert, welches von der EU im Rahmen des FP7-Programms gefördert wird. Gemeinsam mit 15 Projektpartnern aus Wirtschaft und Wissenschaft werden gemeinsam die Ziele dieses Entwicklungsprojektes verfolgt, welche sich wie folgt darstellen (vgl. [MOD14]):

- Die Vision des PI in ihren Kernkomponenten zu realisieren
- Entwicklung von modularen Transportboxen welche zu einem Verbund gestapelt und verbunden werden können
- Etablierung der digitalen Kommunikationsstandards zwischen den Transportboxen

- Entwicklung einer Logistik-Plattform als Netzwerk um eine wesentliche CO<sub>2</sub>- und Kostensenkung zu erreichen
- Auf Basis von Simulationen und Realisierungsszenarien ein „Proof of Concept“ zu erarbeiten
- Synchronisation mit gleichzeitig in USA und Kanada laufenden Projekten im Gesamtkontext des PI um somit eine Grundlage zu schaffen für eine interkontinentale Umsetzung.

Die im Rahmen dieses Entwicklungsprojektes angesiedelte Arbeit beschäftigt sich mit der Überprüfung der Realisierbarkeit relevanter Prozesse des MODULUSCHA Szenarios und soll auch die Entwicklung der modularen Transportbox unterstützen indem, basierend auf dem Realisierungsszenario, Anforderungen an diese abgeleitet werden.

Um relevante Prozesse im MODULUSCHA Szenario identifizieren zu können, wird in folgender Aufzählung der Weg einer Transportbox in diesem Szenario skizziert (vgl. [EHR13]).

1. Ein Produkt wird entsprechend seiner Eigenschaften einer passenden M-Box zugeordnet und in dieser verpackt.
2. Der M-Box wird ein Zielort zugewiesen
3. Die M-Box wird mit anderen M-Boxen gleicher Größe und gleichen Zielortes zu einem Verbund von 1200x800x2400mm (im folgenden UL genannt) gestapelt.
4. Verladung der UL auf einen LKW
5. Transport zu einem PI Hub
6. Übergabe auf einen anderen LKW
7. Transport zum Ziel oder weiteren PI Hub
8. Entladung der UL am Zielort
9. Entnahme der einzelnen M-Boxen von der UL

*Ad 1.)*

Auf Basis der in der Distributionslogistik üblichen Ladehilfs – und Transportmittel wie zum Beispiel Europlatte, Container, LKW usw. haben Meller, Lin und Ellis in [MLE12] die Maße der zu entwickelnden M-Boxen definiert um einen höchstmöglichen Grad der Modularität zu ermöglichen. Die Variantenkombination der identifizierten und in Abbildung 2-2 dargestellten Maße ergibt 440 M-Boxen unterschiedlicher Größe. (vgl. [EHR13]).

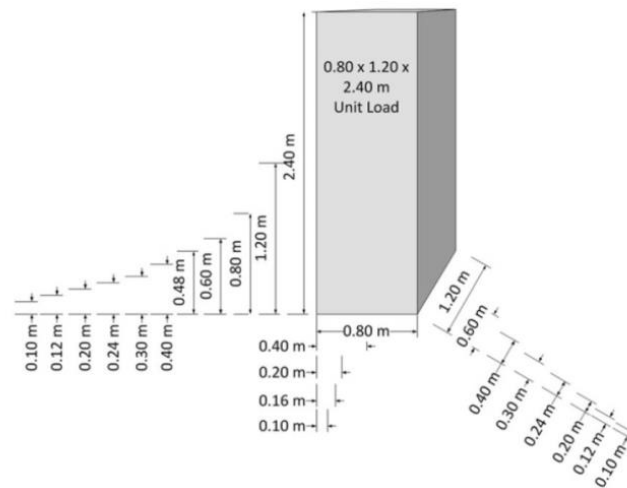


Abbildung 2-2: Modulare Außenmaße der M-Boxen [EHR13]

Die definierten M-Box Größen ergänzen sich gegenseitig auf das Maß der Plattform, um somit eine Modularität zu gewährleisten und einen höchst möglichen Raumnutzungsgrad zu erreichen. Abbildung 2-3 zeigt die Modularität der einzelnen M-Box Dimensionen.

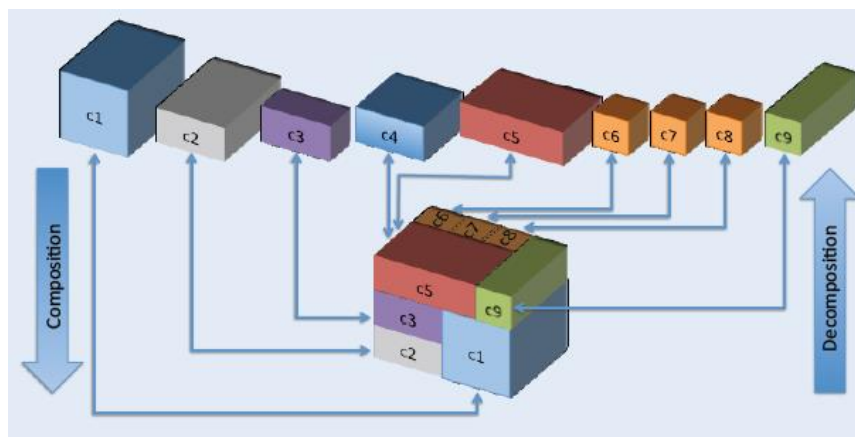


Abbildung 2-3: Modularität der M-Boxen [MOD14]

Als Empfehlung für den Bau einer Prototypenserie wurden 5 Behältergrößen als relevant identifiziert, welche in dieser Arbeit primär behandelt werden. Abbildung 2-4 zeigt diese Boxgrößen und deren Grundfläche. Aus der Abbildung geht auch die Modularität hervor mit der sich die 5 Boxtypen auf das Palettenmaß der Euro Palette ergänzen (vgl. [LJE14])

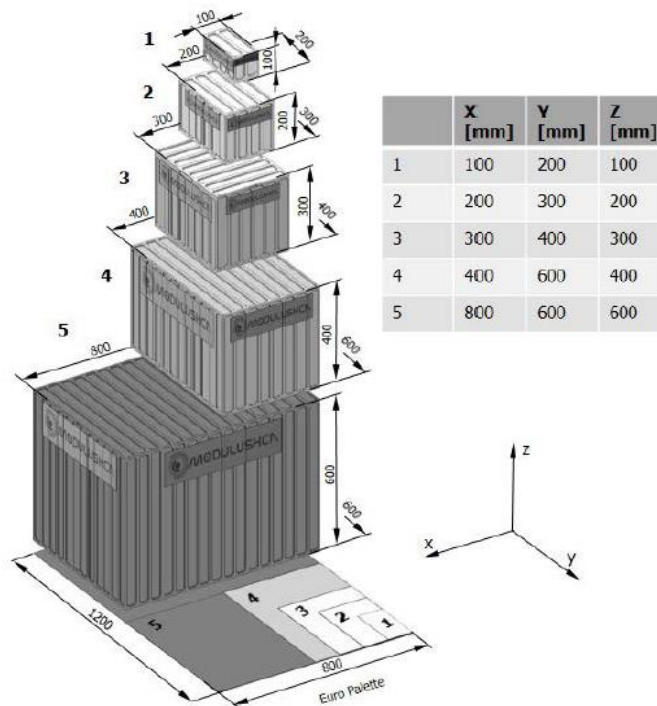


Abbildung 2-4: Behältergrößen für den ersten Prototypenbau [LJE14]

Ad 3.)

Auf Grund der sehr großen Anzahl an M-Boxen im MODULUSHCA Szenario, wird eine UL aus M-Boxen von gleichen Außenabmessungen und gleichem Zielort gebildet. Mittels eines Verriegelungsmechanismus, den jede einzelne M-Box integriert hat, werden die M-Boxen miteinander formschlüssig verbunden, dass die UL selbst eine geschlossene Einheit bildet. Durch die Modularität der einzelnen M-Boxen liegt für jede UL, entsprechend der Abmessungen der verwendeten M-Boxen, das Packmuster bereits fest. Hierbei sei angemerkt, dass diese Prämisse im MODULUSHCA Szenario, als erste Realisierungsvariante des PI, hinlänglich ist. Bei einer weiteren Vertiefung des PI, welche hier nicht Gegenstand der Betrachtung ist, werden Packordnung nach mehreren Gesichtspunkten zu beachten sein und UL somit nicht mehr „sortenrein“ gebildet werden.

Ad 4.)

Jede UL wird einem LKW zugeordnet, der die UL zu ihrem Zielort und/oder zum nächsten PI-Hub bringt.

Ad 5-7.)

Die Route wird gemäß der angestammten Zieldestination festgelegt. Ein wesentliches Merkmal des PI-Szenarios ist, dass ein LKW die UL nicht direkt zum Zielort fährt, sondern zu einem PI-Hub, diese dort ablädt und andere UL für die Retourfahrt auflädt, wie in Abbildung 2-5 dargestellt.

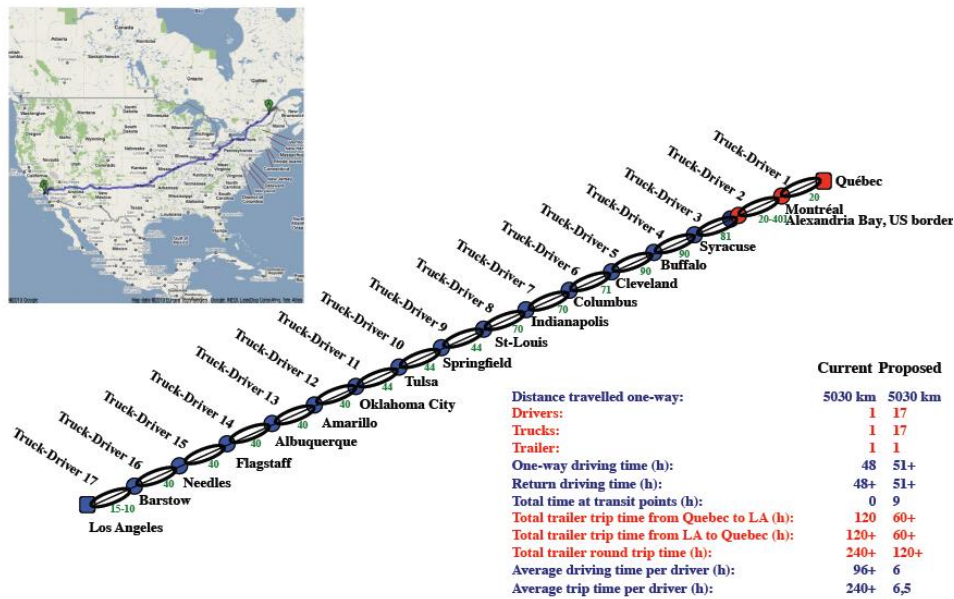


Abbildung 2-5: Routing im PI-Szenario [PII14]

So werden die UL von einem Übergabepunkt zum anderen transportiert und, ob der großen Anzahl, immer wieder mit anderen UL zu einer ganzen LKW Ladung kombiniert um die Ladekapazitäten zu optimieren (vgl. [PII14]).

Durch diese Vorgehensweise wird eine Reihe von Vorteilen generiert:

- Leerfahrten werden de facto unterbunden
- PI-Hubs im Umland von Ballungsräumen ermöglichen es, gebündelt den Bedarf zur Verfügung zu stellen - somit kann die „Letzte Meile“ mit spezialisierten Fahrzeugen oder Zustellungsmodellen erfolgen und der innerstädtische LKW Verkehr minimiert werden
- LKW Fahrer müssen nicht wie jetzt üblich längere Zeit unterwegs sein, sondern können am Abend wieder zu Hause sein

## 2.3 Zielformulierung aus der Problemstellung heraus

Wie in Kapitel 2.2 erörtert, bilden zwei Kernthemen einen wesentlichen Erfolgsfaktor zum Gelingen des Projektes und werden am ITL vorrangig bearbeitet:

1. Abbildung des MODULUSHCA Szenarios anhand von Algorithmen und Überprüfung der Realisierbarkeit
2. Entwicklung und Design der modularen Boxen

Um ein Gelingen des Vorhabens im hohen Maße wahrscheinlich zu machen, müssen die relevanten Prozesse im MODULUSHCA Szenario abgebildet und wesentliche Einflussfaktoren auf die M-Box identifiziert und dargestellt werden. Durch die in Kapitel 2.2 beschriebene Prozessabfolge wurden folgende Prozesse als relevant identifiziert:

1. Das Bilden von UL aus M-Boxen
2. Zuordnen der UL zu LKW
3. Transport der UL

### *Bilden von UL aus M-Boxen (Unit Load Analyse)*

Wie in Kapitel 2.2 erörtert, werden im MODULUSHCA Szenario UL aus M-Boxen mit gleichen Außenabmessungen, gleichem Inhalt und gleichem Zielort gebildet. Durch die Modularität der M-Boxen ist somit für jede UL das Packmuster bereits definiert. Diese Packmuster sollen unter anderem in PTC CREO® automatisiert darstellbar sein, um die Packmuster in einem 3D-CAD-Modell integrieren zu können.

Bei einem Einsatz in Realität müssen aber UL und M-Boxen sowohl Normen und Vorschriften genügen als auch den gegebenen Rahmenbedingungen bei täglicher Verwendung im Distributionsbetrieb angepasst sein. Des Weiteren muss die Gewährleistung der Standsicherheit einer UL für einen sicheren Betrieb immer gesichert sein. Resultierend aus diesen Anforderungen lassen sich drei Ziele formulieren:

1. Ableitung technischer Anforderungen an die M-Boxen aus bindenden Normen und Vorschriften
2. Überprüfung der Standsicherheiten von UL aus jeder möglichen M-Box Kombination und Ableitung von technischen Anforderungen an die M-Box
3. Automatisierte Darstellung von Packmustern in PTC CREO®

Die Abbildung dieser Ziele wird mittels Berechnungsmethodik und -applikation gelöst, wie in Abbildung 2-6 dargestellt und folgend „Unit Load Analyse“ genannt.

### *Zuordnung der UL zu LKW*

Um die UL verschicken zu können, müssen diese auf LKW platziert werden. Um diesen Prozess abbilden zu können, muss ein Algorithmus entwickelt werden, der die UL in einer Weise auf LKW aufteilt, sodass die Anzahl der LKW ein Minimum wird. Dies dient zum einen dem laufenden Betrieb im realisierten MODULUSHCA-Szenario und zum anderen kann somit die Anzahl der benötigten LKW ermittelt werden, weitere Szenarien simuliert und gemäß den formulierten Zielen des MODULUSHCA Szenarios in Kapitel 2.2 ein „Proof of Concept“ erstellt werden.

Folglich kann als Entwicklungsziel formuliert werden:

1. Entwicklung eines Zuordnungs-Algorithmus für UL sodass die Anzahl der benötigten LKW ein Minimum wird.

Die Berechnungsmethodik, welche dieses Ziel verfolgt, wird im Folgenden „Bin Packing Algorithmus“ genannt und dessen Einbettung in die Programmumgebung bzw. softwaretechnische Umsetzung „Unit Load Packer“, gemäß Abbildung 2-6.

### *Transport der UL*

Nachdem die UL an LKW zugeordnet wurden, müssen diese gemäß den technischen Restriktionen welche ein LKW zu erfüllen hat (zulässige Achs- und Sattellasten, Schwerpunkt usw.) am LKW platziert werden, um die theoretischen Ergebnisse der Zuordnung auch realisierbar zu machen. Durch die große Anzahl an UL soll dieser Prozess ebenso automatisiert, basierend auf einem Algorithmus, stattfinden. Die Zielformulierung für diesen Prozess lautet:

1. Entwicklung eines Anordnungs-Algorithmus um UL gemäß normativer und technischer Restriktionen auf einem LKW anzuordnen

Jener Algorithmus, der die UL am LKW anordnet, wird „Balanced Trailer Load Algorithmus“ genannt und in eine Programmstruktur implementiert, welche „Balanced Trailer Load Applikation“ bezeichnet wird, wie aus Abbildung 2-6 hervorgeht.

Diese drei Prozesse sind miteinander gekoppelt und sollen somit eine realisierbare Lösung des MODULUSCA Szenarios unter realen Bedingungen ermöglichen.

Abbildung 2-6 zeigt den oben beschriebenen, zu entwickelnden Prozessablauf mit den benötigten Eingangsgrößen und Zielen der einzelnen Prozesse. Eine Legende zur verwendeten Prozessnotation ist in Abbildung 8-1 dargestellt.

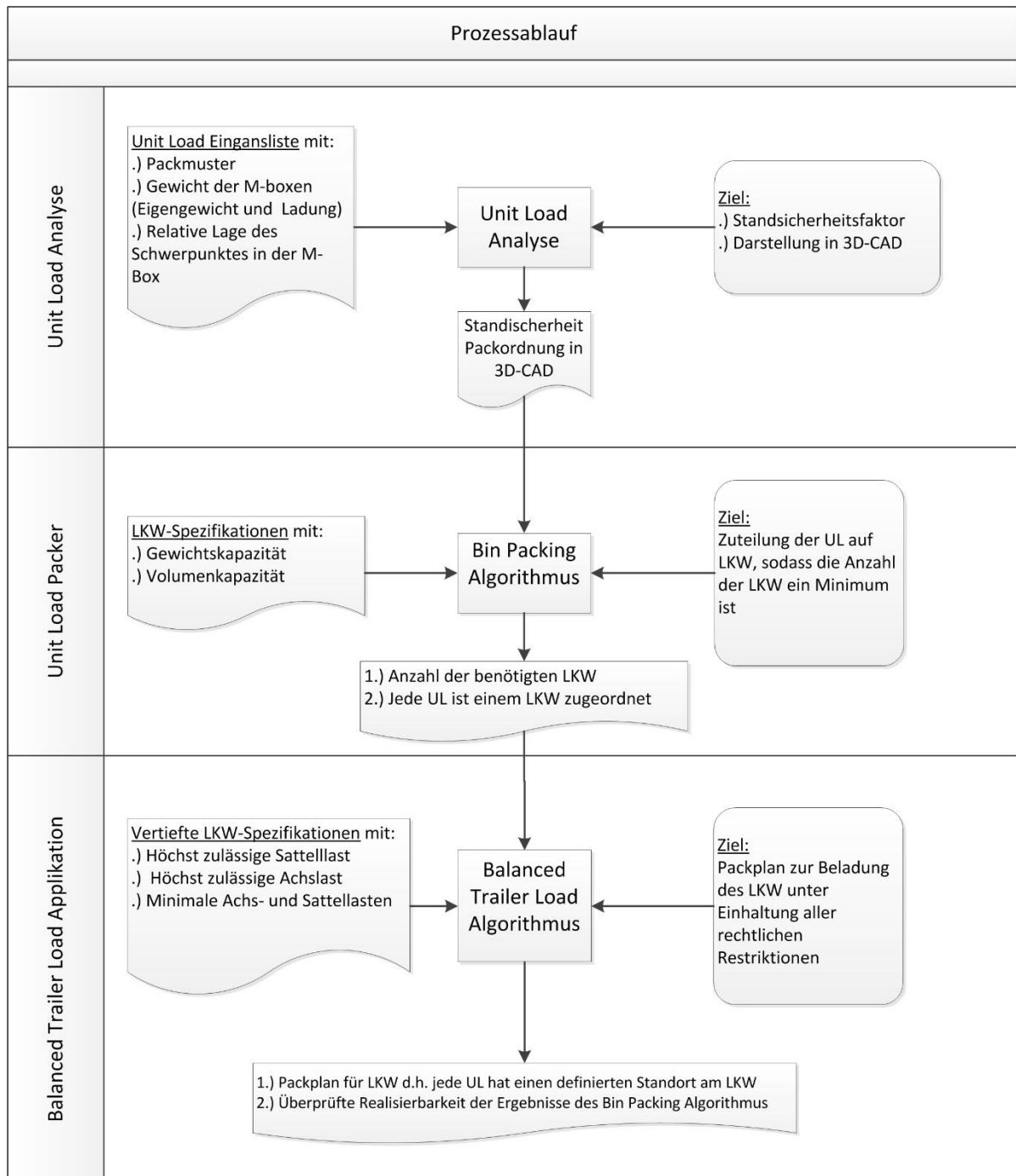


Abbildung 2-6: Prozessablauf

Da wie bereits oben erwähnt die einzelnen Prozesse miteinander gekoppelt sind und die Ergebnisse übergeben werden, orientiert sich auch die Gliederung der folgenden Abschnitte an der in der in Abbildung 2-6 dargestellten Reihenfolge.



## 3 Unit Load Analyse

Dieses Kapitel dokumentiert die methodische Entwicklung der „Unit Load Analyse“. Beginnend bei der Zielformulierung und den abgeleiteten Rahmenbedingungen über Umsetzung bis hin zur Darstellung der Ergebnisse und deren Interpretation.

### 3.1 Entwicklung von Maßnahmen

#### 3.1.1 Zielformulierung

Ziel der Unit Load Analyse ist es, einen bereitgestellten Packplan von M-Boxen automatisiert im 3D-CAD Packet PTC CREO® 1.0 darzustellen und die Standsicherheit der UL zu überprüfen. Die Berechnungsmethodik und deren programmierte Umsetzung soll in der Lage sein über eine universelle Schnittstelle, welche Information über die Lage und Größe einzelner M-Boxen in einem gestapelten Verbund übergibt, die

- Standsicherheit zu berechnen und die
- Packordnung im Baugruppenmodus von PTC CREO® 1.0 automatisiert darzustellen.

Die einzelnen Arbeitsschritte, die diese Berechnungsmethodik- und Applikation zu erfüllen hat sind:

1. Einlesen der Information über
  - a. Art der verwendeten M-Box (Größe, Masse, rel. Schwerpunkt)
  - b. Lage der M-Box im gestapelten Verbund
2. Berechnung der Standsicherheit
3. Erzeugen der gewünschten M-Boxen
4. Zusammenbau der einzelnen M-Boxen gemäß des Packplans und Darstellung im Baugruppenmodus

Ein ergänzendes Ziel der Unit Load Analyse ist durch die Berechnung der Standsicherheit und der Darstellung der M-Boxen im gestapelten Verbund, Anforderungen an das Design des M-Box Prototypen ableiten zu können. Um eine möglichst hohe Variantenvielfalt überprüfen zu können, soll dieser Prozess automatisiert in einem Software-Tool abgebildet werden.

#### 3.1.2 Rahmenbedingungen

Die Berechnung der Standsicherheit wird vom Hauptverband der gewerblichen Berufsgenossenschaften geregelt und muss genauen Vorschriften genügen (vgl. [BRG06]). Jene Vorschriften, welche bei der Berechnung der Standsicherheit tragend sind, werden im folgenden Abschnitt bezugnehmend auf [BRG06] dargestellt. Abbildung 3-1 zeigt einen Stapel aus einzelnen Lagergeräten und die zur Berechnung relevanten Rechengrößen. Diese sind in Tabelle 3-1 zusammengefasst:

Tabelle 3-1: Rechengrößen zur Berechnung der Standsicherheit [BRG06]

Bezeichnung	Beschreibung	Vorgegebene Größe
$Q_G$	Eigengewicht des Lagergerätes	-
$Q$	Nutzlast des Lagergerätes	-
$Q_s$	Eigengewicht + Nutzlast ( $Q_G + Q$ )	-
$H$	Horizontalkraft	1/50 der Gewichtskraft aus $Q_s$
$h_i$	Höhe des einzelnen Lagergeräts	-
$h$	Gesamthöhe des Stapels	-
$H_z$	Zusätzliche Horizontalkraft	Mind. 150N

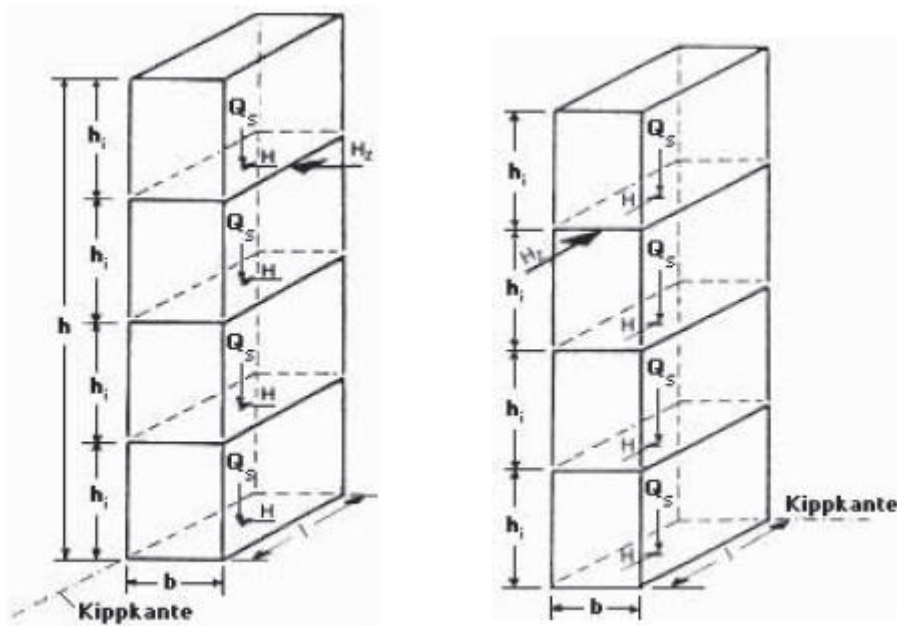


Abbildung 3-1: Darstellung der relevanten Rechengrößen zur Berechnung der Standsicherheit [BRG06]

Daraus resultiert folgende Formel für die Berechnung der Standsicherheit:

$M_{St}$  = Standmoment

$M_k$  = Kippmoment

$n$  = Anzahl der Lagerhilfsmittel im Verbund

$$\frac{M_{St}}{M_k} \geq S \quad \text{Gl. 3-1}$$

$$\frac{M_{St}}{M_k} = \frac{b}{hi} \frac{n \cdot G_s}{(n-1) \cdot (2 \cdot H_z + n \cdot H)} \quad \text{Gl. 3-2}$$

Die Regel hält fest, dass vor allem die Berechnung der Standsicherheit bei leeren Lagergeräten wichtig ist, da wie aus Abbildung 3-1 hervorgeht, die Nutzlast des Verbundes sinkt und somit die Standsicherheit. Des Weiteren müssen folgende Kriterien erfüllt sein (vgl. [BRG06]):

- $v > 2$
- der Stapelverbund muss mindestens die doppelte maximale Nutzlast der Lagergeräte standhalten

### 3.1.3 Methoden

Um automatisiert M-Boxen erzeugen und in gewünschter Lage zusammenbauen zu können und die Standsicherheit gemäß den durch Nomenklatur bestimmten Rahmenbedingungen berechnen zu können, müssen folgende Teilaufgaben in der bezeichneten Reihenfolge erfüllt werden.

1. Einlesen der Information über Art der M-Box, Masse, relative Lage des Schwerpunktes der M-Box und Position im Packmuster
2. Berechnung der Standsicherheit
3. Generierung der einzelnen Boxen
4. Zusammenbau der Boxen in gewünschter Lage

Die Berechnung der Standsicherheit erfolgt in der Programmumgebung, in der auch die Darstellung in PTC CREO® gesteuert wird. Der automatisierte Zusammenbau des Packmusters der M-Boxen muss mittels einer geeigneten Schnittstelle zu PTC CREO realisiert werden.

#### **Berechnung der Standsicherheit:**

Die Berechnung der Standsicherheit einer UL muss unter Berücksichtigung der technischen Aspekte der UL und der M-Boxen geschehen. Wie in Kapitel 2.2 definiert, werden die einzelnen M-Boxen im Stapelverbund durch einen Verriegelungsmechanismus miteinander formschlüssig zu einer Einheit verbunden. Diese Verriegelung sichert, dass die relative Lage der M-Boxen untereinander fixiert ist und somit nicht durch Verrutschen verändert werden kann. Durch diesen Verriegelungsmechanismus kann die Berechnung der Standsicherheit aus [BRG06] zur der in Abbildung 3-2 gezeigten Formel vereinfacht werden.

Folgende Modifizierungen zur Berechnung der Standsicherheit wurden durchgeführt:

- Die Horizontalkräfte, welche an jeder Stapellage angreifen müssen (bezeichnet als H in Tabelle 3-1) können gänzlich vernachlässigt werden da die Lage der M-Boxen im Verbund mittels Formschluss fixiert ist.
- Die Horizontalkraft (bezeichnet als  $H_z$  in Tabelle 3-1) greift bei 2/3 der UL-Höhe an (1.800 mm).
- Die Horizontalkraft ist auf keinen Wert fixiert sondern kann variiert werden um Windlasten, Staplerkollisionen usw. simulieren zu können. Die minimale Horizontalkraft wird in [BRG06] definiert.

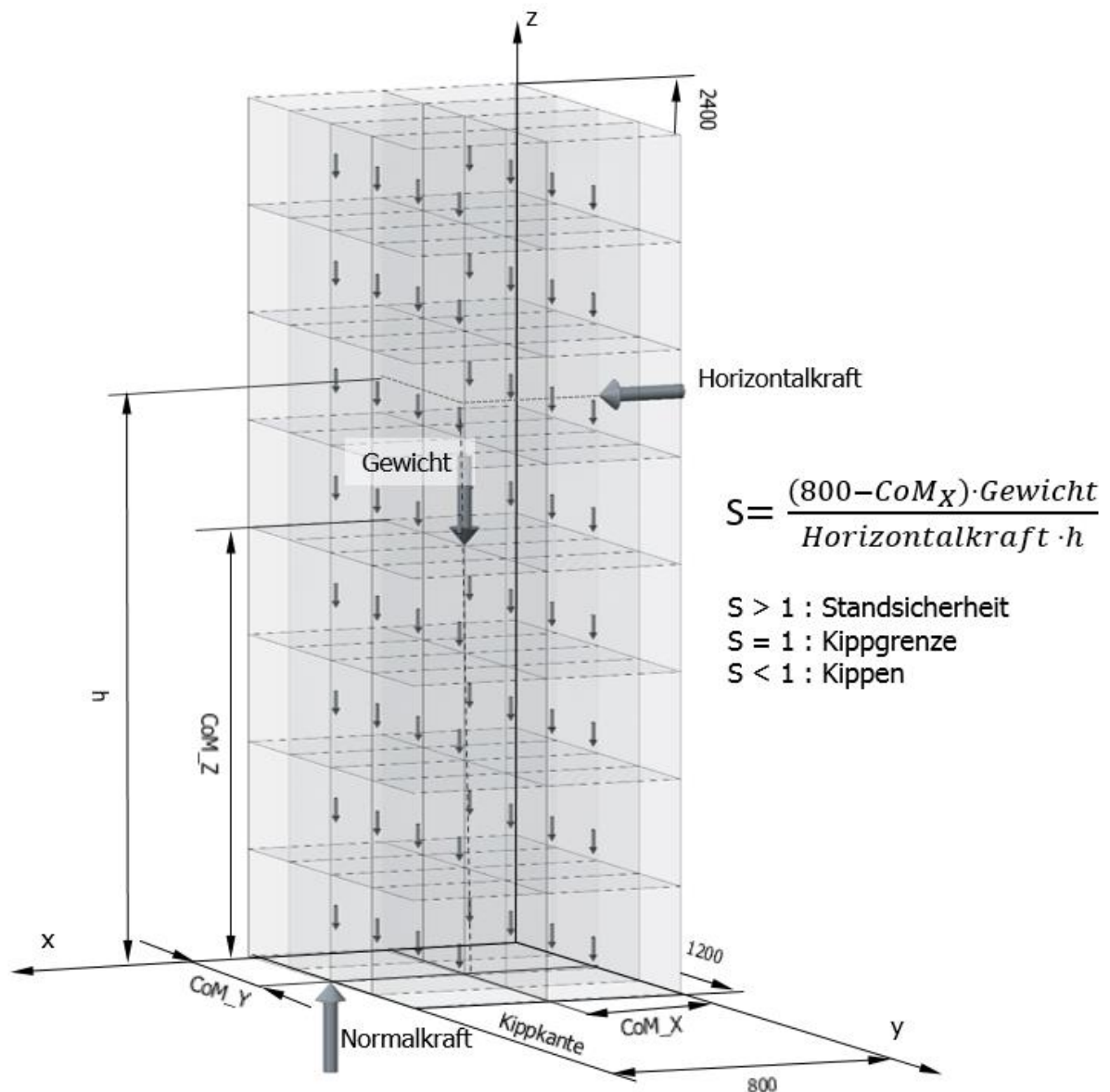


Abbildung 3-2: Standsicherheitsberechnung einer Unit Load (vgl. [LJE14])

### Darstellung der Packordnung in PTC CREO®:

#### **Schnittstellen zu PTC CREO® 1.0:**

Die Schnittstellen zu PTC CREO® ermöglichen die Übergabe der Information über Art und Anzahl der Boxen sowie deren Lage.

*Excel-Analyse:*

Die Excel-Analyse erlaubt es, Parameter auf verschiedenen Ebenen des Konstruktionsmodus wie Skizze, Teil, und Baugruppe, zu bestimmen und somit die Information über die Packordnung an PTC CREO® zu übergeben.

## Vorteile:

- Bereits in PTC CREO® integrierte, einfache Funktion
- Es können sowohl Parameter an PTC CREO® übergeben als auch eingelesen werden
- Microsoft Excel® steht als weit verbreitetes Programm auf beinahe jedem Arbeitsplatz zur Verfügung
- Mittels Visual Basic®-Applikation können innerhalb von Microsoft Excel® Daten verarbeitet und vorbereitet werden

## Nachteile:

- Direkte Steuerung von PTC CREO® ist nicht möglich
- PTC CREO® referenziert absolut auf die verwendeten Excel-Datei

*Visual Basic API:*

Die Visual Basic API (VB-API) ermöglicht die komplette Steuerung von PTC CREO® und bietet trotzdem alle Vorteile der Programmiersprache VB.net. VP-API kommuniziert zu PTC CREO® via COM Schnittstelle (vgl. [PIC14], [PTC10] und [PTC14])

## Vorteile:

- Steuerung aller Funktionen in PTC CREO® auf allen Konstruktionsebenen
- Dynamisches Referenzieren und somit beliebige Weiterverwendung der generierten Teile, Baugruppen und Zeichnungen

## Nachteile:

- Ist nicht in der Standardinstallation von PTC CREO® vorhanden
- Aufwändige Handhabung mit hoher Einarbeitungszeit
- Schlechte Produktdokumentation

**Automatisierte Teile-Generierung**

Folgende, in PTC CREO® integrierte Funktionen ermöglichen automatisiert Teile zu erzeugen bzw. von einem Eltern-Teil abzuleiten.

*Familientabelle:*

Mittels Familientabellen können Varianten von Baugruppen und Teile, sogenannte „Kind-Teile“ erzeugt werden. Hierfür muss zuvor ein Eltern-Teil erzeugt und parametrisiert werden. Wenn eine Ableitung davon benötigt wird, müssen die geforderten und zuvor definierten Parameter eingegeben werden und PTC CREO® erzeugt die gewünschte Variante. Ein typischer Anwendungsfall für die Verwendung einer Familientabelle ist die Erzeugung von Schrauben (vgl. [PTC14]).

## Vorteile:

- Standardfunktion von PTC CREO®
- Vielfach in Verwendung und dokumentiert

Nachteile:

- Kind-Teile werden nicht automatisiert in eine Baugruppe eingebaut sondern stehen als Teil zur Verfügung

### Zusammenfügen zu Baugruppen

Einzelne Teile können in PTC CREO® zu Baugruppen zusammengebaut werden. Die geometrischen und/oder kinematischen Beziehungen der einzelnen Teile untereinander oder zu einem Koordinatensystem in der Baugruppe müssen dafür festgelegt werden. Beim Zusammenbau vieler verschiedener Teile mit unterschiedlichsten Beziehungen zueinander müssen diese Beziehungen vom User für jedes einzelne Teil festgelegt werden. Der Einbau vieler Teile in ähnlicher Lage kann mittels „Muster“ erfolgen.

*Muster:*

Beim gemusterten Einbau von Teilen müssen deren relative Lage zueinander und Referenzflächen- oder Achsen definiert werden. PTC CREO® baut dann das gewünschte Teil so oft an jenen Positionen ein, die durch das Muster bestimmt werden (vgl. [PTC14]).

Vorteile:

- Integrierte Standardfunktion
- Einfach zu verwenden

Nachteil:

- Nur ein Teil kann mittels eines Musters automatisiert eingebaut werden
- Referenzobjekte müssen an definierten Stellen vorhanden sein

### Übersicht

Die vorhandenen Methoden eignen sich wie in Tabelle 3-2 dargestellt für die Erfüllung der jeweiligen Teilprozesse.

**Tabelle 3-2: Eignung der Methoden zur Erfüllung der Teilprozesse - Übersicht**

Prozess	Methode			
	Excel-Analyse	VB-API	Muster	Familiens-tabelle
<b>Schnittstelle (dynamische Steuerung)</b>	Ja	Ja	Nein	Nein
<b>Automat. Bauteil-Generierung</b>	Nein	Ja	Nein	Ja
<b>Automat. Bauteil-Zusammenbau</b>	Nein	Ja	Ja	Nein
<b>Implementierung von Berechnungsmethodik</b>	Ja	Ja	Nein	Nein
<b>Erweiterbarkeit auf PI-Rahmenbedingungen</b>	Nein	Ja	Nein	Nein

Aus Tabelle 3-1 geht hervor, dass nur die VB-API Methode geeignet ist, sämtliche Teilprozesse, welche zur Erreichung der formulierten Ziele notwendig sind, automatisiert zu erfüllen. Deswegen ist die automatisierte Darstellung der Packordnung in PTC CREO® via VB API realisiert und in folgenden Kapiteln die Entwicklung der Steuerung dokumentiert.

## 3.2 Umsetzung

Basierend auf den in Kapiteln 3.1.2 und 3.1.3 dargestellten Normungen, Rahmenbedingungen und Möglichkeiten in PTC CREO® wird in den folgenden Kapiteln der Aufbau und die Funktionsweise der Unit Load Analyse dargestellt.

### 3.2.1 Allgemeine Übersicht

Die beiden Aufgaben der Unit Load Analyse werden in einem VB.net Programm in folgenden drei Schritten ausgeführt.

1. Einlesen der Information über die UL
2. Berechnung der Standsicherheit
3. Steuerung von PTC CREO® mittels VB-API Schnittstelle zur automatischen Erzeugung der M-Boxen und Zusammenbau der UL

#### Einlesen der Information über die UL

Durch die Verwendung von VB.net stehen sämtliche Mechanismen zur Datenverarbeitung zur Verfügung. Um das Einlesen der Information über die darzustellende Packordnung universell zu gestalten und eine möglichst hohe Unabhängigkeit von der Aktualität von Datenformaten zu erreichen, muss die Packordnung Mittels .txt –Dokuments zur Verfügung gestellt werden. Dieses Dokument muss zeilenweise folgende Information für jede M-Box im UL Verbund enthalten. Jede Zeile stellt hierbei eine M-Box im Packmuster dar:

- Spalte 1: X-Koordinate des Positionierungspunktes
- Spalte 2: Y-Koordinate des Positionierungspunktes
- Spalte 3: Z-Koordinate des Positionierungspunktes
- Spalte 4: Geometrische Erstreckung der M-Box in X-Richtung
- Spalte 5: Geometrische Erstreckung der M-Box in Y-Richtung
- Spalte 6: Geometrische Erstreckung der M-Box in Z-Richtung
- Spalte 7: Relative Lage des Schwerpunktes in der M-Box in X-Richtung
- Spalte 8: Relative Lage des Schwerpunktes in der M-Box in Y-Richtung
- Spalte 9: Relative Lage des Schwerpunktes in der M-Box in Z-Richtung
- Spalte 10: Mittlere Dichte der M-Box inklusive Inhalt

Tabelle 3-3 zeigt exemplarisch, wie das Dokument aufgebaut ist. Jede Zeile beschreibt eine M-Box, die erzeugt und eingebaut werden muss.

**Tabelle 3-3: Beispielhafte Darstellung der Packordnung**

Box Nr.	Pos X [mm]	Pos Y [mm]	Pos Z [mm]	X [mm]	Y [mm]	Z [mm]	CoM_ X [-]	CoM_ Y [-]	CoM_ Z [-]	$\rho$ [kg/m <sup>3</sup> ]
1	0	0	0	400	400	600	0,5	0,5	0,5	450
2	400	0	0	200	200	400	0,5	0,5	0,2	200

#### Berechnung der Standsicherheit

Die Berechnung der Standsicherheit erfolgt gemäß den in Kapitel 3.1.2 dargestellten Normen und zulässigen Vereinfachungen und erfolgt in der Programmumgebung unabhängig von der Implementierung in PTC CREO®. Wie in Abbildung 3-2 gezeigt, berechnet das Programm den Standsicherheitsfaktor S über die kürzeste Kante. Die

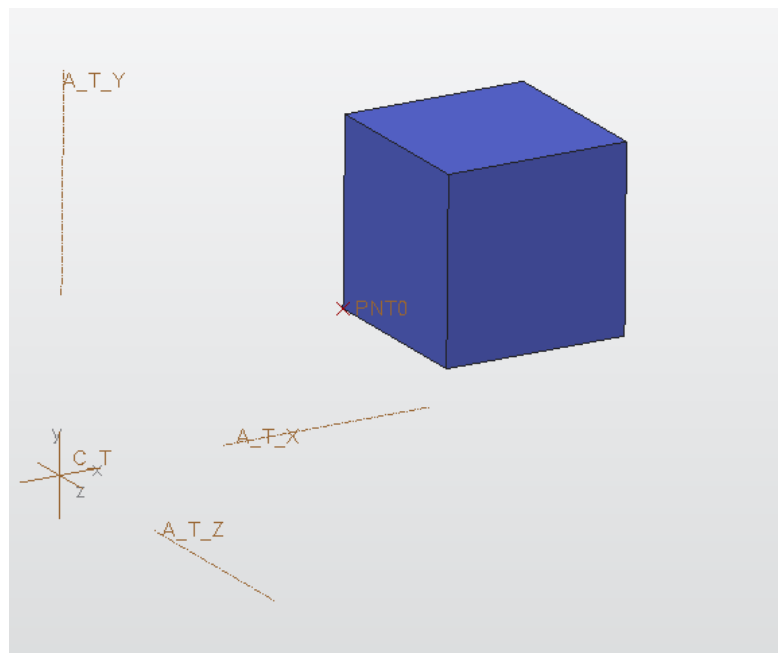
Horizontalkraft, auf welche die Standsicherheit bezogen wird, kann individuell über die Eingabemaske definiert werden.

### Darstellung in PTC CREO® 1.0

#### *Orientierung und Koordinaten*

Um die Positionierung einer M-Box im 3-dimensionalen Raum zu ermöglichen wurde die Definition des Positionierungs-Punkts eingeführt. Dieser Punkt bezeichnet die Position der linken, unteren Ecke einer M-Box in der Packordnung. Die Größe der zu verwendeten M-Box und auch deren Orientierung ist mit den Abmaßen X,Y,Z determiniert.

Abbildung 3-3 zeigt das verwendete Koordinatensystem, den Positionierungs-Punkt (PNT0) sowie die beispielhafte Darstellung der M-Box.



**Abbildung 3-3: Positionierungs-Punkt und Koordinatensystem**

#### *Erzeugung der einzelnen M-Boxen*

Um die einzelnen M-Boxen zu erzeugen, wird ein Eltern-Teil, im verwendeten Programm „MODBOX\_FATHER.prt“ genannt, geöffnet, die nötigen Parameter editiert und mit dem Titel „C\_PosX\_PosY\_Pos\_Z\_X\_Y\_Z“ gespeichert. Wobei PosX, PosY und PosZ als Variablen für den Positionierungs-Punkt der jeweiligen Box stehen und X, Y, Z die Erstreckung der M-Box in die jeweilige Koordinatenrichtung repräsentieren.

Das bedeutet, dass nicht jede M-Box neu erzeugt wird, sondern, gekennzeichnet durch geänderte Parameter, von einem Eltern-Teil abgeleitet wird, wie in Abbildung 3-4 dargestellt.

#### *Aufbau des Elternteiles der M-Box:*

Das Elternteil der M-Box besitzt 6 Parameter: 3 für die Koordinaten des Positionierungs-Punktes und 3 für die Erstreckung in X, Y, Z-Richtung der M-Box. Das bedeutet, dass das abgeleitete Kind-Teil nicht im Ursprung des Standard-Koordinatensystems platziert ist, sondern in der Position, welche es im Verbund einnimmt. Abbildung 3-3 zeigt exemplarisch ein Kind-Teil einer M-Box und dessen Lage.



### *Automatischer Zusammenbau*

Auf Grund der Tatsache, dass die Kind-Teile der M-Box bereits in deren entsprechenden Lage im Verbund erstellt wurden, werden die einzelnen M-Boxen in der von PTC CREO® vordefinierten „Standardorientierung“ eingebaut. „Standardorientierung“ bedeutet, dass das Koordinatensystem jeder M-Box mit dem Koordinatensystem der Baugruppe, dem gestapelten Verbund, deckungsgleich ist. Durch den bereits im „Teile-Modus“ definierten Positionierungs-Punkts ist somit die Lage und die Orientierung der einzelnen Boxen bestimmt.

Da das Programm selbst nur die Kind-Teile der M-Box erzeugt, ist das Vorhandensein folgender, in Tabelle 3-4 dargestellten, Dateien im Programmordner zwingend notwendig.

**Tabelle 3-4: Dateien in „Unit Load Analyse“ zum automatisierten Zusammenbau**

<b>Name</b>	<b>Beschreibung</b>
<b>Boxlist.out</b>	Text-Datei mit der Information über die Packordnung
<b>Assembly.asm</b>	Leere Baugruppe, in welche die M-Boxen eingebaut werden
<b>Modbox_Father.prt</b>	Parametrisch aufgebautes Eltern-Teil der M-Box

Einzelne Functions und Subs mit welchen die programmier-technische Umsetzung erfolgt, sind im Anhang in Kapitel 8.2.2 detailliert dargestellt.

### *Ablauf*

Der für die automatisierte Darstellung notwendige Prozessablauf für die gewählte Realisierungsmethode ist in Abbildung 3-4 dargestellt und im folgenden Absatz beschrieben. Zunächst muss die Eingangsdatei „boxlist.out“ eingelesen und in Eingangsdaten verarbeitet werden. Danach erfolgt die Berechnung der Standsicherheit in der VB.net Programmumgebung. Im darauf folgenden Schritt wird die Verbindung zu PTC CREO® hergestellt und die Baugruppe „Assembly.asm“, die im Programmordner vorhanden ist, geöffnet. Unabhängig von der Baugruppe öffnet das Programm das Bauteil „Modbox\_Father.prt“, setzt die Parameter, gemäß den Eingangsdaten, und definiert somit Größe und Lage der M-Box. Um das Bauteil entsprechend in „Assembly.asm“ einbauen zu können, wird der Bauteilname abgeändert auf „Child\_XY“ (wie oben beschrieben), in den Projektordner gespeichert und in „Assembly.asm“ eingebaut. Das Programm wiederholt diese Schritte bis alle M-Boxen erzeugt wurden. Wenn die gesamte Packordnung dargestellt ist, regeneriert das Programm „Assembly.asm“ und schließt die Verbindung.

**Flow Chart**

Abbildung 3-4 zeigt die einzelnen Arbeitsschritte des Programms. Eine Legende zur verwendeten Prozessnotation ist in Abbildung 8-1 dargestellt.

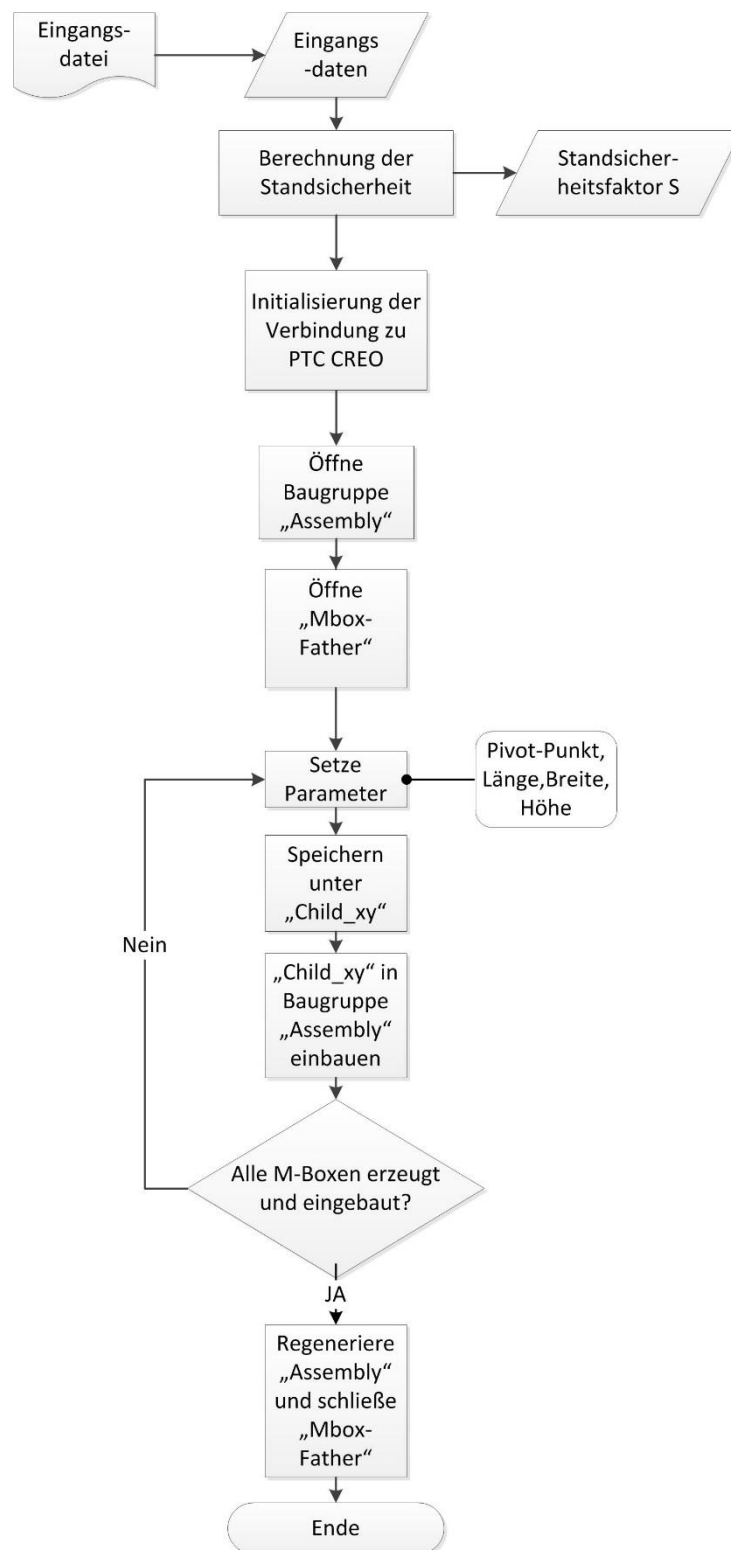


Abbildung 3-4: Flow Chart "Unit Load Analyse"

### 3.2.2 Begründung der Vorgehensweise

Die Berechnung der Standsicherheit wird vom Hauptverband der gewerblichen Berufsgenossenschaften in [BRG06] normiert und lässt keinerlei „Spielraum“ für die Vorgehensweise. Die in Abbildung 3-2 gezeigte normgerechte Berechnungsmethodik wurde mittels VB.net Code implementiert.

In Kapitel 3.1.3 wurden alle in PTC CREO® zur Verfügung gestellten Methoden zur Erfüllung der beschriebenen Teilaufgaben dargestellt. Da keine dieser Funktionen alle Teilaufgaben erfüllen kann, wie aus Tabelle 3-2 hervorgeht, wurde der automatische Zusammenbau mittels VB-API realisiert. Dadurch ist es möglich alle Funktionen in PTC CREO® zu automatisieren und zudem noch sämtliche Funktionen der Programmiersprache VB.net zu verwenden.

Die einzelnen M-Boxen werden in der gewählten Realisierungsvariante nicht neu erzeugt, sondern von einem Eltern-Teil abgeleitet und mit einem neuen Dateinamen gespeichert. Da zum Zeitpunkt der Entstehung dieser Arbeit noch nicht bestimmt ist, wie die Detailgeometrie der einzelnen M-Boxen aussieht, werden beispielhaft rechteckige Profile dargestellt. Durch das Erzeugen der einzelnen M-Boxen von einem Eltern-Teil ist es möglich mit sinnvoller Parametrisierung jede Detailgeometrie, die im Zuge des Entwicklungsprozesses erarbeitet wird, darzustellen.

Der gewählten Vorgehensweise haften folgende Vor- und Nachteile an:

Vorteile:

- Vollständige, automatisierte Erzeugung der einzelnen M-Boxen
- Automatisierter Zusammenbau der M-Boxen
- Parametersteuerung zur Darstellung jeder möglichen M-Box-Geometrie
- Erweiterung der Parametersteuerung durch VB-API einfach möglich
- Universelle Schnittstelle zum Einlesen der Information
- Dynamische Referenzierung und somit getrennte Speicherungsmöglichkeit jeder einzelnen Packordnung

Nachteile:

- Eltern-Teil muss zur Verfügung stehen
- Drei zusätzliche Programmdateien müssen im Projektordner vorhanden sein

### 3.3 Ergebnisse der Vorgehensweise

Die erarbeitete Berechnungsmethodik- und applikation „Unit Load Analyse“ ist buchstäblich als „Werkzeug“ zu gebrauchen, um durch vielseitige, automatisierte Variantenbildung Funktionen und Anforderungen an die M-Box bzw. der daraus gebildeten UL ableiten zu können und die Realisierbarkeit einer Packordnung überprüfen zu können. Diese Vorgehensweise ist im nachfolgenden Abschnitt erklärt.

#### Eingangsdaten

Als Eingangsdaten wurden die für den ersten Prototypen relevanten M-Box Größen verwendet, wie in Kapitel Abbildung 2-4 dargestellt. Aus diesen 5 M-Box Typen, siehe Tabelle 3-5, werden gemäß den Annahmen in Kapitel 2.3 fünf unterschiedliche UL gebildet. Hierbei sei erwähnt, dass diese UL sehr wohl im Gewicht und dem Packmuster differieren können, abhängig von dem geladenen Produkt. Die Außenmaße bleiben aber immer gleich.

Tabelle 3-5: M-Box Größen

Nr.	Länge [mm]	Breite [mm]	Höhe [mm]	Anzahl M-Boxen pro UL [ ]
<b>1</b>	100	200	100	1152
<b>2</b>	200	300	200	192
<b>3</b>	300	400	300	64
<b>4</b>	400	600	400	24
<b>5</b>	800	600	600	8

Des Weiteren resultieren folgende Eingangsdaten aus Normungen (siehe Kapitel 3.1.2) und den Anforderungen aus den MODULUSHCA Projektzielen (siehe Kapitel 2.2).

Tabelle 3-6: Eingangsdaten Unit Load Analyse

Bezeichnung	Wert	Begründung
<b>Minimale Standsicherheit [ ]</b>	2	Resultiert aus [BGR06]
<b>Minimale Horizontalkraft [N]</b>	150	Resultiert aus [BGR06]
<b>Maximales UL-Gewicht [kg]</b>	2.000	Resultiert aus höchstzulässiger Palettenlast
<b>Maximale Ladungsdichte [kg/m<sup>3</sup>]</b>	500	MODULUSHCA Projektszenario

### Ergebnisse

Gemäß der dargestellten Berechnungsmethodik kann unter dem Aspekt der Standsicherheit für die UL Folgendes abgeleitet werden:

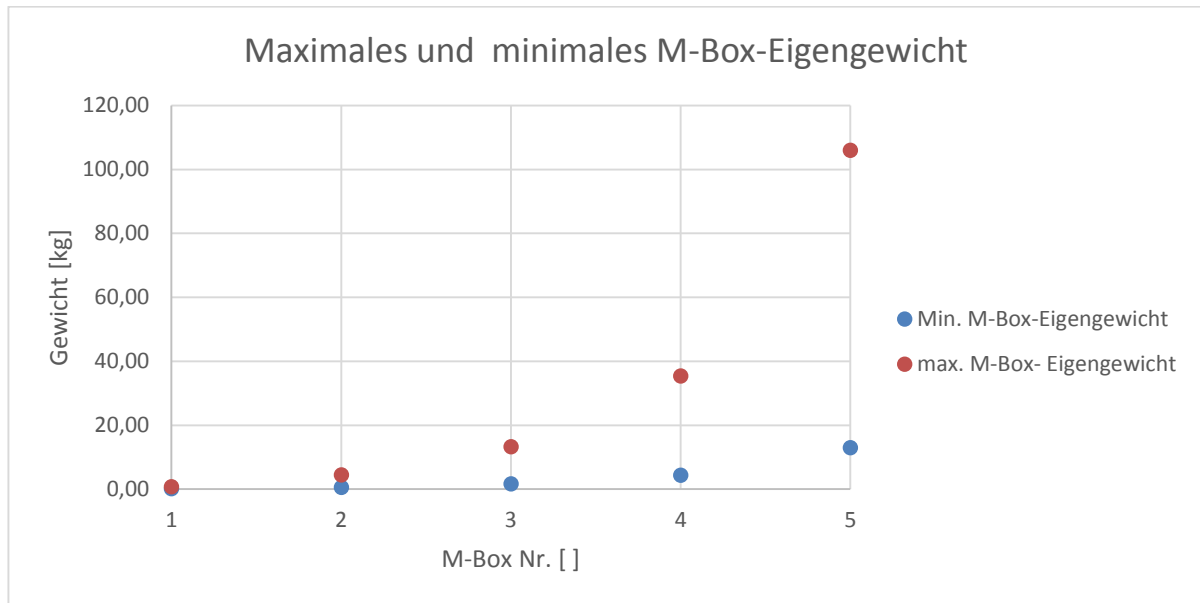
Minimales UL-Gewicht [kg]: 103,3 kg

Daraus resultieren für das Design der M-Boxen folgende in Tabelle 3-7 gezeigten Restriktionen:

Tabelle 3-7: Abgeleitete Ergebnisse für das Design der M-Boxen

Nr.	Minimales M-Box Eigengewicht [kg]	Maximales M-Box Eigengewicht [kg]	Maximales M-Box Gesamtgewicht (Box + Ladung) [kg]
<b>1</b>	0,09	0,37	1,74
<b>2</b>	0,54	4,42	10,42
<b>3</b>	1,61	13,25	31,25
<b>4</b>	4,30	35,33	83,33
<b>5</b>	12,90	106,00	240,00

Abbildung 3-5 verdeutlicht graphisch dargestellt die Ergebnisse für minimales und maximales M-Box Eigengewicht aus Tabelle 3-7.



**Abbildung 3-5: Maximales und minimales M-Box-Eigengewicht**

Die Ergebnisse der Unit Load Analyse zeigen, dass der Faktor Standsicherheit nur Einfluss hat auf das minimale Eigengewicht der M-Boxen. Das maximale M-Box Gesamtgewicht sowie das maximale M-Box Eigengewicht werden von den technischen Restriktionen aus der Projektdefinition von MODULUSHCA abgeleitet.

Als weiteres Ergebnis ist die Darstellung der Packordnungen in PTC CREO® 1.0, nach dem automatisierten Zusammenbau der eingelesenen Packordnung anzuführen.

Abbildung 3-6 und Abbildung 3-7 zeigen exemplarisch die Packordnung für die M-Box Nr.1 und Nr. 2 aus Tabelle 3-5.

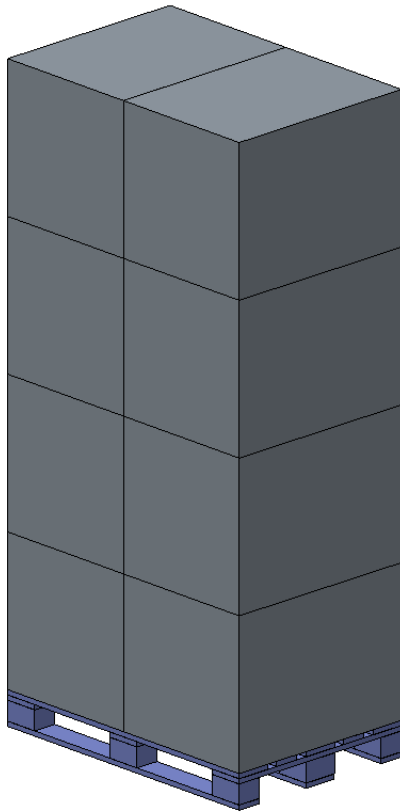


Abbildung 3-6: Packordnung für M-Box Nr. 5 in PTC CREO® 1.0

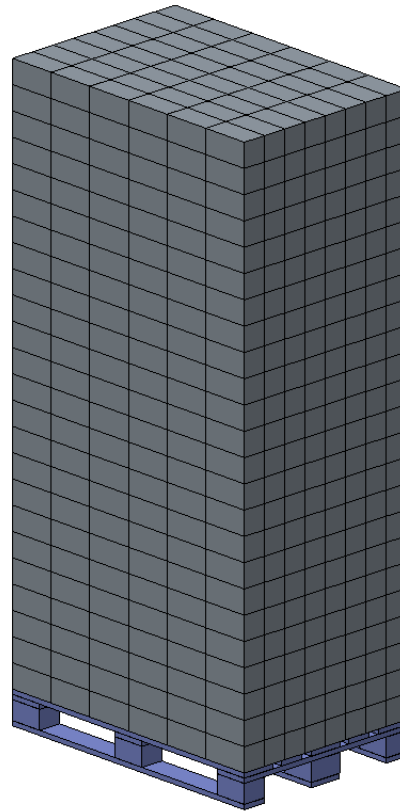


Abbildung 3-7: Packordnung für M-Box Nr. 1 in PTC CREO® 1.0

### 3.4 Bewertung/Beurteilung der Vorgehensweise

Die Unit Load Analyse stellt in der oben beschriebenen Methodik ein Werkzeug dar, das es ermöglicht sämtliche Packordnungen, unter Berücksichtigung des Schwerpunktes eines jeden gepackten Pakets, auf deren Standsicherheit zu untersuchen und mittels VB-API Schnittstelle in PTC CREO® 1.0 darzustellen. Als Ergebnis konnten minimale und maximale Eigengewichte für die M-Boxen abgeleitet werden, ebenso stellt die Berechnungsmethodik und dessen Eingliederung in eine benutzerfreundliche Programmumgebung ein nützliches Werkzeug für den laufenden Betrieb im MODULUSHCA Szenario dar.

Die Umsetzung der automatischen Darstellung der Packordnung in PTC CREO® 1.0 weist die Schwäche auf, dass Initialdateien im Programmordner vorhanden sein müssen, da jedes automatisch erzeugte und eingebaute Paket von einem „Elternteil“ abgeleitet wird. Dies birgt jedoch den Vorteil, dass in späteren Szenarien durch eine ausgeklügelte Parametersteuerung im Elternteil aber auch kompliziertere Geometrien der darzustellenden Boxen verwirklicht werden können.

Letztendlich wurden die in Kapitel 2.3 abgeleiteten Zielformulierungen zur Gänze erfüllt.

## 4 Unit Load Packer

„Unit Load Packer“ ist die Bezeichnung für die Umsetzung des entwickelten Bin Packing-Algorithmus in einer Softwareumgebung. Im folgenden Kapitel wird die Entwicklung des Bin-Packing Algorithmus und dessen Einbettung im Unit Load Packer, basierend auf Zielformulierungen und Rahmenbedingungen dokumentiert sowie Testinstanzen und Ergebnisse dargestellt und interpretiert.

### 4.1 Entwicklung von Maßnahmen

#### 4.1.1 Zielformulierung

Der Unit-Load Packer ist eine Berechnungsmethodik, welche die Aufgabe hat, eine gegebene Menge an UL so auf LKW zu verteilen, dass die Anzahl der LKW ein Minimum wird. An dieser Stelle sei festgehalten, dass es nicht gefordert ist, den Stellplatz der einzelnen UL auf einem LKW zu bestimmen sondern die UL den einzelnen LKW zuzuteilen. Die Erzeugung des Packplans für den LKW, welcher die Lage jeder UL auf dem LKW definiert, wird anschließend in Kapitel 5 behandelt. Gemäß den in Kapitel 2.2 dargestellten Projektzielen und -szenarien können folgende Rahmenbedingungen abgeleitet werden:

- Alle UL haben dieselben geometrischen Maße, können aber unterschiedlich schwer sein
- Alle UL müssen zu einem, gemeinsamen Ort gebracht werden (auf Grund der PI – Idee siehe Abbildung 2-5)
- Alle LKW haben dieselbe Kapazität an Volumen und Nutzlast.
- Der Algorithmus muss für eine große Anzahl an UL eine Lösung in einem angemessenen Zeitraum finden
- Die gefundene Lösung darf das theoretische Minimum<sup>1</sup> um maximal 1 LKW verfehlen

#### 4.1.2 Rahmenbedingungen

Basierend auf den Daten eines Projektpartners, wurden mit Hilfe einer Berechnungsmethodik von Meller, Lin und Ellis die zu verteilende Menge an UL und deren Gewicht bestimmt (vgl. [MLE12]). Diese Daten enthalten die Anzahl und Masse der jeweiligen UL, welcher M-Box Typ (Größe) verwendet wird und welches Produkt darin verpackt ist. Die Datensätze repräsentieren eine durchschnittliche, monatliche Umschlagsmenge eines Distributionszentrums des Unternehmens. Die Anzahl der UL beträgt zwischen 24.000 und 15.000 Stück. Die Verwendung dieser Daten bringt eine Reihe von Vorteilen mit sich:

- Reale Daten aus einem Distributionszentrum und somit reale Gewichtsklassen, -verteilung und Streuung

---

<sup>1</sup> Die „untere Schranke“ wird später genauer behandelt und definiert. An dieser Stelle reicht die intuitive Vorstellung aber aus.

- Durch die genaue Marktkennntnis des Unternehmens variiert der Bedarf an zu versendenden FMCG über das Jahr nur leicht und somit auch die Datensätze
- Großer Planungszeitraum (1 Monat)

Durch diese oben genannten Rahmenbedingungen und der Charakteristika der Datensätze kann als Ziel für den Packalgorithmus folgendes abgeleitet werden:

- Die Berechnungsdauer spielt eine untergeordnete Rolle. Auf Grund des Bedarfes von einem Monat und der hohen Vorhersehbarkeit ist eine Berechnungsdauer von bis zu 12h zulässig
- Auf Grund der hohen Anzahl an UL (24.000 – 15.000 Stück) muss das Ergebnis aber eine sehr hohe Qualität haben und darf maximal die theoretisch kleinste benötigte Menge an LKW um 1 überschreiten.
- Kommerzielle Softwarepakete können für die Lösung nicht verwendet werden. Die Verwendung der Berechnungsmethodik muss auf jedem herkömmlichen Büro-Arbeitsplatz funktionieren können. Die Annahme, dass aber Standardsoftware wie Microsoft Windows® und Office® vorhanden ist, ist zulässig.

### 4.1.3 Umfeld

#### Zuordnung der Problemstellung

Die beschriebene Aufgabenstellung wird in der Literatur als „2D Vector Bin Packing“ oder auch als „Cardinality constrained Bin Packing“ beschrieben. Diese sind ein Sonderfall des regulären 2D Bin Packing Problems. Beim regulären 2D- Bin Packing Problems müssen 2-dimensionale Elemente (Quadrate, Rechtecke usw.) auf einer Grundfläche angeordnet werden. Beim hier vorliegenden Spezialfall müssen Elemente (UL) in einem Behältnis (LKW) zugeordnet werden, welches 2 Restriktionen unterliegt. Zum einen kann nur eine bestimmte Gewichtslast und/oder eine bestimmte Menge an Elementen aufgenommen werden. Das Vector Bin Packing Problem ist in der Behandlung etwas einfacher als das reguläre Bin-Packing Problem da in der zweiten Dimension nur gezählt werden muss.

Das hier vorliegende Problem ist bewiesen NP-vollständig, was bedeutet, dass es Nicht-deterministisch und zeitlich Polynomiell ist. (vgl. [KP99])

Dabei bedeutet „nicht deterministisch“ dass nicht exakt eine Berechnung zu einer bestimmten Problemstellung durchlaufen werden kann, sondern dass es bei gleicher Eingangsgröße mehrere „Wege“ gibt den jeweils darauf folgenden, nächsten Zustand zu erreichen.

Die Beschreibung der Zeitkomplexität definiert, wie sich der Zeitaufwand bei der Berechnung eines optimalen Algorithmus verändert, wenn die Eingabemenge wächst. Bei der Beschreibung von Komplexitätsklassen bedeutet „zeitlich polynomiell“, dass der Zeitbedarf für das Lösen der Aufgabe mit der Eingabemenge nicht stärker zunimmt als mit einer Polynomfunktion beschrieben werden kann. NP-vollständige Probleme gehören gemäß der Komplexitätstheorie zu den am schwersten lösebaren Problemen, was bedeutet dass sie mit hoher Wahrscheinlichkeit<sup>2</sup> nicht effizient lösbar sind (vgl. [WEG03]).

Dadurch wächst der Zeitbedarf mit der Anzahl der Eingabemenge beträchtlich und häufig sind einfache Aufgabenstellungen schon nicht mehr lösbar. Folgendes Zahlenbeispiel soll das verdeutlichen:

---

<sup>2</sup> Anmerkung: Mathematischer, exakter Beweis wurde noch keiner gefunden und ist eines der sogenannten „Milleniumprobleme“



Eine 4-elementige Menge kann in 24 verschiedenen Varianten angeordnet werden, eine 8-elementige in 3.628.800 und eine 33-elementige Menge in  $8,633 \cdot 10^{36}$  Varianten.

Eine wesentliche Messgröße zur Darstellung der Effizienz von Vector Packing Algorithmen ist neben der benötigten Berechnungszeit die „Untere Schranke“, welche in der englischen Literatur als „Lower Bound“ bezeichnet wird.

### Untere Schranken - Lower Bounds (LB)

Caprara und Toth bezeichnen die „Untere Schranke“ als jene Größe, welche das theoretisch erreichbare Minimum der zu findenden Lösung darstellt. Die untere Schranke kann vor der Verwendung des Packalgorithmus berechnet werden, und somit der Datensatz klassifiziert werden. Auf Grund der 2-Dimensionalität werden zwei untere Schranken definiert, jene bei der das zu packende Volumen bestimmend ist und jene bei der das zu packende Gesamtgewicht die untere Schranke definiert.

Bezeichnet werden im Folgenden  $W$  als das gesamt zu packende Gewicht aller UL,  $M$  die Anzahl aller zu packenden UL,  $c^w$  als die Gewichtskapazität eines LKW und  $c^v$  als die maximale Anzahl an UL, die ein LKW laden kann.

$$\text{Untere Schranke „Volumen“} \quad LB^V = \left\lceil \frac{M}{c^V} \right\rceil \quad \text{Gl. 4-1}$$

$$\text{Untere Schranke „Gewicht“} \quad LB^w = \left\lceil \frac{W}{c^w} \right\rceil \quad \text{Gl. 4-2}$$

$$\text{Untere Schranke} \quad LB = \max \{LB^w, LB^V\} \quad \text{Gl. 4-3}$$

In der Literatur werden noch weitreichendere untere Schranken definiert, welche im Zuge der definierten Aufgabenstellung jedoch keine Bedeutung haben, da keine „schweren“ UL zu packen sind. Als „schwer“ werden jene zu packende Objekte bezeichnet, von denen mindestens eine Dimension größer ist als das halbe Kapazitätslimit der jeweiligen Dimension (vgl. [CT00]).

Da auf Grund der Rahmenbedingungen keine UL schwerer sein kann als 2.000kg (siehe Begriffsdefinition der UL in Kapitel 2.1) und nur 1 Stellplatz am LKW benötigt, ist auf Grund der Gewichts- und Volumenkapazitäten der betrachtenden LKW kein „schweres“ Objekt zu packen.

## 4.1.4 Methoden

### Literatur

In der Literatur sind viele Näherungsalgorithmen und Heuristiken dokumentiert. Eine ausführliche Darstellung und Analyse dieser Algorithmen würde an dieser Stelle zu weit gehen, daher werden Arbeiten herangezogen, welche die prominentesten Berechnungsmethodiken vergleichen. Zwei Publikationen von Remic, Zerovnic J. und Zerovnic G. [RZZ12] und Dell’Amico, Iori und Martello [AIM04] werden hierbei speziell betrachtet. Die Publikation von Remic, Zerovnic J. und Zerovnic G. ist von besonderem Interesse da die beschriebene Problemstellung ohne „schwere“ Objekte als Testinstanz verwendet wird und somit die Aufgabenstellung dieser Arbeit am besten abbildet.

Von Remic, Zerovnic J. und Zerovnic G. werden in [RZZ12] die Algorithmen „First Fit Decreasing“ (FFD), „Refined First Fit“ (RFF), getestet, eine Algorithmus von Zhang, Chai und Wang [ZCW00] sowie jener von Kellerer und Pferschi [KP99] und 3 von Zerovnic J. und Zerovnic G. in [ZZ11] vorgestellte Algorithmen (Alg. 1, Alg. 2, Alg. 3). Die Ergebnisse

mit dem betrachteten Datensätzen (0% „schwere“ Objekte) werden wie in Tabelle 4-1 zusammengefasst dargestellt.

**Tabelle 4-1: Zusammengefasste Ergebnisse aus [RZZ12] für 0% "schwere" Objekte**

Algorithmus	Ergebnis über „unterer Schranke“
<b>FFD</b>	12,75 %
<b>RFF</b>	33,17 %
<b>ZCW</b>	36,07 %
<b>KP</b>	4,15 %
<b>Alg. 1</b>	2,63 %
<b>Alg. 2</b>	4,94 %
<b>Alg. 3</b>	2,59 %

Eine Darstellung der Berechnungsdauer wird nicht explizit ausgewiesen (vgl. [RZZ12]).

Des Weiteren vergleichen Dell’Amico, Iori und Martello 6 weitere Algorithmen für das Vector Bin Packing Problem. Der in dieser Publikation vorgestellte Algorithmus namens „B&B“ erreicht bei dort verwendeten Testinstanzen exzellente Ergebnisse in äußerst kurzer Berechnungszeit. Da aber die Charakteristika der getesteten Datensätze sehr weit von denen abweichen, die in dieser Aufgabenstellung zu bewältigen sind, ist es nicht möglich die Eignung des Algorithmus für die beschriebene Aufgabe zu bestimmen (vgl. [AIM04]).

## 4.2 Umsetzung

Basierend auf der Recherche in Kapitel 4.1.4 werden hier zwei Algorithmen vorgestellt, die der Problemstellung des „Change Making Problem“ ähnlich sind. Das „Change Making Problem“ beschreibt jenes Problem, welches eine Servierkraft im Restaurant beim Herausgeben des Restbetrages hat: Die Anzahl der Münzen soll ein Minimum sein und der Restbetrag aber exakt erreicht werden (vgl. [MT90]). Basierend auf der Annahme, dass für das 2D Vector Bin Packing Problem eine Lösung innerhalb von  $LB + 1$  gefunden werden kann wenn ein LKW beide Kapazitäten (Volumen und Gewicht) zur Gänze ausnutzt. Das bedeutet, dass UL Kombinationen gefunden werden müssen, sodass so viele UL wie möglich auf einen LKW platziert werden und dass die Summe der UL Einzelgewichte möglichst nahe an der Gewichtskapazitätsgrenze des LKW liegt. Hier zeigt sich auch die Analogie zum „Change Making Problem“.

Je nachdem welche untere Schranke, basierend auf Gewicht oder Volumen, die definierende ist, werden die beiden Algorithmen angewandt. Diese Unterscheidung ist sinnvoll, da die untere Schranke vor dem Beginn der Berechnung ermittelt werden kann und ein aussagekräftiges Kriterium zur Klassifizierung der Problemstellung darstellt. Da beide Algorithmen sehr ähnlich sind werden im folgenden Abschnitt die Gemeinsamkeiten dargestellt und bei entsprechenden Abweichungen Alg. 1 und Alg. 2 markiert.

Die Algorithmen wurden in einen VB.net Code implementiert, welcher die Datensätze an einen geeigneten Microsoft EXCEL 2013® Solver übergibt und an entsprechender Stelle inkrementiert. Versuche haben gezeigt, dass die besten Ergebnisse (wie später unten dargestellt) mit dem „Branch and Bound“ Solver sowie „non-linear“ Solver erzielt werden, abhängig vom verwendeten Algorithmus. Tabelle 4-2 zeigt in einer Übersicht welcher Algorithmus bei welcher Aufgabenstellung zur Anwendung kommt und in welchen Solver dieser implementiert ist.

Tabelle 4-2: Übersicht Algorithmen, Anwendung und Implementierung

Algorithmus	Anwendung	Verwendeter Solver
1	Untere Schranke definiert durch Gewicht	non-linear
2	Untere Schranke definiert durch Volumen	Branch and Bound

### 4.2.1 Bin Packing - Algorithmus und Implementierung

#### Notation

$i$  = Index für LKW

$j$  = Index für ULs

$m$  = Index für Rechendurchgänge während LKW  $i$  gepackt wird

$y_{ij}$  = Entscheidungsvariable für UL  $j$  gepackt auf LKW  $i$

$x_i$  = Entscheidungsvariable für LKW  $i$

$M$  = Gesamtzahl an UL

$M_i$  = Anzahl der gepackten ULs auf LKW  $i$

$W$  = Gesamtgewicht der UL

$W_i$  = Gewicht der gepackten ULs auf LKW  $i$

$N$  = Anzahl benötigter Trailer

$w_j$  = Gewicht von UL  $j$

$c_i^v$  = Volumenkapazität von LKW  $i$

$c_i^w$  = Gewichtskapazität von LKW  $i$

$c_i^w(m)$  = Modifizierte Gewichtskapazität von LKW  $i$  im Rechendurchlauf  $m$

Wenn alle Trailer dieselbe Volumen- und Gewichtskapazität besitzen gilt:

$$c_i^v = c^v \quad \forall i \quad \text{Gl. 4-4}$$

$$c_i^w = c^w \quad \forall i \quad \text{Gl. 4-5}$$

#### Sätze

$$j \in L \quad \text{wobei } |L| = M \quad \text{Gl. 4-6}$$

$$i \in T \quad \text{wobei } \sum_{i=1}^M x_i = N \quad \text{Gl. 4-7}$$

$$m \in \mathbb{N}^+ \quad \text{Gl. 4-8}$$

**Limits**

$$M_1 \equiv M \quad \text{Gl. 4-9}$$

$$M_i = M_{i-1} - \sum_{j=1}^M y_{ij} \quad \forall i > 1 \quad \text{Gl. 4-10}$$

$$W_1 \equiv W \quad \text{Gl. 4-11}$$

$$W_i = W_{i-1} - \sum_{j=1}^M y_{ij} * w_j \quad \forall i > 1 \quad \text{Gl. 4-12}$$

**Algorithmus 1**

$$c_i^w(1) \equiv c^w \quad \forall i \quad \text{Gl. 4-13}$$

$$c_i^w(m) = c_i^w(m-1) - 10 \quad \forall i; \forall m > 1 \quad \text{Gl. 4-14}$$

$$x_i, y_{ij} \in \{0,1\} \quad \forall i; \forall i,j \quad \text{Gl. 4-15}$$

**Algorithmus 2**

$$c_i^v(1) \equiv c^v \quad \forall i \quad \text{Gl. 4-16}$$

$$c_i^v(m) = c_i^v(m-1) - 1 \quad \forall i; \forall m > 1 \quad \text{Gl. 4-17}$$

$$x_i, y_{ij} \in \{0,1\} \quad \forall i; \forall i,j \quad \text{Gl. 4-18}$$

**Parameter**

Jedes zu lösende Problem kann in folgenden Parametern variieren:

$L(M, W, \{w_j\})$  &  $T(c^v, c^w)$

Zum Beispiel:

5 ULs mit 900 kg/UL

20 ULs mit 800 kg/UL

27 UL mit 500 kg/UL

19 ULs mit 300 kg/UL sollen auf Trailer selben Typs gepackt werden.

$M = 71$

$W = 39.700 \text{ kg}$

$w_j \in \{900, 800, 500, 300\}$

$c^v = 33 \text{ ULs/LKW}$

$c^w = 24.000 \text{ kg/LKW}$

$LB = 3$

**Entscheidungsvariablen**

Der Algorithmus legt fest, welche UL auf welchen LKW gepackt werden

$$y_{ij} = 0 \text{ oder } 1 \quad y_{ij} = \begin{cases} 1 & \text{wenn UL } j \text{ auf Trailer } i \text{ gepackt wird} \\ 0 & \text{sonst} \end{cases} \quad \text{Gl. 4-19}$$

$$i, j = 1, \dots, M \quad \text{Gl. 4-20}$$

$$x_i = 0 \text{ oder } 1 \quad x_i = \begin{cases} 1 & \text{wenn Trailer } i \text{ verwendet wird} \\ 0 & \text{sonst} \end{cases} \quad \text{Gl. 4-21}$$

$$i = 1, \dots, M \quad \text{Gl. 4-22}$$

**Ziel**

Ziel des gesamten Packprozesses ist die Minimierung der verwendeten LKW N. Aber die Zielzellen der Solverimplementierung variieren, je nach Algorithmus.

**Algorithmus 1**

$$\text{maxmiere } \sum_{j=1}^M y_{ij} \quad \forall i \quad \text{Gl. 4-23}$$

**Algorithmus 2**

$$\text{maxmiere } \sum_{j=1}^M y_{ij} * w_j \quad \forall i \quad \text{Gl. 4-24}$$

**Einschränkungen**

$$N = \sum_{i=1}^M x_i \quad \text{Gl. 4-25}$$

$$\sum_{i=1}^M y_{ij} = 1 \quad \forall j \quad \text{Gl. 4-26}$$

Gewichtsbeschränkung:

**Algorithmus 1**

$$\sum_{j=1}^M y_{ij} * w_j \equiv c_i^w(1) \equiv c^w \quad \forall i, m = 1 \quad \text{Gl. 4-27}$$

$$\sum_{j=1}^M y_{ij} * w_j \equiv c_i^w(m-1) - 10 \quad \forall i, \forall m > 1 \quad \text{Gl. 4-28}$$

**Algorithmus 2**

$$\sum_{j=1}^M y_{ij} * w_j \leq c_i^w \quad \forall i \quad \text{Gl. 4-29}$$

Volumenbeschränkung:

**Algorithmus 1**

$$\sum_{j=1}^M y_{ij} \leq c^v \quad \forall i \quad \text{Gl. 4-30}$$

**Algorithmus 2**

$$\sum_{j=1}^M y_{ij} \equiv c_i^v(1) \equiv c^v \quad \forall i, m = 1 \quad \text{Gl. 4-31}$$

$$\sum_{j=1}^M y_{ij} \equiv c_i^v(m-1) - 1 \quad \forall i, \forall m > 1 \quad \text{Gl. 4-32}$$

$$x_i, y_{ij} \in \{0,1\} \quad \forall j; \forall i, j \quad \text{Gl. 4-33}$$

**Implementierung:**

Der bei der jeweiligen Problemstellung verwendete Algorithmus wird in den Unit Load Packer wie in Abbildung 4-1 gezeigt, implementiert. Die Programmstruktur ist deswegen relevant, da diese die einzelnen Schleifen in dargestellter Weise steuert und inkrementiert. Der Unit Load Packer liest die Eingangsdatei, wandelt sie in die benötigten Eingangsdaten um und setzt beim ersten Durchlauf den Index für den zu packenden LKW auf eins. Danach werden die Parameter an den Solver übergeben und gestartet. Der Algorithmus findet in der oben beschriebenen Berechnungsmethodik eine Lösung für den zu packenden LKW. Die Ergebnisse werden wieder vom Unit Load Packer übernommen, eine neue Packliste mit den veränderten Eingangsdaten generiert und dem Bin Packing Algorithmus übergeben. Dieses Procedere wird so lange wiederholt, bis alle UL einem LKW zugeordnet sind und eine Ergebnisdatei generiert und ausgegeben werden kann.

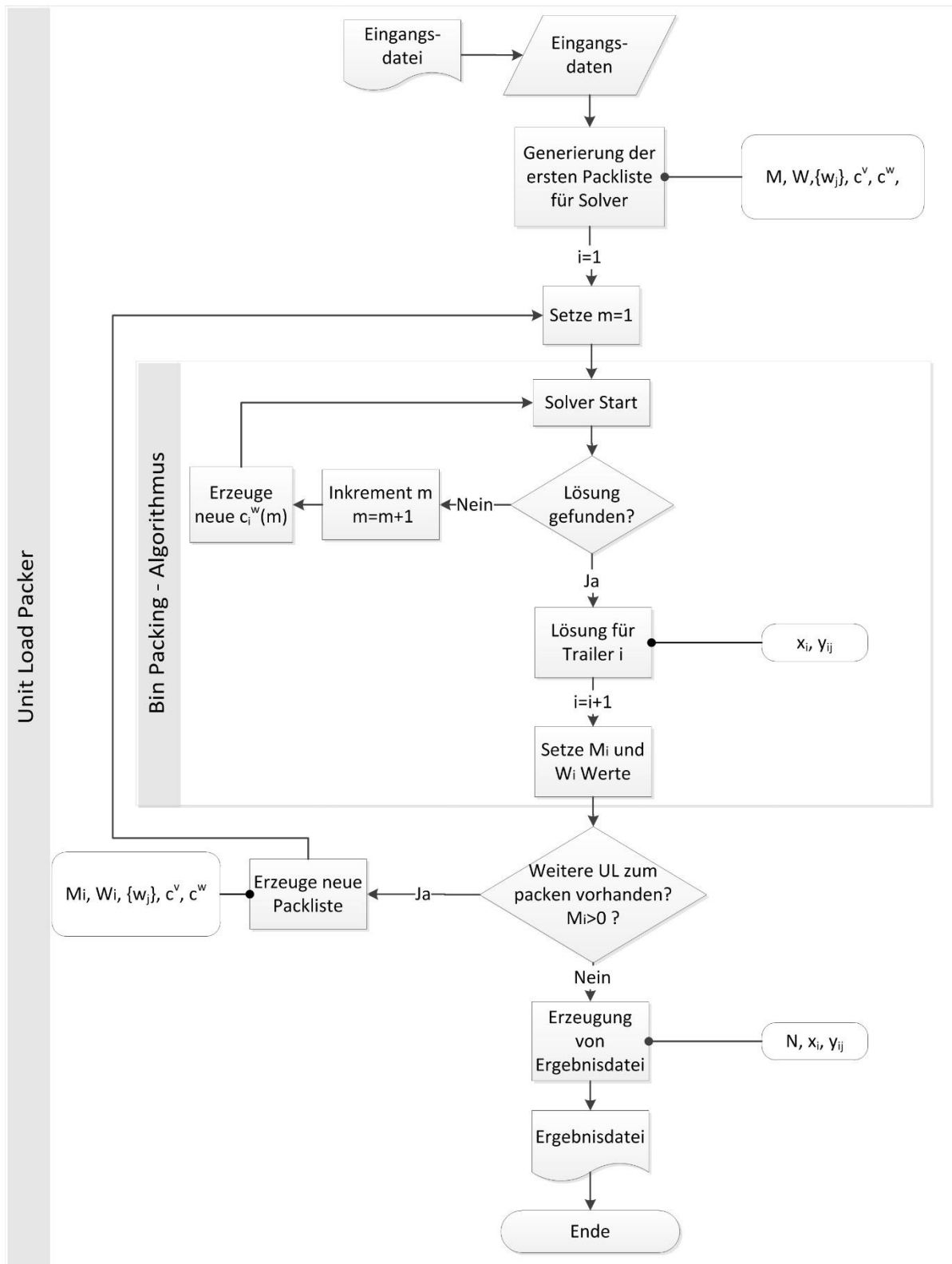


Abbildung 4-1: Implementierung der Algorithmen in Unit Load Packer



## 4.2.2 Begründung

Die Ergebnisse der in Kapitel 3.1.3 dargestellten und getesteten Algorithmen weisen eine zu hohe Abweichung von der geforderten Zielformulierung auf oder die Charakteristika der Testinstanzen variieren zu stark von jenen dieser Arbeit, sodass eine Bewertung für die Eignung für diese Problemstellung nicht durchgeführt werden kann. Des Weiteren ist auf Grund des großen Rechenzeitraumes und der guten Vorhersehbarkeit des Bedarfes an FMCG die Rechendauer für eine Probleminstanz von untergeordneter Rolle. Deswegen wurden mit der zu Grunde gelegten Annahme, dass eine Lösung innerhalb der unteren Schranke gefunden werden kann wenn ein LKW beide Kapazitäten ganz auslastet, die beiden Algorithmen mit variierenden Zielvorgaben entwickelt.

Die Unterscheidung welcher Algorithmus zu wählen ist, ist einfach, da die untere Schranke im Vorhinein berechnet und somit die Problemstellung klassifiziert werden kann. Auf dieser Entscheidung variieren die beiden Algorithmen. Wenn die untere Schranke bestimmt ist durch die Volumenbeschränkung (Gl. 4-1) packt Algorithmus 1 so viele UL wie möglich auf den LKW (Gl. 4-23) unter der Bedingungen dass das Gesamtgewicht der UL exakt die Gewichtskapazität erreicht (Gl. 4-27). Ist eine Konstellation nicht möglich, die genau diese Bedingung erfüllt, so wird diese Gewichtskapazität modifiziert (Gl. 4-28). Genau umgekehrt verhält sich Algorithmus 2, der zur Anwendung kommt, wenn die untere Schranke durch die Gewichtsbeschränkung definiert ist (Gl. 4-2). Die Zielvorgabe lautet hier so viel Gewicht wie möglich auf den LKW zu packen (Gl. 4-24) unter der Bedingung dass die Anzahl der gepackten UL exakt der Volumenkapazität entspricht (Gl. 4-31). Kann diese Vorgabe nicht erreicht werden, wird die Volumenkapazität modifiziert (Gl. 4-32).

Durch diese Vorgaben, dass eine zweite Grenze exakt erreicht werden muss, wird eine zweite Zielformulierung in den Lösungsprozess des Solvers implementiert wodurch die zu Grunde liegende Forderung, beide Kapazitätsgrenzen auszulasten, erfüllt werden kann.

Die modifizierten Volumen- und Gewichtskapazitäten verringern ihren Wert pro Inkrement (siehe Abbildung 4-1) um 1 bzw. 10. Dies ist möglich da nur ganze UL verladen werden und die UL in 10kg Schritte klassifiziert werden.

## 4.3 Ergebnisse der Vorgehensweise

In diesem nun folgenden Abschnitt werden die Prüfdaten, mit welchen der Algorithmus getestet wurde, sowie die erreichten Ergebnisse vorgestellt.

### 4.3.1 Prüfdaten

Die Prüfdaten wurden von Russel Meller basierend auf dem MODULUSHCA Szenario erarbeitet und deren Zustandekommen in [MLE12] dokumentiert. Die erarbeiteten Datensätze sind jene, die hier Eingang finden und angewandt werden um den Algorithmus zu testen. In Tabelle 4-3 bis Tabelle 4-8 werden jene wesentliche Eigenschaften der Datensätze gezeigt, welche zur späteren Verwendung und Beurteilung der Ergebnisse relevant sind.

Getestet wurde der Unit Load Packer auf einem ACER® Büro PC mit Intel® Core™ i7 – 3930K CPU @ 3.2GHz, 64GB Arbeitsspeicher und Windows 7 64bit SP1.

Mit „Diversifikation“ wird die Anzahl der unterschiedlichen UL – Klassen bezeichnet. Die leichteste UL Klasse ist 100kg und die schwerste 2.000kg, gemäß den Ergebnissen aus Kapiteln 3.3 und 2.1. Da die UL Klassen in den Datensätzen oft nur im Bereich von Gramm

variieren und der Tatsache, dass EXCEL® 2013 Solver eine maximale Zellenanzahl von 200 erlaubt (vgl. [MIC14]), werden die Eingangsdaten in 10, 20, 50 und 100kg Schritten klassifiziert und zusammengefasst.

Das bedeutet, dass zum Beispiel die Anzahl der UL zwischen 110 und 120kg ermittelt wird und entsprechend des Mittelwertes auf 110 oder 120kg gerundet wird. Der dadurch entstehende Fehler muss in der Ergebnisbewertung berücksichtigt und dargestellt werden. Die maximale Gewichtskapazität bei LKW wird mit 4% toleriert, wie später in Kapitel 5.1.2 dargestellt. Dies führt zu der Tatsache, dass eine Fehlermarke von unter 4% durch das Klassifizieren für die zu lösende Gesamtproblemstellung zulässig ist.

Tabelle 4-3 bis Tabelle 4-8 zeigen die Eigenschaften der als Testinstanzen verwendeten Datensätze.

Tabelle 4-3: Übersicht Eingangsdaten PG\_Output\_5mbox\_new

Datensatz	PG_Output_5mbox_new				
	wahr	10	20	50	100
Klasse [kg]					
Anzahl UL [ ]	24.579	24.579	24.579	24.579	24.579
UL Gesamtgewicht [kg]	14.760.208,60	14.761.920	14.776.560	14.750.400	14.725.880
Abweichung durch Klassifizierung	-	-0,012%	-0,111%	-0,066%	-0,233%
Mittelwert [kg]	600,52	600,59	601,19	600,12	599,12
LBoW	616	616	616	616	616
LBoV	745	745	745	745	745
LB	745	745	745	745	745
Diversifikation	1250	118	68	33	18

Tabelle 4-4: Übersicht Eingangsdaten PG\_Output\_6mbox\_new

Datensatz	PG_Output_6mbox_new				
	wahr	10	20	50	100
Klasse [kg]					
Anzahl UL [ ]	20.360	20.360	20.360	20.360	20.360
UL Gesamtgewicht [kg]	13.480.741,20	13.502.390	13.506.790	13.466.150	13.440.650
Abweichung durch Klassifizierung	-	-0,163%	-0,195%	-0,106%	-0,295%
Mittelwert [kg]	662,11	663,18	663,40	661,40	660,15
LBoW	562	562	562	562	562
LBoV	617	617	617	617	617
LB	617	617	617	617	617
Diversifikation	1234	112	62	28	16

Tabelle 4-5: Übersicht Eingangsdaten PG\_Output\_9.8mbox

Datensatz	PG_Output_9.8mbox				
	wahr	10	20	50	100
Klasse [kg]					
Anzahl UL [ ]	21.503	21.503	21.503	21.503	21.503
UL Gesamtgewicht	14.065.844,50	14.068.950	14.076.250	14.055.890	14.092.240
Abweichung durch Klassifizierung	-	-0,022%	-0,074%	0,071%	-0,188%
Mittelwert [kg]	654,13	654,28	654,62	653,67	655,36
LBoW	587	587	587	587	587
LBoV	652	652	652	652	652
LB	652	652	652	652	652
Diversifikation	1250	124	72	33	18

Tabelle 4-6: Übersicht Eingangsdaten PG\_Output\_12mbox

Datensatz	PG_Output_12mbox				
	wahr	10	20	50	100
Klasse [kg]					
Anzahl UL [ ]	16.855	16.855	16.855	16.855	16.855
UL Gesamtgewicht [kg]	11.471.547,10	11.461.620	11.476.980	11.453.120	11.489.690
Abweichung durch Klassifizierung	-	0,087%5	-0,047%	0,161%	-0,158%
Mittelwert [kg]	680,60	680,01	680,92	679,51	681,68
LBoW	478	478	478	478	478
LBoV	511	511	511	511	511
LB	511	511	511	511	511
Diversifikation	1197	137	78	35	18

Tabelle 4-7: Übersicht Eingangsdaten PG\_Output\_Tophalf\_(78)\_mbox\_10

Datensatz	PG_Output_Tophalf_(78)_mbox_10				
	wahr	10	20	50	100
Klasse [kg]					
Anzahl UL [ ]	16.399	16.399	16.399	16.399	16.399
UL Gesamtgewicht [kg]	13.198.489,80	13.202.960	13.203.130	13.217.690	13.195.400
Abweichung durch Klassifizierung	-	-0,034%	-0,035%	-0,145%	0,023%
Mittelwert	804,84	805,11	805,12	806,01	804,65
LBoW	550	551	551	551	550
LBoV	497	497	497	497	497
LB	550	511	511	511	511
Diversifikation	1248	136	75	33	17

Tabelle 4-8: Übersicht Eingangsdaten PG\_Output\_440.155mbox

Datensatz	PG_Output_440.155mbox				
	wahr	10	20	50	100
Klasse [kg]					
Anzahl UL [ ]	16.389	16.389	16.389	16.389	16.389
UL Gesamtgewicht [kg]	13.155.531,40	13.158.710	13.160.510	13.146.970	13.145.160
Abweichung durch Klassifizierung	-	-0,024%	-0,038%	0,065%	0,079%
Mittelwert [kg]	802,70	802,90	803,01	802,18	802,07
LBoW	549	549	549	548	548
LBoV	497	497	497	497	497
LB	549	549	549	548	548
Diversifikation	1250	140	77	33	18

### 4.3.2 Ergebnisse

Die Ergebnisse des in Kapitel 4.2.1 verwendeten Beispiel-Datensatzes werden in Tabelle 4-9 detailliert dargestellt.

Tabelle 4-9: Detaillierergebnisse des Beispieldatensatzes

LKW Nr.	Gesamtzahl UL [#]	Gesamtgewicht [kg]	Anzahl der UL [#]	UL Gewicht [kg/UL]
<b>1</b>	33	24.000	1	300
			8	500
			19	800
			5	900
<b>2</b>	33	14.200	13	300
			19	500
			1	800
<b>3</b>	5	1.500	5	300

Da eine detaillierte Darstellung aller Ergebnisse der Datensätze an dieser Stelle zu weit gehen würde, ist in Tabelle 4-10 bis Tabelle 4-14 eine Übersicht der Ergebnisse, berechnet mit den Datensätzen aus Kapitel 4.3.1, dargestellt.

Ergebnisse die mit [-] gekennzeichnet sind, wurden nach einer Berechnungsdauer von 24h abgebrochen.

Tabelle 4-10: Übersicht der Ergebnisse für PG\_Output\_5mbox\_new

Allgemein	Datensatz	PG_Output_5mbox_new			
	Klasse [kg]	10	20	50	100
	Wahrer LB	745	745	745	745
	LB nach Klassifizierung	745	745	745	745
	Ergebnis	745	745	745	745
Algo-rithmus 1	Abweichung wahren LB	0,00%	-	0,00%	0,00%
	Abweichung LB	0,00%	-	0,00%	0,00%
	Berechnungszeit [hh:mm:ss]	03:32:00	-	00:44:45	00:27:29
Algo-rithmus 2	Ergebnis	745	745	745	745
	Abweichung wahren LB	0,00%	0,00%	0,00%	0,00%
	Abweichung LB	0,00%	0,00%	0,00%	0,00%
	Berechnungszeit [hh:mm:ss]	00:31:36	00:26:06	00:20:53	00:18:20

Tabelle 4-11: Übersicht der Ergebnisse für PG\_Output\_6mbox\_new

Allgemein	Datensatz	PG_Output_6mbox_new				
	Klasse [kg]	10	20	50	100	
Allgemein	Wahrer LB	617	617	617	617	
	LB nach Klassifizierung	617	617	617	617	
	Algorithmus 1	Ergebnis	617	617	619	630
	Abweichung wahren LB	0,00%	0,00%	0,324%	2,107%	
Algorithmus 1	Abweichung LB	0,00%	0,00%	0,324%	2,107%	
	Berechnungszeit	03:15:00	02:56:00	00:33:00	00:22:00	
	Algorithmus 2	Ergebnis	617	617	617	617
	Abweichung wahren LB	0,00%	0,00%	0,00%	0,00%	
Algorithmus 2	Abweichung LB	0,00%	0,00%	0,00%	0,00%	
	Berechnungszeit [hh:mm:ss]	00:29:41	00:22:43	00:17:03	00:15:33	

Tabelle 4-12: Übersicht der Ergebnisse für PG\_Output\_12mbox –

Allgemein	Datensatz	PG_Output_12mbox				
	Klasse [kg]	10	20	50	100	
Allgemein	Wahrer LB	511	511	511	511	
	LB nach Klassifizierung	511	511	511	511	
	Algorithmus 1	Ergebnis	511	–	511	528
	Abweichung wahren LB	0,00%	–	0,00%	3,327%	
Algorithmus 1	Abweichung LB	0,00%	–	0,00%	3,327%	
	Berechnungszeit [hh:mm:ss]	02:53:34	–	00:49:49	01:06:00	
	Algorithmus 2	Ergebnis	511	511	511	511
	Abweichung wahren LB	0,00%	0,00%	0,00%	0,00%	
Algorithmus 2	Abweichung LB	0,00%	0,00%	0,00%	0,00%	
	Berechnungszeit [hh:mm:ss]	00:45:53	00:37:43	00:27:42	00:23:42	

Tabelle 4-13: Übersicht der Ergebnisse für PG\_Output\_Tophalf\_(78)\_mbox\_10

Allgemein	Datensatz	PG_Output_Tophalf_(78)_mbox10			
	Klasse [kg]	10	20	50	100
	Wahrer LB	550	550	550	550
	LB nach Klassifizierung	551	551	551	550
Algorithmus 1	Ergebnis	–	–	551	550
	Abweichung wahren LB	–	–	0,182%	0,00%
	Abweichung LB	–	–	0,00%	0,00%
	Berechnungszeit [hh:mm:ss]	–	–	04:29:09	00:41:00
Algorithmus 2	Ergebnis	552	552	552	554
	Abweichung wahren LB	0,364%	0,364%	0,364%	0,727%
	Abweichung LB	0,181%	0,181%	0,181%	0,727%
	Berechnungszeit [hh:mm:ss]	00:38:40	00:30:33	00:21:38	00:18:24

Tabelle 4-14: Übersicht der Ergebnisse für PG\_Output\_440.155mbox

Allgemein	Datensatz	PG_Output_440.155mbox			
	Klasse [kg]	10	20	50	100
	Wahrer LB	549	549	549	549
	LB nach Klassifizierung	549	549	548	548
Algorithmus 1	Ergebnis	–	–	548	548
	Abweichung wahren LB	–	–	-0,182%	-0,182%
	Abweichung LB	–	–	0,00%	0,00%
	Berechnungszeit [hh:mm:ss]	–	–	01:11:00	00:29:01
Algorithmus 2	Ergebnis	551	551	550	551
	Abweichung wahren LB	0,364%	0,364%	0,182%	0,364%
	Abweichung LB	0,364%	0,364%	0,365%	0,547%
	Berechnungszeit [hh:mm:ss]	00:28:29	00:21:42	00:15:00	00:14:24

#### 4.4 Bewertung/Beurteilung der Vorgehensweise

Auf Basis der speziellen Anforderungen, die eine nachrangige Notwendigkeit der Zeiteffizienz aufweisen, wurde ein Packalgorithmus entworfen, der, abhängig von den Charakteristika der Aufgabenstellung, verschiedene Ziele verfolgen kann um die Anzahl der zu verwendeten LKW zu minimieren. Die in Kapitel 5.3.1 ausgewiesenen Ergebnisse zeigen, dass für jede Aufgabenstellung in zumindest einer Klasse (Erläuterung siehe Kapitel 4.3.1) die untere Schranke erreicht wird. Der entwickelte Algorithmus erfüllt alle Ziele, die für diese Anwendung definiert wurden.

Wie die Ergebnisse zeigen, ist die Analyse des Datensatzes auf dessen Lösbarkeit schwierig und anscheinend vor dem Beginn des Berechnungsprozesses nicht absehbar. Verschiedenste Versuche mit unterschiedlichen Solver-Typen führten zur Entwicklung der beiden Algorithmen. Die Ergebnisse zeigen, dass sich weder die Anzahl der UL noch deren Diversifikation auf die Lösbarkeit auswirken. Die Unterscheidung welche untere Schranke bestimmend ist lässt zumindest auf den besser geeigneten Algorithmus schließen, sagt aber nichts über die einfache Lösbarkeit aus.



Einige Schwächen des Algorithmus müssen aber an dieser Stelle aufgezeigt werden:

- Die Daten müssen zumindest in 10kg Schritte klassifiziert werden, wie in Kapitel 4.3.1 erläutert.
- Durch die Iterationsschleifen beim Modifizieren der Volumen- oder Gewichtsbeschränkungen, ist der Algorithmus bei Aufgaben mit einer sehr geringen unteren Schranke viel langsamer als die in [AIM04] vorgestellten Lösungsmethoden.
- Nicht jede Klasse der Datensätze konnte innerhalb von 24h Stunden gelöst werden.
- Für die Verwendung muss eine Microsoft EXCEL® 2013 Softwarelizenz zur Verfügung stehen

## 5 Balanced Trailer Load

Im folgenden Kapitel wird die Entwicklung der „Balanced Trailer Load“ Berechnungsmethodik dargestellt. Beginnend bei der Zielformulierung, die aus den Anforderungen des Gesamtkontext, dargestellt in 2.3, resultieren, über Rahmenbedingungen, Umsetzung sowie Ergebnisse und Interpretation.

### 5.1 Entwicklung von Maßnahmen

#### 5.1.1 Zielformulierung

Der Balanced Trailer Load Algorithmus knüpft an die Ergebnisse des Unit Load Packer an. Wie in 4.1.1 beschrieben ist die Aufgabe des Unit Load Packers das Aufteilen der UL an LKW sodass die Anzahl der LKW ein Minimum wird. Da diese Problemstellung und deren Lösung von mathematischer Natur ohne Berücksichtigung von technischen Einschränkungen ist, müssen die Ergebnisse auf deren Realisierbarkeit überprüft werden. Diese Überprüfung der Realisierbarkeit fokussiert hierbei auf das Anordnen der zugeteilten UL am LKW. Lastkraftwagen sind gesetzlichen Bestimmungen unterworfen, welche zum Beispiel maximale Achs- und Sattellast sowie das höchst zulässige Gesamtgewicht beschränken. Um eine gegebene Menge an UL vom Distributionszentrum verschicken zu können, müssen diese auf LKW angeordnet werden unter folgenden Gesichtspunkten:

- Einhalten gesetzlicher Bestimmungen
- Rücksichtnahme auf technische Aspekte zum sicheren Transport

Ziel dieser Anwendung ist es, für jeden LKW einen Packplan generieren zu können, der festlegt welche UL an welchen Platz auf der Ladefläche positioniert werden muss um die oben genannten technischen und rechtlichen Bedingungen erfüllen zu können. Aus dieser Zielformulierung geht auch hervor, dass keine exakte oder optimale Lösung benötigt wird. Eine Lösung, welche sämtliche Rahmenbedingungen erfüllt, ist ausreichend.

Die Balanced Trailer Load Berechnungsmethodik dient zum einen der Überprüfung der Realisierbarkeit der Lösungen vom Unit Load Packer und zum anderen auch als Werkzeug für den laufenden Distributionsbetrieb.

## 5.1.2 Rahmenbedingungen

### Fahrzeugkombination

Betrachtet wird ein in der Realität häufig verbreitete Kombination aus Kraftmaschine und Sattelanhänger mit den in Tabelle 5-1 gezeigten Spezifikationen:

Tabelle 5-1: Spezifikationen der betrachteten Fahrzeugkombination

Eigenschaft	Wert
Eigengewicht (Kraftmaschine und Sattelanhänger)	26.000 kg
Ladefläche (Länge x Breite)	13,6 x 2,5 m
Bereifung (Sattelanhänger)	Dreifachachse mit weniger als 1,3m Radstand
Anzahl der Achsen der Fahrzeugkombination	5

Sattelanhänger, die auf für den Transport von Paletten konzipiert wurden, besitzen ein Absperrsystem um das Verrutschen von Paletten zu verhindern. Dadurch können einzelne Stellplätze oder auch Zeilen frei bleiben ohne die Ladungssicherung zu gefährden. Die Abmessungen von UL und Sattelanhänger erlauben zwei Anordnungsmöglichkeiten von UL am LKW: In 3er Reihe, wie in Abbildung 5-1 dargestellt, und in 2er Reihe, wie in Abbildung 5-2 gezeigt. Um weiter eine Entscheidung treffen zu können, welche Anordnung gewählt wird, werden die Unterschiede hinsichtlich Kapazitätsausnutzung und Praxisrelevanz aufgezeigt und daraus Vor- und Nachteile abgeleitet.



Abbildung 5-1: Anordnung in 3er Reihe



Abbildung 5-2: Anordnung in 2er Reihe

**Kapazität:** Die Anordnung in 2er Reihe ermöglicht das Laden von 34 UL, die Anordnung in 3er Reihe hingegen 33 UL. Die Kapazitätserhöhung bei Verwendung der 2er Reihe ist als Vorteil zu nennen.

**Anordnungsspielraum:** Bei Verwendung der 3er Reihe, bleiben 400 mm „Spielraum“ um die UL am LKW zu platzieren, in 2er Reihe hingegen 0 mm. Dies bringt

den Nachteil mit sich, dass praxisrelevante Prozesse wie das Schließen der Türen und das Anbringen von Ladungssicherungen erschwert werden oder nicht möglich sind.

**Gemischte Anordnung:** Wird in 3er Reihe gepackt, hat der/die für die Ladungs- und Lastverteilung Zuständige, die Möglichkeit eine gemischte Anordnung zu verwenden, wenn die Volumenkapazität des LKW nicht ausgelastet wird, wie in Abbildung 5-3 gezeigt. Diese Anordnung birgt den Vorteil, dass Lücken einfach durch Querstellen der angrenzenden UL geschlossen werden können ohne zusätzliche Ladesicherungsvorkehrungen treffen zu müssen. Das Verwenden der gemischten Anordnung wird umso relevanter, je mehr die Kapazitätsauslastung vom maximal zulässigen Ladungsgewicht bestimmt wird.

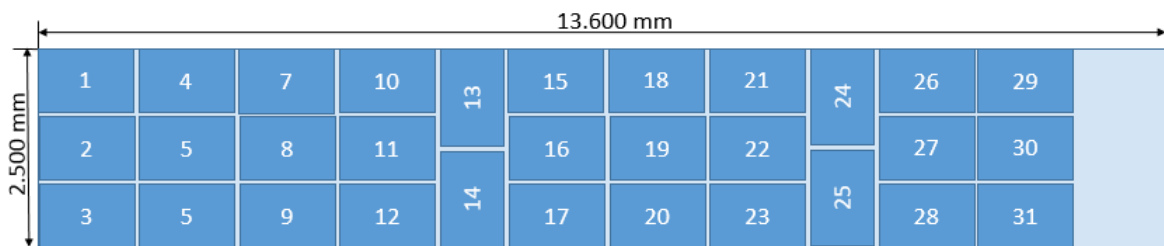


Abbildung 5-3: Gemischte Anordnung von UL am LKW zur Vermeidung von Lücken

Die oben dargestellten Eigenschaften der jeweiligen Anordnungsvarianten sind in Tabelle 5-2 gegenübergestellt.

Tabelle 5-2: Anordnungsvarianten von UL am LKW - Übersicht

Eigenschaft	2er Reihe	3er Reihe
Kapazität [UL]	34	33
Anordnungsspielraum [mm]	0	400
Möglichkeit zur gemischten Anordnung	Nein	Ja

Auf Grund der Tatsache, dass nur die Kapazitätssteigerung der Volumenkapazität für die Anordnung in 2er Reihe spricht und damit aber wesentliche, praxisrelevante Nachteile einhergehen, wird die Anordnung in 3er Reihe als Standard-Packmuster implementiert.

**Restriktionen:**

Als Definitionen für die zulässigen rechtlichen und technischen Rahmenbedingungen werden in dieser Arbeit 3 Instanzen identifiziert:

- Kraftfahrzeuggesetz Österreich (KFG)
- VDI 2700
- Praxisanwendung von Transportunternehmen

*KFG Österreich:*

Tabelle 5-3 zeigt die für die Aufgabenstellung relevanten, gesetzlichen Rahmenbedingungen.

**Tabelle 5-3: Gesetzliche Bestimmungen laut KFG**

Reglementierung	Wert	Paragraph
Höchstzulässiges Gesamtgewicht	40.000 kg	KFG § 4, Abs. 7a
Höchstzulässige Achslast (angetrieben)	11.500 kg	KFG § 4, Abs. 8
Höchstzulässige Achslast (Doppelachse mit weniger als 1m Radstand)	11.500 kg	KFG § 4, Abs. 8b
Höchstzulässige Achslast (Doppelachse mit 1m bis weniger 1,3m Radstand)	16.000 kg	KFG § 4, Abs. 8b
Höchstzulässige Achslast (Dreifachachse mit weniger 1,3m Radstand)	21.000 kg	KFG § 4, Abs. 8b

Des Weiteren besagt KFG § 4 Abs.9a:

*„Bei Lastkraftwagen, Sattelkraftfahrzeugen und Lastkraftwagen mit Anhängern darf das Gewicht auf der oder den Antriebsachsen nicht weniger als 25 vH des Gesamtgewichtes des Fahrzeuges oder der Fahrzeugkombination betragen“*

Die gesetzlichen Bestimmungen, kombiniert mit den Spezifikationen der betrachteten Fahrzeugkombination ergeben somit die in Tabelle 5-4 dargestellten Rahmenbedingungen.

**Tabelle 5-4: Gesetzliche Rahmenbedingungen der betrachteten Fahrzeugkombination**

Name	Höchst zulässige Sattellast [kg]	Höchst zulässige Achslast [kg]	Höchst zulässiges Gesamtgewicht [kg]	Höchst zulässiges Ladungsgewicht [kg]
<b>Wert</b>	11.500	16.000	40.000	24.000

*VDI 2700 [VDI04]:*

Die VDI 2700 wurde in der Erstfassung im Oktober 1975 präsentiert und gilt als anerkannte, technische Richtlinie für Ladungssicherung im Lastverkehr. Neben Ladungssicherungsmaßnahmen beschreibt diese Richtlinie auch die Erstellung eines Lastverteilungsplanes. Die Richtlinie ist in Österreich und im gesamten EU-Raum gültig und gilt als anerkannte technische Regel, welche auch im Falle von Gerichtsstreitigkeiten als Referenz zur Anwendung kommt. Für diese Arbeit interessiert vor allem die Erstellung des Lastverteilungsplans.

Dieser berücksichtigt folgende rechtliche und technische Restriktionen (vgl. [VDI04]):

- Höchst zulässige Vorderachslast
- Höchst zulässige Nutzlast
- Höchst zulässige Hinterachslast

- Mindestlenkachslast (variiert je nach Fahrzeug zwischen 20% und 35% des Fahrzeugmomentangewichts)

Abbildung 5-4 zeigt den in der VDI 2700 geregelten Lastverteilungsplan.

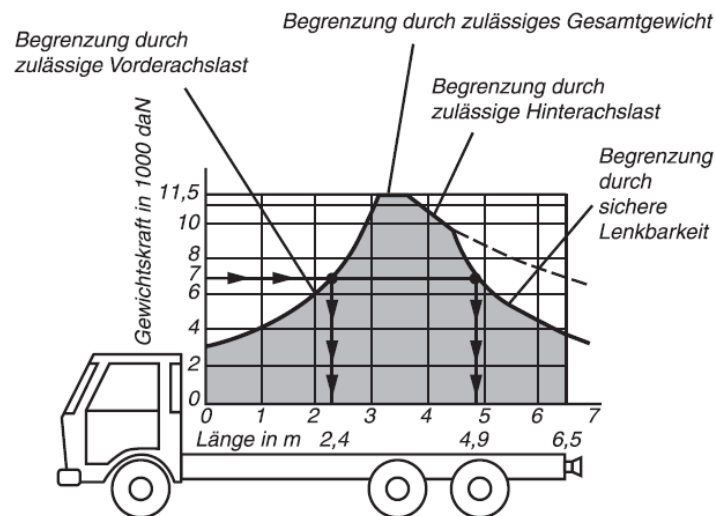
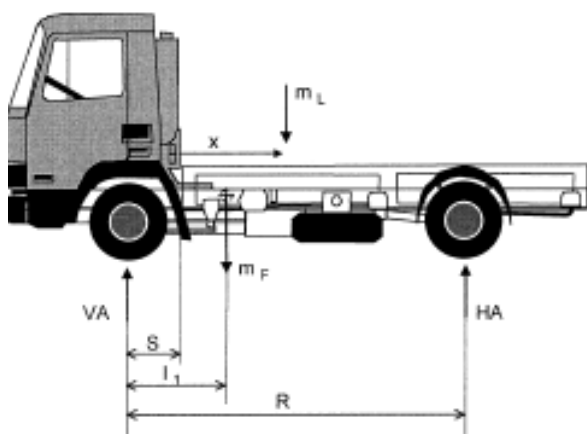


Abbildung 5-4: Lastverteilungsplan gemäß VDI 2700 [VDI00]

Die Berechnung der Kurvenverläufe ist im Weiteren von Interesse und wird daher hier erläutert:

Die für die Berechnung des Lastverteilungsplans notwendige Notation ist in Abbildung 5-5 dargestellt.



Formelzeichen	Dimension
Masse des Fahrzeugs leer	= $m_F$ kg
Masse der Zuladung	= $m_L$ kg
Vorderachslast Fahrzeug leer	= $VA_{leer}$ kg
Vorderachslast Fahrzeug vollbeladen	= $VA_{voll}$ kg
Hinterachslast Fahrzeug leer	= $HA_{leer}$ kg
Hinterachslast Fahrzeug vollbeladen	= $HA_{voll}$ kg
Radstand	= $R$ m
Abstand Vorderachse – Ladefläche	= $S$ m
Schwerpunktlage Fahrzeug leer	= $l_1$ m
Sichere Lenkbarkeit <sup>1)</sup>	= $SL$ %
Sichere Traktion <sup>1)</sup>	= $ST$ %
Beschreibung eines Punktes der jeweiligen zu berechnenden Kurve	= $m_y$ kg
Laufvariable ab Beginn der Ladefläche	= $x$ m

Abbildung 5-5: Notation zur Berechnung des Lastverteilungsplans [VDI00]

Zur Bestimmung der Schwerpunktlage  $l_1$  des leeren Fahrzeugs wird das Momentengleichgewicht um die Vorderachse gebildet.

$$l_1 = \frac{HA_{leer} \cdot R}{m_F} \quad \text{Gl. 5-1}$$

Die Kurvenverläufe werden zur besseren Übersicht in Abbildung 5-6 benannt und deren Berechnung im Folgenden erklärt.

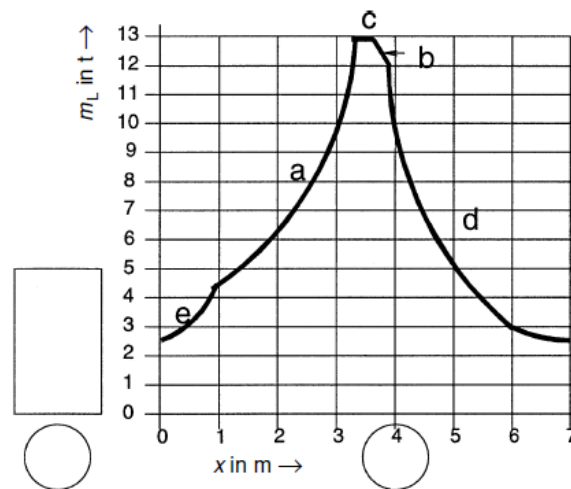


Abbildung 5-6: Kurvenverläufe im Lastverteilungsplan [VDI00]

Die Berechnung des Kurvenverlaufs „a“ basiert auf der Annahme, dass die Summe aller Momente um die Hinterachse gleich null ist. Begrenzt wird „a“ von der vom Fahrzeughersteller festgelegten maximal zulässigen Vorderachslast. Daraus ergibt sich für die Masse der Zuladung  $m_L$ :

$$m_L = \frac{VA_{Voll} \cdot R - m_F \cdot (R - l_1)}{R - S - x} \quad \text{Gl. 5-2}$$

Wobei  $x$  eine laufende Koordinaten, beginnend bei der Stirnwand der Ladungsfläche, darstellt.

Der Verlauf der Kurve „b“ basiert auf der Annahme, dass die Summe aller Momente um die Vorderachse gleich null ist. Dieser wird von der maximal zulässigen Hinterachslast limitiert. Das Momentengleichgewicht um die Vorderachse liefert somit:

$$m_L = \frac{HA_{Voll} \cdot R - m_F \cdot l_1}{S + x} \quad \text{Gl. 5-3}$$

Beide Kurven „a“ und „b“ werden durch die höchst zulässige Nutzlast beschränkt. Diese stellt Kurve „c“ dar.

Der Kurvenverlauf „d“ für die Mindestlenkachslast wird wiederum mit dem Momentengleichgewicht um die Hinterachse wie folgt berechnet:

$$m_L = \frac{l_1 - (1 - SL) \cdot R}{(1 - SL) \cdot R - S - x} \quad \text{Gl. 5-4}$$

Die Mindesthinterachslast, in Abbildung 5-6 mit „e“ gekennzeichnet, beginnt bei der fahrerseitigen Stirnseite der Ladefläche und schneidet Kurve „a“. Berechnet wird diese wie folgt:

$$m_L = \frac{l_1 - ST \cdot R}{ST \cdot R - S - x} m_F \quad \text{Gl. 5-5}$$

In der VDI 2700 wird aber angemerkt, dass in den meisten praktischen Anwendungen die erforderliche Hinterachslast erreicht wird und der Kurvenverlauf „e“ somit häufig nicht im relevanten Bereich liegt.

Wenn der Lastverteilungsplan für das Fahrzeug berechnet ist, muss die Ladung so aufgeteilt werden, dass der Ladungsschwerpunkt innerhalb dieser Kurve liegt (vgl. [VDI00] und [VDI04]).

*Praxisanwendung von Transportunternehmen:*

Als drittes Beladungsmodell wird auch die in der Praxis gängige Methode betrachtet. Um diese „Packregeln“ zu erfassen, wurde ein intensiver Austausch mit einem langjährigen Spediteur gepflegt. Konkret handelt sich hierbei um Hr. Johann Pucher, geschäftsführender Gesellschafter der Johann Pucher GesmbH [PUC14]. 30-jährige Berufserfahrung als LKW-Lenker und anschließend erfolgreicher, selbständiger Spediteur unterstreichen seine Qualifikationen als Quelle zur Definition der praxisrelevanten Rahmenbedingungen.

Für den Sattelaufleger, der in dieser Arbeit als Standardmodell referenziert wird, gelten demnach folgende, aus langjähriger Berufserfahrung resultierende Packregeln (vgl. [PUC14]):

- Auf die erste Hälfte des Sattelauflegers (Stellplätze 1-15 in Abbildung 5-8) werden maximal 11.000 kg Ladung gepackt
- Auf die zweite Hälfte (Stellplätze 16-33) wird die Differenz zwischen maximal erlaubten Ladungsgewicht und 11.000 kg gepackt. Im vorliegenden Fall 13.000kg
- Das Gewicht auf der ersten Hälfte soll möglichst homogen über die Länge des Aufliegers verteilt sein, wobei die erste Reihe (Stellplätze 1-3) wenn möglich leichter sein soll als die übrigen der ersten Hälfte, um ein Durchbiegen im abgesattelten Zustand zu minimieren und Materialermüdung zu verringern.
- Spalte 2 (Stellplätze 4-6) liegt direkt über der Sattelkupplung und leitet somit die Gewichtskraft direkt in die Rahmenkonstruktion der Zugmaschine ohne ein Biegemoment aufzuprägen. Analog dazu liegt Spalte 9 (Stellplätze 25-27) über der Hinterachse.
- Das Gewicht, das auf der zweiten Anhängerhälfte lastet, soll sich im Bereich zwischen Anhängermitte und Achsen konzentrieren, wobei über es den Achsen am günstigsten ist.
- Die beiden hintersten Reihen (Stellplätze 18-33), vor allem aber die letzte (Stellplätze 31-33), sollen so leicht wie möglich sein um eine Mindestsattellast zu erreichen und somit Lenkbarkeit und Traktion der Zugmaschine garantieren zu können. Dadurch wird auch die konvexe Durchbiegung verringert und somit einer frühzeitigen Materialermüdung entgegen gewirkt.
- Da die Straßen so gebaut sind, dass sie leicht nach rechts neigen um stehendes Wasser zu vermeiden, soll der Schwerpunkt der Last in Anhängerrichtung auf der linken Seite (Fahrerseite) liegen um den Reifenverschleiß zu minimieren. Dieses Kriterium wird auch durch die Tatsache unterstrichen, dass die Treibstofftanks an der Zugmaschine im Regelfall auf der linken Seite größer sind als auf der rechten.
- Im Winter oder bei nassen Fahrbahnverhältnissen ist besonders darauf zu achten, dass genügend Last auf den angetriebenen Achsen lastet. Dafür kann auch die



gesetzliche Restriktion der maximalen Achslast für angetriebene Achsen gebrochen und eine Strafe in Kauf genommen werden.

- Das Einhalten der Mindestachslasten der nicht angetriebenen Achsen ist in der Praxis nur in Sonderfällen ein auftretendes Problem.
- Prinzipiell sollte die Ladung so verteilt werden, dass die konvexe Durchbiegung des Anhängers minimiert wird.
- Das Überschreiten der maximalen Achslasten wird bis 200kg toleriert.
- Die maximale Höchstlast von Zugmaschine und Aufleger wird mit 4% toleriert

#### Übersicht Restriktionen:

Aus der Beschreibung der beiden limitierenden Instanzen VDI 2700 und KFG geht hervor, dass diese in keinem Widerspruch zu einander stehen und die VDI 2700 mit dem Ziel geschaffen wurde, das Einhalten der gesetzlichen Restriktionen zu erleichtern. Sehr wohl zeigen sich, wie in Tabelle 5-5 als Übersicht dargestellt, Diskrepanzen bei den berücksichtigten Faktoren zwischen VDI 2700 bzw. KFG Österreich und den Packrichtlinien aus der Praxis. Wie die Ergebnisse aus Kapitel 5.3 zeigen, erzeugen die Packrichtlinien aus der Praxis Ergebnisse, die den Restriktionen von VDI 2700 bzw. KFG genügen, aber berücksichtigen zusätzlich die Faktoren Materialermüdung und Reifenverschleiß. Bei speziellen Witterungsverhältnissen wird jedoch, um die Traktions- und Lenkfähigkeit des LKW zu erhalten, die maximale Sattellast bzw. höchst zulässige Achslast (angetrieben) überschritten. Dieser Umstand wird als Sonderfall behandelt, ihm aber durch die dynamische Implementierung der Gewichtsgrenzen Rechnung getragen.

**Tabelle 5-5: Berücksichtigte Faktoren bei VDI 2700/KFG und Praxisanwendung - Übersicht**

Berücksichtigter Faktor	VDI 2700/KFG	Praxisanwendung
<b>Traktionsfähigkeit (Minimale Achslast –angetrieben und nicht angetrieben)</b>	Ja	Ja
<b>Witterungsverhältnisse</b>	Nein	Ja
<b>Materialermüdung</b>	Nein	Ja
<b>Reifenverschleiß</b>	Nein	Ja
<b>Ladungssicherung</b>	Ja	Ja

Da diese Arbeit versucht, technische Rahmenbedingungen und Restriktionen zu berücksichtigen, werden die Packrichtlinien aus der Praxis im Balanced Trailer Load-Algorithmus implementiert.

### 5.1.3 Methoden

Die Gattung der Problemstellung ist den „Zuordnungsproblemen“ einzuordnen. Zuordnungsprobleme werden in der Literatur jene Probleme genannt, die eine Abbildung der Menge der Nachfrager auf die Menge der Anbieter unter Einbehaltung von Rahmenbedingungen abbilden, wie in Abbildung 5-7 gezeigt. Hierbei gibt es eine disjunkte Menge an Anbietern und Nachfragern. Rahmenbedingungen können unterschiedlich gewichtet sein, wie die in Abbildung 5-7 gezeigten Zahlen bei den Verbindungslinien.

Ziel der mathematischen Abbildung der Menge der Nachfrager auf die Menge der Anbieter ist es, möglichst wenige Rahmenbedingungen zu verletzen und die Verletzungen multipliziert mit deren Gewichtung minimal werden zu lassen.

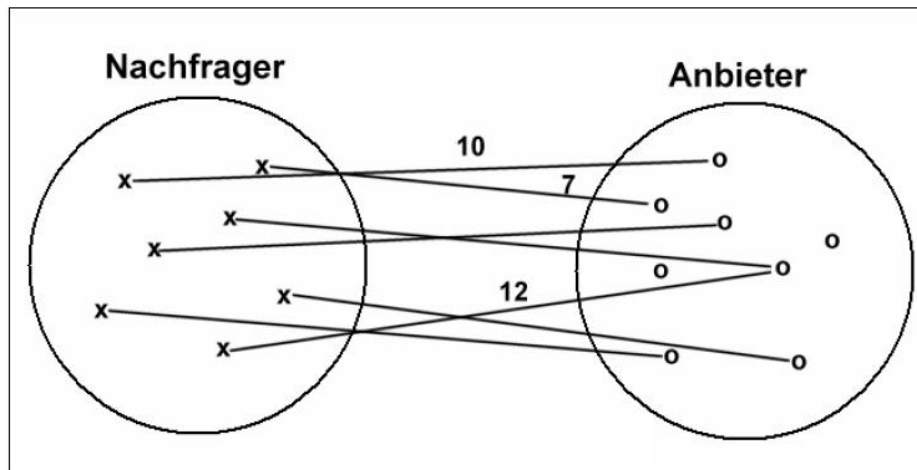


Abbildung 5-7: Veranschaulichung Zuordnungsprobleme [HER00]

Man unterscheidet „harte“ Randbedingungen, das sind jene die unbedingt erfüllt werden müssen, und „weiche“ Randbedingungen. „Weiche“ Randbedingungen müssen nicht zwingend erfüllt werden, dienen aber als Optimierungskriterium.

Das hier vorliegende Zuordnungsproblem ist, wie das in Kapitel 4.1.3 beschriebene 2D Vector Bin Packing Problem, NP-vollständig. Das bedeutet, dass das Finden einer exakten Lösung im angemessenen Zeitrahmen nicht möglich ist (vgl. [HER00]).

Herrler identifiziert in [HER00] zum Lösen von Zuordnungsproblemen folgende Lösungsmethoden:

- Exakte mathematische Verfahren
  - Findet die beste Lösung von eins- zu- eins- Problemen in polynomieller Zeit
  - Zum Beispiel „Ungarische Methode“ oder „Busaker-Goven Algorithmus“
- Mathematische Näherungsverfahren
  - Am häufigsten verwendete Verfahren im Operations Research
  - Häufig in ganzzahlige lineare und nichtlineare Programmen implementiert
  - Vielfältige Algorithmen in großer Zahl vorhanden
- Genetische Algorithmen
  - Bilden ein evolutionäres Verhalten einer Population einer Lösung nach. Eine initiale Menge an Lösungen wird mittels Mutation, Crossover und Selektion angewandt
- Suchverfahren
  - Intuitive Lösungssuche nach vorgegebenen Zielkriterien
  - Vollständige Suchverfahren benötigen genauso wie exakte Lösungsmethoden exponentielle Laufzeit
- Vorschlagen und Vertauschen
  - Heuristisches, approximatives Lösungsverfahren, das schrittweise eine Initiale Lösung zu optimiert versucht
  - Besitzt Anytime-Eigenschaft. Das heißt dass die Qualität der Lösung von der benötigten Rechenzeit abhängt
  - Unterbrechungseigenschaft: Durch die inkrementelle Lösungseigenschaft kann der Lösungsprozess vom Benutzer jederzeit unterbrochen werden und trotzdem eine Lösung erhalten
  - Randbedingungen können beliebig komplex sein

## 5.2 Umsetzung

Basierend auf der Recherche in Kapitel 5.1 wird hier ein Lösungsalgorithmus vorgestellt, der geeignet ist sowohl den in VDI 2700 dargestellten Packplan zu erfüllen oder, durch die dynamische Implementierung der Grenzwerte, den Praxisbedingungen zu folgen. Eingebettet ist die Lösungsmethode in einen VB.net Code, der Anfangs- und Rahmenbedingungen an Microsoft EXCEL® 2013 „evolutionary“ Solver übergibt und die Lösungen wieder übernimmt. Abbildung 5-9 zeigt, wie die Lösungsmethode in das Programm implementiert wurde.

### 5.2.1 Balanced Trailer Load - Algorithmus und Implementierung

#### Notation

Sattelanhängereinteilung und Koordinaten:

1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9	1-10	1-11
2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9	2-10	2-11
3-1	3-2	3-3	3-4	3-5	3-6	3-7	3-8	3-9	3-10	3-11

Abbildung 5-8: UL Standplatz-Notation

$i$  = Index für Positions-Zeile

$h$  = Index für Positions-Spalte

$j$  = Index für Unit Load

$y_{ih}^j$  = Entscheidungsvariable für UL  $j$  platziert auf Position  $ih$

$x_{ih}$  = Entscheidungsvariable für Position  $ih$

$M$  = Anzahl der UL

$W$  = Gesamtgewicht der UL

$w_j$  = Gewicht von UL  $j$

$c^v$  = Volumenkapazität des Trailers

$c^w$  = Gewichtskapazität des Trailers

$COM_x$  = Schwerpunkt in  $x$ -Richtung

$COM_y$  = Schwerpunkt in  $y$ -Richtung

$L_{FW}^{Max}$  = Maximale Sattellast

$L_A^{Max}$  = Maximale Achslast

$L_{FW}^{Min}$  = Minimale Sattellast

$L_A^{Min}$  = Minimale Achslast

**Sätze:**

$$j \in L \quad \text{wobei } |L| = M; j \in \mathbb{N}^+ \quad \text{Gl. 5-6}$$

$$i \in T \quad i \in \mathbb{N}^+ \{1 \dots 3\} \quad \text{Gl. 5-7}$$

$$h \in Z \quad h \in \mathbb{N}^+ \{1 \dots 11\} \quad \text{Gl. 5-8}$$

$$COM_Y, \quad COM_X, \quad L_{FW}^{max}, \quad L_A^{max} \in \mathbb{R} \quad \text{Gl. 5-9}$$

$$COM_X \geq 0 \quad COM_X \in \mathbb{R}^+ \quad \text{Gl. 5-10}$$

**Limits**

$$L_{FW}^{min} \leq \sum_{i=1;h=1;j=1}^{i=3;h=5;j=M} y_{ih}^j * w_j \leq L_{FW}^{max} \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-11}$$

$$L_A^{min} \leq \sum_{i=1;h=7;j=1}^{i=3;h=11;j=M} y_{ih}^j * w_j \leq L_A^{max} \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-12}$$

$$\sum_{i=1;h=1;j=1}^{i=3;h=1;j=M} y_{ih}^j * w_j \leq \sum_{i=1;h=2;j=1}^{i=3;h=2;j=M} y_{ih}^j * w_j \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-13}$$

$$\sum_{i=1;h=1;j=1}^{i=3;h=1;j=M} y_{ih}^j * w_j \leq \sum_{i=1;h=3;j=1}^{i=3;h=3;j=M} y_{ih}^j * w_j \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-14}$$

$$\sum_{i=1;h=11;j=1}^{i=3;h=11;j=M} y_{ih}^j * w_j \leq \sum_{i=1;h=10;j=1}^{i=3;h=10;j=M} y_{ih}^j * w_j \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-15}$$

$$COM_Y \leq 0 \quad COM_Y \in \mathbb{R} \quad \text{Gl. 5-16}$$

**Parameter**

Jedes zu lösende Problem kann in folgenden Parametern variieren:

$$L(M, W, \{w_j\}) \text{ \& } T(c^w, L_{FW}^{max}, L_{FW}^{min}, L_A^{max}, L_A^{min})$$

Zum Beispiel

1 UL mit 300 kg/UL

8 ULs mit 500 kg/UL

19 UL mit 800 kg/UL

5 UL mit 900 kg/UL

$M = 33$

$W = 24.000 \text{ kg}$

$w_j \in \{150, 500, 700\}$

$c^w = 24.000 \text{ kg/LKW}$

$L_{FW}^{Max} = 11.500 \text{ kg}$

$L_A^{Max} = 16.000 \text{ kg}$

$L_{FW}^{Min} = 0 \text{ kg}$

$L_A^{Min} = 0 \text{ kg}$

$L_A^{Min} = 0 \text{ kg}$

**Entscheidungs-Variablen**

Der Algorithmus entscheidet, welche UL auf welchen Stellplatz im Trailer platziert wird

$$y_{ih}^j = 0 \text{ oder } 1 \quad y_{ih}^j = \begin{cases} 1 & \text{wenn UL } j \text{ platziert wird auf Position } ij \\ 0 & \text{sonst} \end{cases} \quad \text{Gl. 5-17}$$

$$j = 1, \dots, M$$

$$i = 1, \dots, 3$$

$$h = 1, \dots, 11$$

$$x_{ih} = 0 \text{ oder } 1 \quad x_i = \begin{cases} 1 & \text{wenn Position } ih \text{ verwendet wird} \\ 0 & \text{sonst} \end{cases} \quad \text{Gl. 5-18}$$

$$i = 1, \dots, 3$$

$$h = 1, \dots, 11$$

**Ziel**

Das Ziel des Algorithmus ist die Schwerpunktlage in x-Richtung zu minimieren

$$\text{minimiere } COM_x \quad \text{Gl. 5-19}$$

**Einschränkungen**

$$\sum_{i=1;h=1;j=1}^{i=3;h=11;j=M} y_{ih}^j \equiv M \quad \forall j, \forall i, \forall h \quad \text{Gl. 5-20}$$

## Volumen-Einschränkung

$$\sum_{i=1;h=1;j=1}^{i=3;h=11;j=M} y_{ih}^j \leq c^v \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-21}$$

## Gewichts-Einschränkung

$$\sum_{i=1;h=1;j=1}^{i=3;h=11;j=M} y_{ih}^j * w_j \leq c^w \quad \forall i; \forall h; \forall j \quad \text{Gl. 5-22}$$

$$x_{ih}, y_{ih}^j \in \{0,1\} \quad \forall i, h; \forall i, h, j \quad \text{Gl. 5-23}$$

### Implementierung

Die Abbildung 5-9 zeigt die Implementierung in die Programmumgebung. Eine Legende zur verwendeten Prozessnotation ist in Abbildung 8-1 dargestellt.

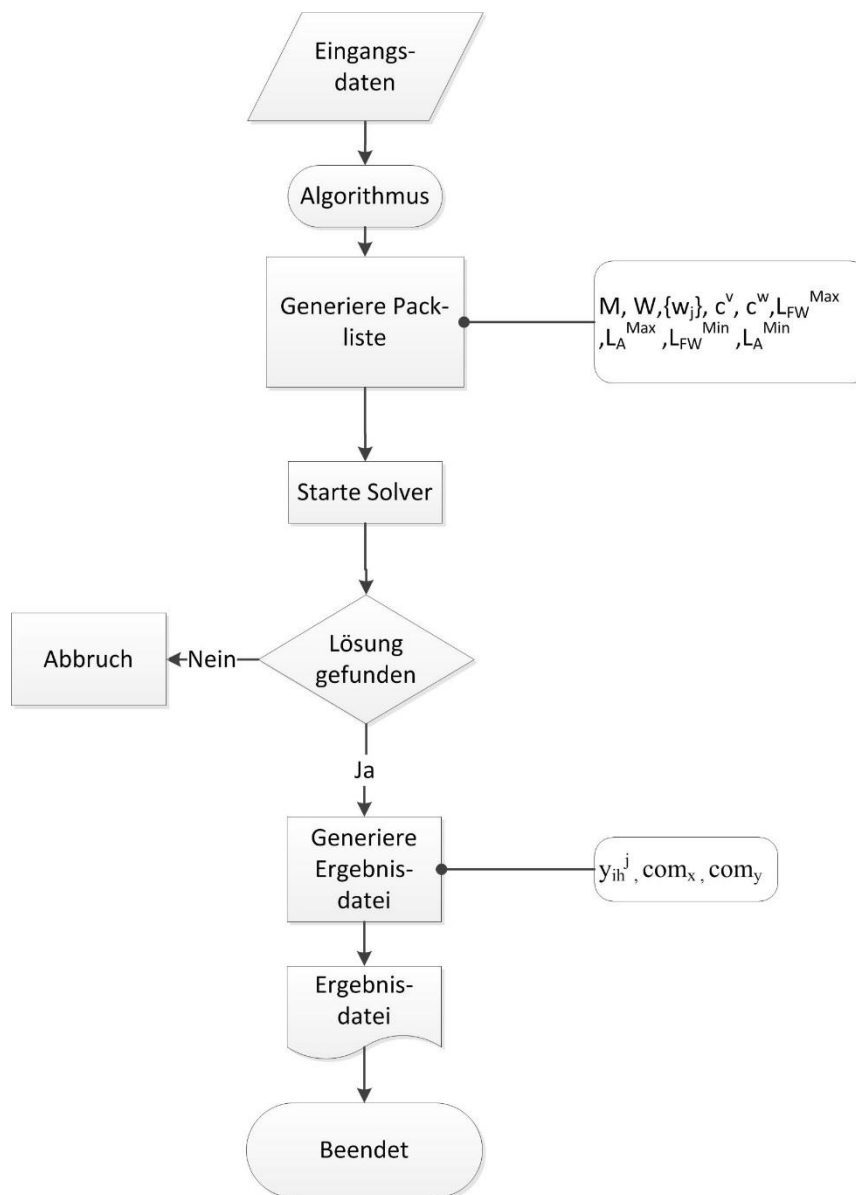


Abbildung 5-9: Implementierung in Programmumgebung

### 5.2.2 Begründung

Wie aus der Zielformulierung aus Kapitel 5.1.1 hervorgeht, dient der Balanced Trailer Load Algorithmus der Verifizierung der Ergebnisse des Unit Load Packers. Dies bedeutet, dass UL, welche dem LKW zugeordnet werden, auf diesem auch so platziert werden können, dass dieser auch gemäß StVO und KFG transportieren darf. Von besonderer Bedeutung ist die Überprüfung bei jenen Ladungen, deren untere Schranke vom Gewicht bestimmt wird und folglich ein durchschnittliches UL-Gewicht von über 727,27 kg aufweist.

Da diese Aufgabenstellung über sehr viele Randbedingungen verfügt und nicht immer ein 1-zu-1-Zuordnungsproblem vorliegt, da auch mehr Stellplätze als UL „angeboten“ werden können, wurde die Lösungsmethode in eine Solverarchitektur implementiert, der dem

Prinzip des „Vorschlagen und Vertauschen“ folgt. Dieser erfüllt jene Vorteile, die speziell für diese Aufgabenstellung tragend sind:

- Die Qualität der Lösung hängt von der zur Verfügung gestellten Berechnungsdauer ab. Da aber keine optimale, sondern nur eine gültige Lösung gesucht wird, ist diese ausreichend.
- Die Berechnung kann unterbrochen werden und trotzdem eine funktionierende Lösung anbieten. Das bedeutet, dass gewissen Ladungsklassen (Anzahl UL, mittleres Gewicht, Streuung) eine bestimmte Berechnungsdauer zugewiesen werden kann. Somit wäre dies ein effizientes und geeignetes Werkzeug für den laufenden Distributionsbetrieb.
- Der Lösungsalgorithmus ist unabhängig von den Randbedingungen. Da diese alle dynamisch implementiert sind und somit vom Benutzer gewählt werden können, wird dadurch Gestaltungsfreiheit gewonnen und es können sowohl Packordnungen gemäß VDI 2700 als auch nach eigener Präferenz erzeugt werden.

Da die Recherche ergab, dass die Packregeln nach VDI 2700 von jenen der Praxis variieren, wurden die einzelnen Gewichtsrestriktionen dynamisch in den Algorithmus implementiert und somit kann beiden Prinzipien gefolgt werden. Als primäre Richtlinie wurde aber die in der Praxis etablierte Methode gewählt, da diese erprobt ist und funktioniert. Die Ergebnisse können in späterer Folge verglichen werden.

## **5.3 Ergebnisse der Vorgehensweise**

Basierend auf den Ergebnissen des Unit Load Packer, dargestellt in Tabelle 4-9, für den Beispieldatensatz, wurden die in Kapitel 5.3.1 gezeigten Ergebnisse anhand der Packregeln aus der Praxis erreicht.

### **5.3.1 Ergebnisse**

In Tabelle 5-6 wird die Übersicht der Ergebnisse dargestellt. Mit „Last Spalte 1-5“ wird jene Last bezeichnet, die in der vorderen Hälfte des Anhängers platziert wird, „Last Spalte 7-11“ bezeichnet die hintere Hälfte gemäß der in der Praxis etablierten Packregeln. Spalte 6 ist in der geometrischen Mitte des Anhängers und hier besitzen sowohl LKW Fahrer als auch der implementierte Packalgorithmus einen erweiterten Entscheidungsspielraum. Um die Ergebnisse mit der VDI 2700 vergleichen zu können, werden sowohl Sattel- als auch Achslast gemäß der in [VDI00] vorgeschriebenen Berechnungsmethodik anhand des Momentengleichgewichts um Vorder- bzw. Hinterachse des Sattelanhängers berechnet und dargestellt. Dabei liegt die Vorderachse direkt unter Spalte 2 und die Hinterachsen unter Spalte 9.

Abbildung 5-10 zeigt zur besseren Veranschaulichung die relative Lage der UL am LKW sowie die betrachteten LKW Spezifikationen. Aus der Abbildung gehen die geometrische Mitte, Schwerpunktlage sowie die Lage von Sattelpunkt und Achsen hervor.



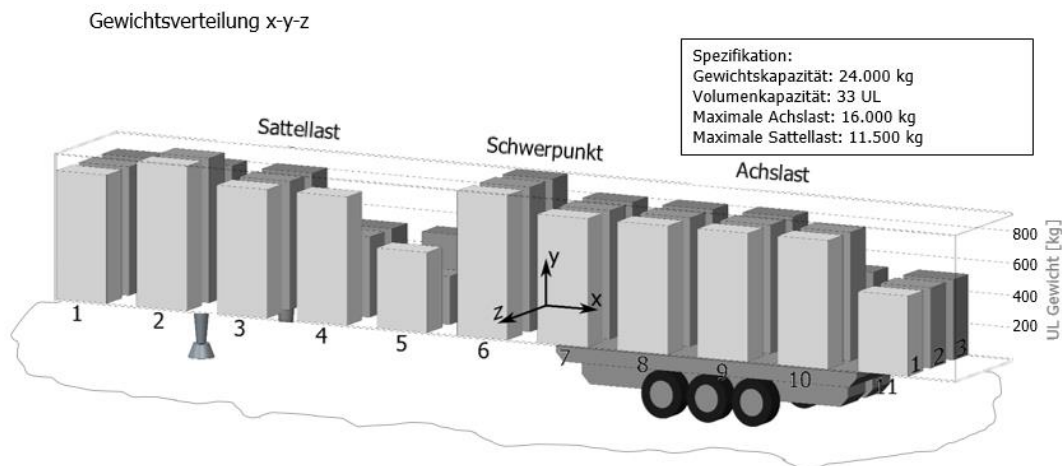


Abbildung 5-10: Darstellung der relativen Lage der UL am LKW (vgl. [LJE14])

Abbildung 5-11 bis Abbildung 5-19 zeigen die Detailergebnisse des „Balanced Trailer Load“ – Algorithmus, welche in Tabelle 5-6 zusammengefasst dargestellt werden. Die Berechnungsdauer betrug immer weniger als 2 Minuten.

Tabelle 5-6: Ergebnisse des Beispieldatensatzes

Datensatz	Last Spalte 1-5 [kg]	Last Spalte 7-11 [kg]	Last Spalte 6 [kg]	Sattellast nach VDI 2700 [kg]	Achslast nach VDI 2700 [kg]	Schwerpunkt in x [m]	Schwerpunkt in y [m]
Beispiel, LKW1	10.900	11.000	2.100	10.285	13.715	0,000	-0,040
Beispiel, LKW2	6.400	6.300	1.500	6.085	8.115	0,000	-0,011
Beispiel, LKW3	600	600	300	642	858	0,000	-0,240

Die Darstellungen „Gewichtsverteilung“ zeigen, wie die UL am LKW angeordnet wurden, wobei die Höhe der Säulen das jeweilige Gewicht repräsentiert. Die Diagramme „kumuliertes Ladungsgewicht/Spalte“ zeigen das summierte Gewicht der UL pro Spalte. In den Diagrammen „kumulierter Ladungsgewicht/Reihe“ zeigen analog dazu das summierte Gewicht der UL pro Reihe.

Beispiel, LKW1:

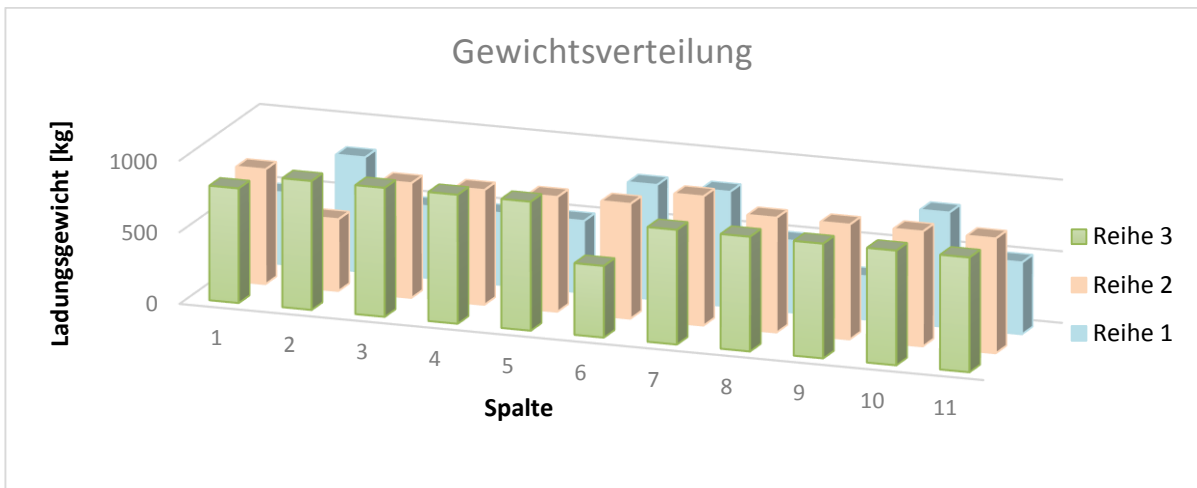


Abbildung 5-11: Gewichtsverteilung LKW 1

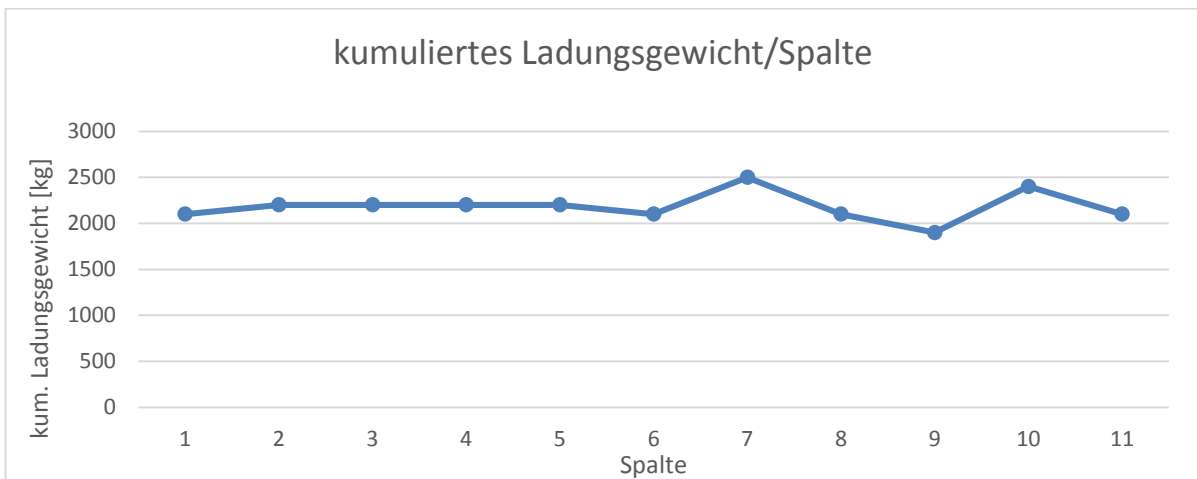


Abbildung 5-12: Gewichtsverteilung entlang der x-Achse LKW 1

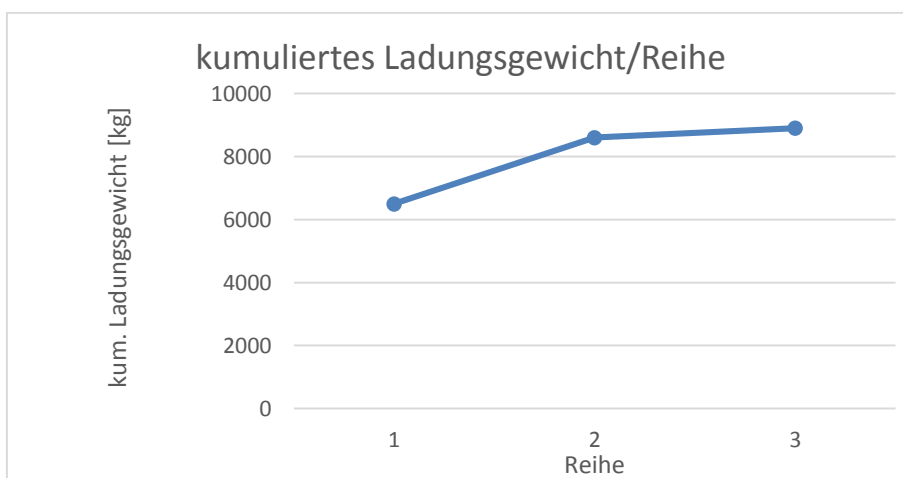


Abbildung 5-13: Gewichtsverteilung entlang der y-Achse LKW 1

Beispiel, LKW2:

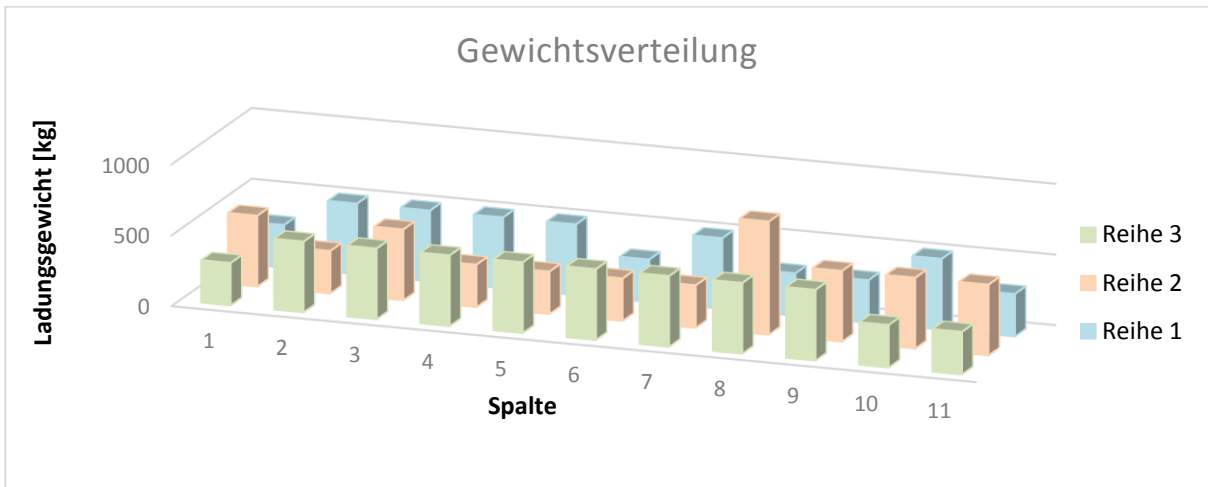


Abbildung 5-14: Gewichtsverteilung LKW 2

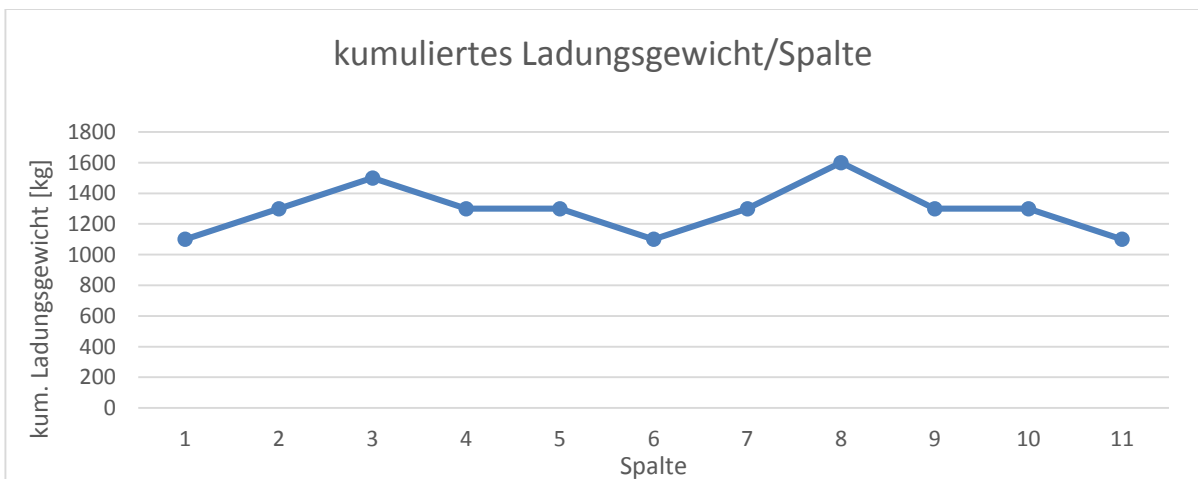


Abbildung 5-15: Gewichtsverteilung entlang der x-Achse LKW 2

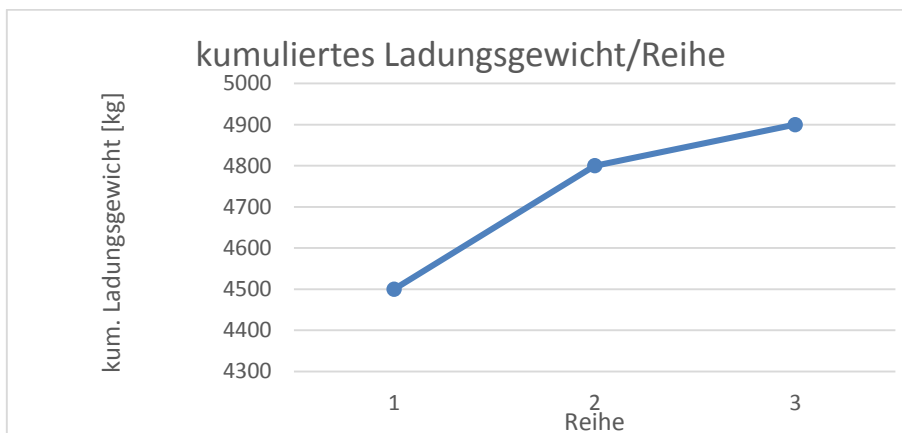


Abbildung 5-16: Gewichtsverteilung entlang der y-Achse LKW 2

Beispiel, LKW3:

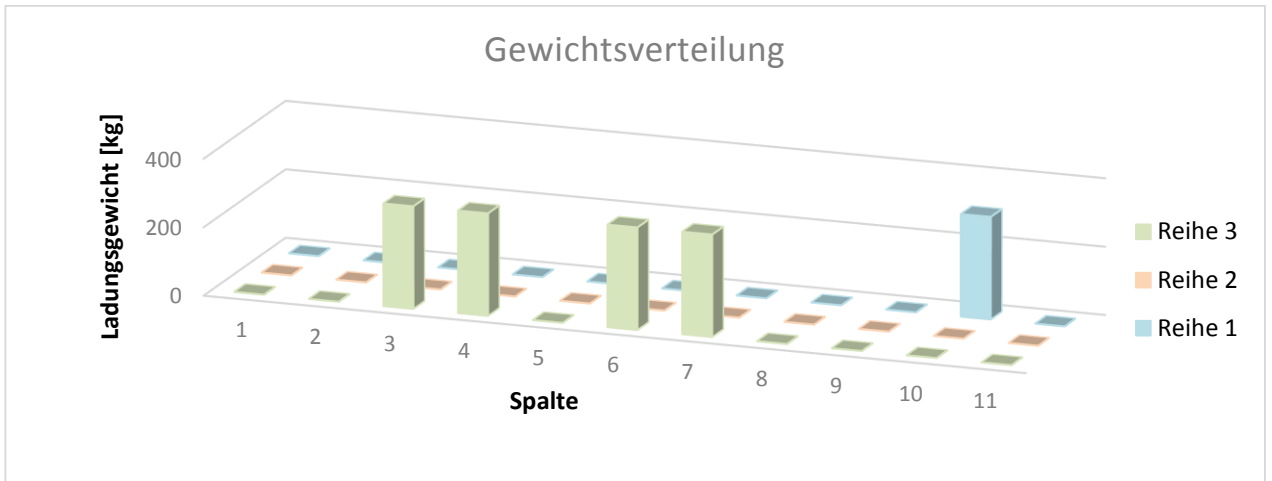


Abbildung 5-17: Gewichtsverteilung LKW 3

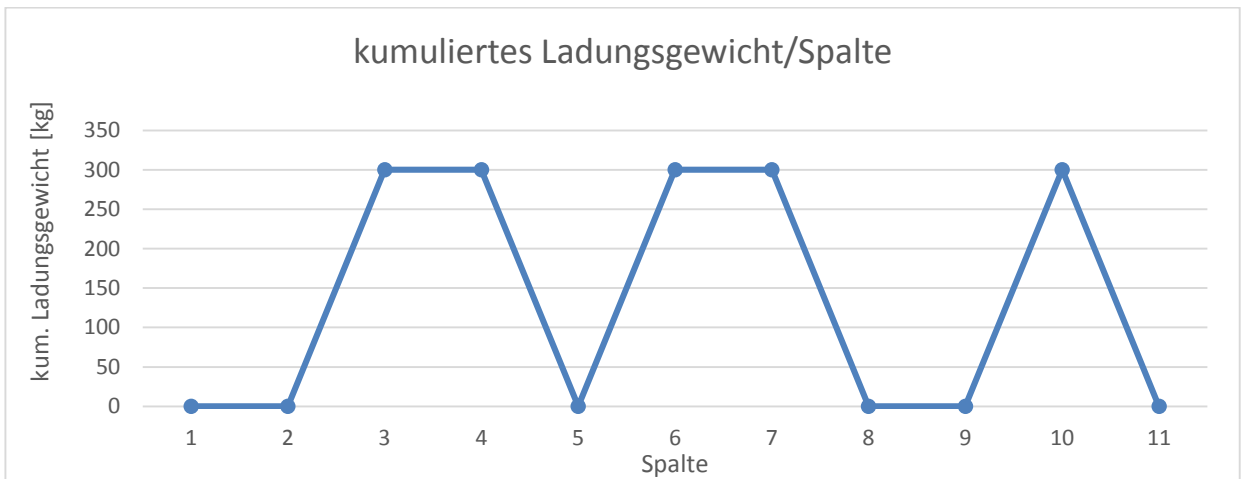


Abbildung 5-18: Gewichtsverteilung entlang der x-Achse LKW 3

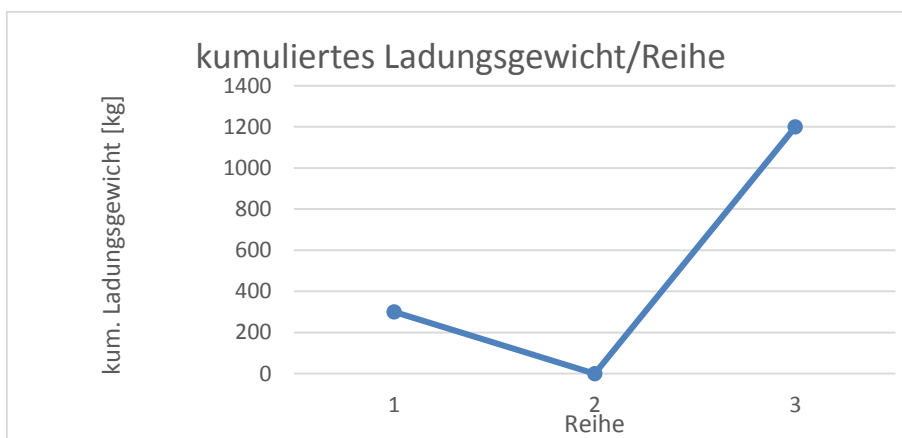


Abbildung 5-19: Gewichtsverteilung entlang der y-Achse LKW 3

## 5.4 Bewertung/Beurteilung der Vorgehensweise

### *Implementierte Packregeln:*

Die Recherche ergab, dass die Packregel gemäß VDI 2700 in der Praxis wenig etabliert ist und die Packordnung eher nach persönlicher Erfahrung und „Intuition“ des für die Lastverteilung zuständigen LKW Fahrers erstellt wird. Folgende, wesentliche Abweichungen zwischen der etablierten Praxis und VDI 2700 wurden in dieser Arbeit identifiziert:

- Die Fahrbahnverhältnisse ändern sich mit den Jahreszeiten und dem aktuellen Wetter und erfordern somit eine variierende Gestaltung der Sattellast um genügend Traktion zu gewährleisten. Fallweise muss dafür sogar die gesetzlich erlaubte Höchstlast überschritten werden
- Die Schwerpunktlage in y-Richtung (gemäß der Notation in Abbildung 5-8) soll, um Reifenverschleiß zu minimieren, leicht negativ sein. Gemäß VDI 2700 soll dieser jedoch möglichst mittig liegen

Die Ergebnisse in Kapitel 5.3.1 zeigen aber, dass jene Packordnung, die durch Anwendung der in der Praxis etablierten Regeln entsteht, auch der VDI 2700 genügt, wenn die höchst zulässigen Lasten gemäß der gesetzlichen Restriktionen dem Algorithmus übergeben werden.

Dies äußert sich durch die Tatsache, dass die erlaubten Höchstlasten für angetriebene und nicht angetriebene Achsen gemäß der Berechnungsmethodik nach VDI 2700 nicht überschritten bzw. Mindestlasten unterschritten werden. Des Weiteren liegt der Schwerpunkt der Ladung in x-Richtung, auf Grund der harten Algorithmus-Randbedingungen, immer innerhalb der in Abbildung 5-6 gezeigten Berandungskurve womit die gefundene Lösung der Richtlinie entspricht. Zwar liegt der Schwerpunkt der „Balanced Trailer Loading“ – Lösung immer im negativen y-Bereich, aber ausreichend nahe an der Fahrzeugmitte um der VDI 2700 zu genügen.

Der „Balanced Trailer Loading“ Algorithmus wurde mit dem Ziel entwickelt die Realisierbarkeit der Ergebnisse vom UL-Zuteilungs-Algorithmus des „Unit Load Packers“ zu verifizieren. Auf Grund der großen Anzahl an benötigten LKW in den Datensätzen, die in Kapitel 4.3.1 präsentiert wurden, war es nicht möglich jedes Detailergebnis zu überprüfen. Anhand der Ergebnisse des repräsentativen Beispieldatensatzes und der intensiven Tests während der Entwicklung ist aber die Annahme gerechtfertigt, dass die Lösungen der anderen Datensätze realisierbar sind.

Funktionen und Anforderungen an die M-Boxen, die zur Realisierbarkeit von Packordnungen beitragen, konnten keine abgeleitet werden.

### *Validierung:*

Die Ergebnisse zeigen, dass die im Zuordnungsalgorithmus implementierten Restriktionen Ergebnisse erzeugen, die den etablierten Packregeln der Praxis genügen. Die Ergebnisse für LKW1 und LKW2 zeigen, dass eine homogene Verteilung für die erste Hälfte des Sattelauflegers erreicht wird und der Schwerpunkt in Längsrichtung zwischen der geometrischen Mitte des Anhängers und den Achsen liegt. Des Weiteren liegt der Schwerpunkt in y-Richtung leicht auf der Fahrerseite, womit der Reifenverschleiß minimiert werden kann. Zusammengefasst können folgende Aussagen über den Algorithmus getroffen werden:

- Dynamische Implementierung der Gewichtsrestriktionen und somit Packmöglichkeit nach VDI 2700, individuellen Wünschen oder Fahrbahnverhältnissen möglich
- Minimierung von Materialermüdung und Reifenverschleiß durch die Packordnung realisiert
- Realisierbare Lösungen innerhalb von 2 Minuten gefunden

Als weiterer Aspekt soll an dieser Stelle festgehalten werden, dass durch das Bestreben des Unit Load Packers beide Kapazitäten des LKW voll auszulasten, die Diversifikation der zugeordneten UL pro LKW groß ist. Das führt zur Tatsache, dass das Finden einer gültigen Lösung im „Balanced Trailer Loading“ Algorithmus erleichtert wird.

Der Zeitaufwand betrug bei jedem Rechendurchlauf weniger als 2 Minuten, wobei aber schon früher funktionierende Lösungen gefunden wurden und diese im Sinne des „Vorschlagen und Vertauschen“ optimiert wurden.

Aus der Lösung für LKW3 geht hervor, dass der Algorithmus für eine kleine Anzahl an UL nicht gut geeignet ist. Da im PI dieser Anwendungsfall per Definition nicht eintritt beziehungsweise es ein Projektziel des MODULUSHCA ist, diesen Fall zu vermeiden, kann diese Tatsache von untergeordneter Bedeutung.

## 6 Zusammenfassung

Maßgeschneidert auf die Anforderungen des MODULUSHCA Szenarios, wurden in dieser Arbeit Werkzeuge erarbeitet, welche theoretische Konzepte unter realen Bedingungen abbilden und auf deren Realisierbarkeit prüfen. Anknüpfend an einer früheren Arbeit von Meller, Lin und Ellis in [MLE12] kann somit der Weg eines Produktes vom Packen in eine M-Box, über das Stapeln zu einem Verbund bis hin zur Verladung am LKW nachgebildet und simuliert werden. Dabei werden Anforderungen aus Normungen sowie gesetzlichen und technischen Bestimmungen erfüllt und somit eine Realisierung im hohen Maße wahrscheinlich gemacht. Im Detail können mit dem vorgestellten Methodenbaukasten folgende Arbeitsschritte im Distributionsprozess nachgebildet werden:

- Analyse einer Packordnung auf dessen Standsicherheit, bezogen auf eine frei wählbare Horizontalkraft oder der gesetzlich geforderten Mindestkraft
- Darstellungsmöglichkeit von jeder Packordnung in PTC CREO® 1.0 durch automatisierten Zusammenbau, gesteuert durch die VB-API Schnittstelle
- Aufteilung einer großen Anzahl ( ~ 25.000) UL auf LKW sodass die Anzahl der LKW die untere Schranke erreicht.
- Erstellung eines Packplans unter Rücksichtnahme der Minimierung von Reifenverschleiß und Materialermüdung gemäß
  - VDI 2700 und gesetzlichen Restriktionen oder
  - Individuellen Restriktionen

Durch Anwendung der gezeigten Werkzeuge konnten auch Funktionen betreffend der M-Box abgeleitet werden um die gesetzlich geforderte Standsicherheit zu gewährleisten.

Da diese Arbeit unter den Rahmenbedingungen des MODULUSHCA Projektes erstellt wurde, ergeben sich für die Anwendung in einem möglichen PI Szenario folgende Verbesserungsmöglichkeiten:

- Implementierung eines parametergesteuerten CAD Modells zur Darstellung der gesamten Geometrie der M-Box im Packmuster
- Entwicklung eines Packalgorithmus auf M-Box Ebene sodass UL nicht sortenrein gebildet werden müssen sondern verschiedene Packziele verfolgen können wie zum Beispiel
  - Schwerpunkt optimiert
  - Nach dem “First In Last Out” Prinzip
- Optimierung des Zeitbedarfes des “Unit Load Packers” oder aber
- Entwicklung eines Zuordnungsalgorithmus, der UL direkt am LKW unter Rücksichtnahme der realen Bedingungen anordnet, sodass die Anzahl der verwendeten LKW ein Minimum wird. Auf Grund der Tatsache, dass die Problemstellung NP-vollständig ist, ist dies „*wahrscheinlich*“ nicht effizient lösbar.

## 7 Verzeichnisse

### 7.1 Literaturverzeichnis

- [AIM04] **Dell’ Amico, Mauro; Iori, Manuel; Martello, Silvano:** *Heuristic Algorithms and Scatter Search for the Cardinality Constrained P//Cmax Problem*. In: Journal of Heuristics, Ausgabe 10, Seite 169-204, 2004.
- [BGR06] **Hauptverband der gewerblichen Berufsgenossenschaften (Hrsg.):** *BGR 234 – Lagereinrichtungen und -geräte*, BG Regel, Deutschland, 2006.
- [CT00] **Caprara, Alberto; Toth, Paolo:** *Lower bounds and algorithms for the 2-dimensional vector-packing problem*. In: Discrete Applied Mathematics, Ausgabe 111, Seite 231-262, 2001.
- [DIN04] **Deutsches Institut für Normung - Normungsausschuss für Verpackungswesen (Hrsg.):** *DIN EN 13698-1 – Produktspezifikation für Paletten*. Deutschland, 2004.
- [EHR13] **Ehrentraut, Florian:** *MODULUSHCA*. In: Logistikwerkstatt, 24.Mai 2013, Graz.
- [FM05] **Fleisch, Elgar; Mattern, Friedemann (Hrsg.):** *Das Internet der Dinge- Ubiquitous Computing und RFID in der Praxis, Visionen, Technologien, Anwendungen, Handlungsanleitungen*. Berlin/Heidelberg/New York, Springer Verlag, 2005.- ISBN 10 3-540-24003-9.
- [HER00] **Herrler, Rainer:** *Effizienzuntersuchungen in der “Vorschlagen und Vertauschen” – Methode für Zuordnungsprobleme*. Diplomarbeit, Universität Würzburg, 2000.
- [KP99] **Kellerer, Hans; Pferschy Ulrich:** *Cardinality constrained bin-packing problems*. In: Annals of Operations Research, Ausgabe 92, Seite 335-348, 1999
- [LJE14] **Landschützer, Christian; Jodin, Dirk; Ehrentraut, Florian:** *Modular Boxes for the Physical Internet – Technical Aspects*. In: 7<sup>th</sup> International Scientific Symposium on Logistics, Köln, 2014.
- [MIC14] **Microsoft Corporation:** *Microsoft EXCEL® 2013 online Hilfe*. <http://office.microsoft.com/en-us/support/> . U.S.A. 2014.
- [MLE12] **Meller, Russel; Lin, Yen-Hung; Ellis, Kimberly:** *The Impact of Standardized Physical Internet Containers on Shipping Volume*. In: Proceedings of ILS 2012: 4th International Conference on Information Systems, Logistics and Supply.
- [MMT12] **Meller, Russel; Montreuli, Benoit; Thivierge, Collin; Montreuli, Zachary:** *Functional Design of Physical Internet Facilities – A Road based Transit Center*. Progress in Material Handling Research 2012, U.S.A. 2012.



- [MMJ06] **Muraoka, Hideki; Miki Kubo; Joao P. Pedroso; Ohnishi, Makoto:** *An Algorithm for the cardinality constrained bin-packing problem.* In: Euro Special Interest Group On Cutting and Packing, Kongressbeitrag, Porto, 2006.
- [MOD13] **MODULUSHCA Deliverable 3.1 – Functional specification of iso modular logistics units in FMCG,** Brussels. 2013.
- [MOD14]: **MODULUSHCA Homepage:** <http://www.modulushca.com/>. Zugriff am 26.Mai 2014.
- [MT90] **Martello, Silvano; Toth, Paolo:** *Knapsack Problems - Algorithms and Computer Implementations,*New York, Jon Wiley & Sons, 1990. ISBN 0 471 – 924 20 2.
- [PIC13] **Pichler, Alexander:** *Knowledge-based Engineering zur Grobplanung und Gestaltung automatisierter Hochregallager* , Diplomarbeit, Technische Universität Graz, 2013.
- [PII14] **Physical Internet Initiative (Hrsg.):** *Physical Internet Manifesto,* <http://physicalinternetinitiative.org/> , Zugriff am 26.Mai 2014.
- [PTC10] **Parametric Technology Corporation (Hrsg.):** *Pro/Engineer® Wildfire® 5.0 VB API User's Guide.* U.S.A. 2010.
- [PTC14] **Parametric Technology Corporation (Hrsg.):** *PTC CREO® 1.0 User Guide,* U.S.A. 2010.
- [PUC14] **Pucher, Johann:** *Informationsgespräch mit Thomas Stöhr.* Geschäftsführender Gesellschafter der Johann Pucher GesmbH. Informationsgespräch am 02.11.2013, Oberdorf am Hohegg.
- [RZZ12] **Remic, Maja; Zerovnic, Gaspar; Zerovnik, Janez:** *An experimental comparison of some heuristics for cardinality constrained bin packing problem.* In: Business Systems Research, Ausgabe 3, Seite 57-63, 2012.
- [VDI00] **VDI Gesellschaft Fördertechnik Materialfluss Logistik (Hrsg.):** *VDI-Richtlinie 2700 Blatt 4– Ladungssicherung auf Straßenfahrzeugen-Lastverteilungsplan.* Berlin/Düsseldorf, 2000.
- [VDI04] **VDI Gesellschaft Fördertechnik Materialfluss Logistik (Hrsg.):** *VDI-Richtlinie 2700 – Ladungssicherung auf Straßenfahrzeugen.* Berlin/Düsseldorf, 2004.
- [WEG03] **Wegener, Ingo:** *Komplexitätstheorie – Grenzen der Anwendbarkeit von Algorithmen.* Berlin/Heidelberg, Springer Verlag, 2003.- ISBN 978-3-540-00161-4.
- [ZCW00] **Zhang, Guochuan; Chai, Xiaogiang; Wong, C.K.:** *Linear time approximation algorithms for bin packing.* In: Operations Research Letters, Ausgabe 26, Seite 217-222, 2000.

- [ZZ11] **Zerovnic, Gaspar; Zerovnik, Janez:** *Constructive heuristics for the canister filling problem.* In: Central European Journal of Operations Research, Ausgabe 19, Seite 371-389, 2011

## 7.2 Abbildungsverzeichnis

Abbildung 2-1: Synergien zwischen „Digital Internet“ - "Internet der Dinge" - "Physical Internet" [PII14] .....	4
Abbildung 2-2: Modulare Außenmaße der M-Boxen [EHR13] .....	6
Abbildung 2-3: Modularität der M-Boxen [MOD14] .....	6
Abbildung 2-4: Behältergrößen für den ersten Prototypenbau [LJE14] .....	7
Abbildung 2-5: Routing im PI-Szenario [PII14] .....	8
Abbildung 2-6: Prozessablauf .....	10
Abbildung 3-1: Darstellung der relevanten Rechengrößen zur Berechnung der Standsicherheit [BRG06] .....	13
Abbildung 3-2: Standsicherheitsberechnung einer Unit Load (vgl. [LJE14]) .....	15
Abbildung 3-3: Positionierungs-Punkt und Koordinatensystem .....	19
Abbildung 3-4: Flow Chart "Unit Load Analyse" .....	21
Abbildung 3-5: Maximales und minimales M-Box-Eigengewicht .....	24
Abbildung 3-6: Packordnung für M-Box Nr. 5 in PTC CREO® 1.0 .....	25
Abbildung 3-7: Packordnung für M-Box Nr. 1 in PTC CREO® 1.0 .....	25
Abbildung 4-1: Implementierung der Algorithmen in Unit Load Packer .....	35
Abbildung 5-1: Anordnung in 3er Reihe .....	46
Abbildung 5-2: Anordnung in 2er Reihe .....	46
Abbildung 5-3: Gemischte Anordnung von UL am LKW zur Vermeidung von Lücken .....	47
Abbildung 5-4: Lastverteilungsplan gemäß VDI 2700 [VDI00] .....	49
Abbildung 5-5: Notation zur Berechnung des Lastverteilungsplans [VDI00] .....	49
Abbildung 5-6: Kurvenverläufe im Lastverteilungsplan [VDI00] .....	50
Abbildung 5-7: Veranschaulichung Zuordnungsprobleme [HER00] .....	53
Abbildung 5-8: UL Standplatz-Notation .....	54
Abbildung 5-9: Implementierung in Programmumgebung .....	58
Abbildung 5-10: Darstellung der relativen Lage der UL am LKW (vgl. [LJE14]) .....	60
Abbildung 5-11: Gewichtsverteilung LKW 1 .....	61
Abbildung 5-12: Gewichtsverteilung entlang der x-Achse LKW 1 .....	61
Abbildung 5-13: Gewichtsverteilung entlang der y-Achse LKW 1 .....	61
Abbildung 5-14: Gewichtsverteilung LKW 2 .....	62
Abbildung 5-15: Gewichtsverteilung entlang der x-Achse LKW 2 .....	62
Abbildung 5-16: Gewichtsverteilung entlang der y-Achse LKW 2 .....	62
Abbildung 5-17: Gewichtsverteilung LKW 3 .....	63
Abbildung 5-18: Gewichtsverteilung entlang der x-Achse LKW 3 .....	63
Abbildung 5-19: Gewichtsverteilung entlang der y-Achse LKW 3 .....	63
Abbildung 8-1: Prozess-Legende .....	72

### 7.3 Tabellenverzeichnis

Tabelle 3-1: Rechengrößen zur Berechnung der Standsicherheit [BRG06] .....	13
Tabelle 3-2: Eignung der Methoden zur Erfüllung der Teilprozesse - Übersicht.....	17
Tabelle 3-3: Beispielhafte Darstellung der Packordnung .....	18
Tabelle 3-4: Dateien in „Unit Load Analyse“ zum automatisierten Zusammenbau.....	20
Tabelle 3-5: M-Box Größen.....	23
Tabelle 3-6: Eingangsdaten Unit Load Analyse .....	23
Tabelle 3-7: Abgeleitete Ergebnisse für das Design der M-Boxen .....	23
Tabelle 4-1: Zusammengefasste Ergebnisse aus [RZZ12] für 0% "schwere" Objekte .....	29
Tabelle 4-2: Übersicht Algorithmen, Anwendung und Implementierung .....	30
Tabelle 4-3: Übersicht Eingangsdaten PG_Output_5mbox_new .....	38
Tabelle 4-4: Übersicht Eingangsdaten PG_Output_6mbox_new .....	38
Tabelle 4-5: Übersicht Eingangsdaten PG_Output_9.8mbox.....	39
Tabelle 4-6: Übersicht Eingangsdaten PG_Output_12mbox.....	39
Tabelle 4-7: Übersicht Eingangsdaten PG_Output_Tophalf_(78)_mbox_10 .....	40
Tabelle 4-8: Übersicht Eingangsdaten PG_Output_440.155mbox.....	40
Tabelle 4-9: Detailergebnisse des Beispieldatensatzes .....	41
Tabelle 4-10: Übersicht der Ergebnisse für PG_Output_5mbox_new .....	41
Tabelle 4-11: Übersicht der Ergebnisse für PG_Output_6mbox_new .....	42
Tabelle 4-12: Übersicht der Ergebnisse für PG_Output_12mbox –.....	42
Tabelle 4-13: Übersicht der Ergebnisse für PG_Output_Tophalf_(78)_mbox_10.....	43
Tabelle 4-14: Übersicht der Ergebnisse für PG_Output_440.155mbox .....	43
Tabelle 5-1: Spezifikationen der betrachteten Fahrzeugkombination .....	46
Tabelle 5-2: Anordnungsvarianten von UL am LKW - Übersicht .....	47
Tabelle 5-3: Gesetzliche Bestimmungen laut KFG .....	48
Tabelle 5-4: Gesetzliche Rahmenbedingungen der betrachteten Fahrzeugkombination.....	48
Tabelle 5-5: Berücksichtigte Faktoren bei VDI 2700/KFG und Praxisanwendung - Übersicht .....	52
Tabelle 5-6: Ergebnisse des Beispieldatensatzes .....	60

## **8 Anhang**

### **8.1 Anhang 1 – veröffentlichter Kurzttext**

Publizierte Kurzfassung der Arbeit aus dem TUG-online – System in deutscher und englischer Sprache.

#### **8.1.1 A1 Deutsch**

Diese Arbeit beschreibt Pack- und Zuordnungsprozesse der Distributionslogistik im „Physical Internet“ unter Einbindung von praxisrelevanten Anforderungen. Anhand der Analyse einzelner Schritte im Distributionsprozess modularer Transportboxen werden relevante Prozesse identifiziert, wesentliche Einschränkungen, die aus technischen und gesetzlichen Restriktionen oder bindenden Normen resultieren, erhoben und mittels Algorithmen abgebildet. Folgende Teilaufgaben werden hierbei gelöst: 1.) Analyse eines Packmusters auf dessen Standsicherheit und dessen Darstellung mittels automatisierten Zusammenbaus in PTC CREO® 1.0; 2.) Entwicklung eines 2D-Vector-Packing Algorithmus zur Zuteilung von Ladungsträger an Transportmittel 3.) Entwicklung eines Anordnungs-Algorithmus zum Anordnen der Ladungsträger auf einem Sattelanhänger unter Einhaltung bindender Normen und Gesetze sowie Bedingungen aus der Praxis. Diese Algorithmen werden mit Datensätzen aus einem FMCG-Distributionszentrums ausführlich getestet und Stärken und Schwächen sowie Diskrepanzen zwischen Reglementierungen und der Umsetzung in der Praxis aufgezeigt.

#### **8.1.2 A2 Englisch**

Fields of investigation of this thesis are packing- and allocation processes that are identified in the distributions process within the “Physical Internet” in consideration of practical requirements. Based on the analysis of the distribution processes of modular boxes relevant processes are identified. Further essential constraints resulting from technical aspects and guidelines as well as restrictions of law are illustrated. Regarding these aspects, algorithms are developed. The following processes are respected: 1.) Analysis of a packing pattern focusing on steadiness and depiction of this packing pattern in PTC CREO® 1.0 through automatized assembly 2) Development of a 2D-Vector-Packing algorithm to distribute carrier to means of transportation 3) Development of an allocation algorithm to allocate carrier on a trailer in respect of technical guidelines and restrictions of law as well as practical requirements. These algorithms are tested with data sets of a FMCG distribution centre, their strength and weaknesses illustrated and discrepancies between restrictions and practical realization identified.

## 8.2 Anhang 2 – Zusatzinformationen

### 8.2.1 Legenden

Abbildung 8-1 zeigt die in dieser Arbeit verwendete Notation zur Darstellung von Prozessen.

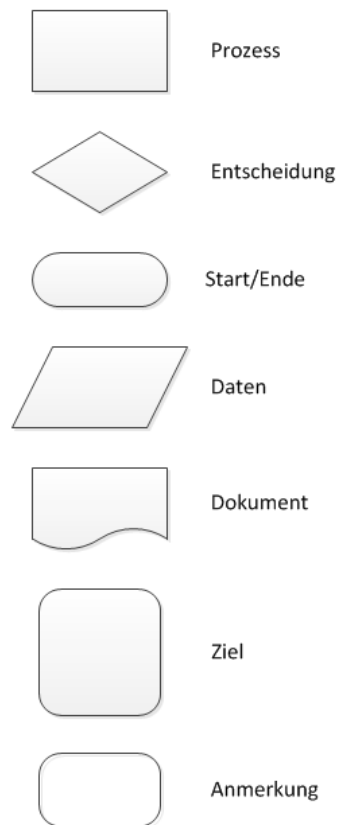


Abbildung 8-1: Prozess-Legende

### 8.2.2 Programmcodes

Da die Dokumentation in der VB-API minimal ist, werden im folgenden Abschnitt wesentliche Code-Elemente und Funktionen der automatisierten Baugruppenerstellung in PTC CREO® 1.0 dargestellt.

*Verbindung zu PTC CREO® herstellen:*

```
Public Function ConnectToProe() As Boolean
    statusLabel.Text = "Verbinde mit Pro/Engineer"

    options = (New CCpfcRetrieveModelOptions).Create
    options.AskUserAboutReps = False

    Try
        asyncConnection = (New pfcls.CCpfcAsyncConnection).Connect(DBNull.Value,
        DBNull.Value, DBNull.Value, DBNull.Value)
        session = asyncConnection.Session
        statusLabel.Text = "Verbindung erfolgreich initialisiert"
        Return True
    End Try
End Function
```

```

    Catch ex As Exception
        Return False
    End Try

End Function

```

*Modell regenerieren:*

```

private Function regenModel() As Boolean

    Try
        statusLabel.Text = "Regeneriere Modell"
        Dim partModel As IpfcModel
        Dim componentModel As IpfcSolid

        Dim regen As IpfcRegenInstructions
        regen = CType((New CCpfcRegenInstructions).Create(Nothing, Nothing,
Nothing), IpfcRegenInstructions)
        regen.AllowFixUI = True
        regen.ForceRegen = True
        regen.RefreshModelTree = True
        regen.ResumeExcludedComponents = True
        regen.UpdateInstances = True

        partModel = session.CurrentModel

        componentModel = CType(partModel, IpfcSolid)
        componentModel.Regenerate(regen)

        Dim window As IpfcWindow
        window = session.GetModelWindow(partModel)
        window.Refresh()
        window.Repaint()

        Return True
    Catch ex As Exception
        MsgBox(ex.Message + Chr(13) + ex.StackTrace.ToString & 1)
        Return False
    End Try

End Function

```

*Einen Parameter als Double erstellen von einem eingelesenen String*

```

Public Function CreateDoubleParametersFromString(ByVal s As String) As IpfcParamValue
    Try
        If IsNumeric(s) Then
            Return ((New CMpfcModelItem).CreateDoubleParamValue(CDb1(s)))
        Else
            Throw (New Exception("CreateDoubleParametersFromString: IsNumeric
false"))
        End If
    Catch ex As Exception
        Throw (New Exception("CreateDoubleParametersFromString: IsNumeric false"
& vbCrLf & ex.Message))
    End Try
End Function

```

*Einen Parameter als Integer erstellen von einem String*

```

Public Function CreateParametersFromString(ByVal s As String) As IpfcParamValue
    Try
        If (s.Equals("Y", StringComparison.OrdinalIgnoreCase)) Or
(s.Equals("true", StringComparison.OrdinalIgnoreCase)) Then

            Return (New CMpfcModelItem).CreateBoolParamValue(True)

        ElseIf (s.Equals("N", StringComparison.OrdinalIgnoreCase)) Or
(s.Equals("false", StringComparison.OrdinalIgnoreCase)) Then

            Return (New CMpfcModelItem).CreateBoolParamValue(False)

        ElseIf IsNumeric(s) Then
            Return ((New CMpfcModelItem).CreateIntParamValue(CType(s, Integer)))

        Else
            Return ((New CMpfcModelItem).CreateStringParamValue(s))

        End If

    Catch ex As Exception
        MsgBox(ex.Message.ToString + Chr(13) + ex.StackTrace.ToString)

        Return Nothing
    End Try

End Function

```

*Setzen der Parameter der einzelnen Boxen und Ableiten vom Elternteil*

```

Private Function CreateBoxes(ByVal Box_X As String, ByVal Box_Y As String, ByVal
Box_Z As String, ByVal CBox_X As String, ByVal CBox_Y As String, ByVal CBox_Z As
String)

    statusLabel.Text = "Parameter werden geändert"

    Dim Param_Box_X As IpfcParameter
    Dim Param_Box_Y As IpfcParameter
    Dim Param_Box_Z As IpfcParameter
    Dim Param_CBox_X As IpfcParameter
    Dim Param_CBox_Y As IpfcParameter
    Dim Param_CBox_Z As IpfcParameter

    Dim PBox_X As IpfcParamValue
    Dim PBox_Y As IpfcParamValue
    Dim PBox_Z As IpfcParamValue
    Dim PCBox_X As IpfcParamValue
    Dim PCBox_Y As IpfcParamValue
    Dim PCBox_Z As IpfcParamValue

    Dim pOwner As IpfcParameterOwner
    Dim model As IpfcModel

    Try

```

```

    Dim path As String = "L:\DA\Creo_Implementierung\"
    Dim father As String = "modbox_father.prt.8"
    Dim child As String = "C_" & Box_X & "_" & Box_Y & "_" & Box_Z & "_" &
CBox_X & "_" & CBox_Y & "_" & CBox_Z & ".prt"

    If System.IO.File.Exists(path & father) And Not
System.IO.File.Exists(path & child) Then
        System.IO.File.Copy(path & father, path & child)
    End If

    Dim options As IpfcRetrieveModelOptions
    options = (New CCpfcRetrieveModelOptions).Create()
    options.AskUserAboutReps = False

    modelDesc = (New CCpfcModelDescriptor).CreateFromFileName(child)
    model = session.RetrieveModelWithOpts(modelDesc, options)

    pOwner = CType(model, IpfcParameterOwner)

    Param_Box_X = pOwner.GetParam("Box_X")
    PBox_X = (New CMpfcModelItem).CreateIntParamValue(Box_X.ToString)
    Param_Box_X.SetScaledValue(PBox_X, Nothing)

    Param_Box_Y = pOwner.GetParam("Box_Y")
    PBox_Y = (New CMpfcModelItem).CreateIntParamValue(Box_Y.ToString)
    Param_Box_Y.SetScaledValue(PBox_Y, Nothing)

    Param_Box_Z = pOwner.GetParam("Box_Z")
    PBox_Z = (New CMpfcModelItem).CreateIntParamValue(Box_Z.ToString)
    Param_Box_Z.SetScaledValue(PBox_Z, Nothing)

    Param_CBox_X = pOwner.GetParam("CBox_X")
    PCBox_X = (New CMpfcModelItem).CreateIntParamValue(CBox_X.ToString)
    Param_CBox_X.SetScaledValue(PCBox_X, Nothing)

    Param_CBox_Y = pOwner.GetParam("CBox_Y")
    PCBox_Y = (New CMpfcModelItem).CreateIntParamValue(CBox_Y.ToString)
    Param_CBox_Y.SetScaledValue(PCBox_Y, Nothing)

    Param_CBox_Z = pOwner.GetParam("CBox_Z")
    PCBox_Z = (New CMpfcModelItem).CreateIntParamValue(CBox_Z.ToString)
    Param_CBox_Z.SetScaledValue(PCBox_Z, Nothing)

    model.Save()

    statusLabel.Text = "Boxen erzeugt"

    Return True
Catch ex As Exception

    MsgBox(ex.Message.ToString + Chr(13) + ex.StackTrace.ToString)
    Return False
End Try

statusLabel.Text = "alle Boxen parametrisiert"

End Function

```



*Zusammenbau der abgeleiteten Boxen*

```
Private Function assemble(ByVal componentFileName As String)
    Try
        model = session.CurrentModel

        If model Is Nothing Then
            Throw New Exception("Model not present")
        End If
        If (Not model.Type = EpfcModelType.EpfcMDL_ASSEMBLY) Then
            Throw New Exception("Model is not an assembly")
        End If

        assembly = CType(model, IpfcAssembly)
        modelDesc_1 = (New
CCpfcModelDescriptor).CreateFromFileName(componentFileName)
        componentmodel = session.GetModelFromDescr(modelDesc_1)

        If componentmodel Is Nothing Then
            componentmodel = session.RetrieveModel(modelDesc_1)
        End If

        asmcomp = assembly.AssembleComponent(componentmodel, Nothing)

        compDatum =
componentmodel.GetItemByName(EpfcModelItemType.EpfcITEM_COORD_SYS, "C_T")
        asmDatum = assembly.GetItemByName(EpfcModelItemType.EpfcITEM_COORD_SYS,
"C_B")

        Return True
    Catch ex As Exception
        MsgBox(ex.Message.ToString + Chr(13) + ex.StackTrace.ToString)
        Return False
    End Try
End Function
```