

Routing in dezentral gesteuerten, modularen Fördersystemen

Von der Fakultät für Maschinenbau
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von
M. Sc. Simon Sohr

2021

1. Referent: Prof. Dr.-Ing. Ludger Overmeyer
2. Referent: Prof. Dr.-Ing. habil. Peter Nyhuis
Vorsitz: Prof. Dr.-Ing. Gerhard Poll

Tag der Promotion: 15.12.2020

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Transport- und Automatisierungstechnik (ITA) der Gottfried Wilhelm Leibniz Universität Hannover im Rahmen des Forschungsvorhabens „Vernetzte, kognitive Produktionssysteme - netkoPs“. Das Forschungsvorhaben wurde von dem Bundesministerium für Bildung und Forschung (BMBF) gefördert.

Mein Dank gilt Herrn Prof. Dr.-Ing. Ludger Overmeyer für die mir übertragene Verantwortung zur Durchführung des Forschungsvorhabens und für die Möglichkeit zur Erstellung dieser Arbeit. Prof. Dr.-Ing. habil. Peter Nyhuis danke ich für die Übernahme des Korreferats und Prof. Dr.-Ing. Gerhard Poll danke ich für die Übernahme des Prüfungsvorsitzes.

Weiterhin bedanke ich mich bei den Projektpartnern des netkoPs Forschungsvorhabens. Besonders gerne erinnere ich mich an die Zusammenarbeit mit Herrn Dr.-Ing Heiko Stichweh, Herrn Frank Hartmann, Herrn André Heinke und Herrn Dr.-Ing. Nikita Shchekutin.

Besonders bedanke ich mich für die mir zu Teil gewordene Unterstützung von Herrn Dr.-Ing. Steffen Kleinert, Herrn Dr.-Ing. Tobias Krühn, Herrn Lars Bindszus, Herrn Dr.-Ing Tim Wolfer, Herrn Falk Deppe, Herrn Gerd-Albert Hoffmann, Herrn Daniel Schrein, Herrn M. Bechir Hachicha, Herrn Manuel Fernández und Herrn Heiko Wolff. Diese Dankesworte gelten ebenso den Studierenden, die mich entweder in der Form von Abschlussarbeiten oder als wissenschaftliche Hilfskraft unterstützt haben.

Weiterhin möchte ich mich bei meinem Freundeskreis bedanken. Hierbei sind meine Klettergruppe und die Improvisationstheatergruppe besonders hervorzuheben.

Mein ganz besonderer Dank geht an meine Familie, ohne die diese Arbeit nicht fertiggestellt worden wäre.

Hannover, im Dezember 2020

Simon Sohrt

Abstract

Durch die fortschreitende Digitalisierung der industriellen Produktionskette (Industrie 4.0) werden auch neue Anforderungen an industrielle Fördersysteme gestellt. So müssen diese autonom und flexibel auf Änderungen innerhalb der Produktion reagieren, zu denen z.B. Layoutänderungen zählen. Bereits existierende Steuerungsalgorithmen für modulare Fördersysteme können schon heute Konflikte verhindern, zu denen Kollisionen, Verklemmungen, Livelocks und das Verhungern zählen. Jedoch ist keiner der existierenden Algorithmen zeitfensterbasiert. Durch einen zeitfensterbasierten Steuerungsalgorithmus wird es möglich, unmittelbar die wichtigste Zielgröße eines Fördersystems zu optimieren: Den Durchsatz.

In dieser Arbeit wird ein zeitfensterbasierter Steuerungsalgorithmus für modulare Fördersysteme vorgestellt. Die Korrektheit des Algorithmus wird bewiesen, seine Laufzeit bestimmt und das Systemverhalten in einer Simulationsumgebung untersucht.

Schlagwörter: modulare Fördersysteme, Multi-Agenten Systeme, dezentrale Steuerung, konfliktfrei, bidirektionales Routing

Title: Routing for Decentralized Controlled Modular Conveyors

The ongoing digitalization of the industrial production chain (Industry 4.0) places new demands on industrial conveying systems. These systems must be able to react autonomously and flexibly to changes within production such as layout changes. Existing control algorithms for modular conveying systems can already prevent conflicts including collisions, deadlocks, livelocks, and starvation. However, none of the existing algorithms is time-window based. With a time-window based control algorithm, it is possible to optimize the most important performance indicator of a conveyor system: the throughput.

In this thesis, a time window-based control algorithm for modular conveyeing systems is presented. The correctness of the algorithm is proven, its runtime is determined and the system behavior is analyzed in a simulation environment.

Keywords: modular conveyor, multi-agent system, decentralized control, conflict-free, bi-directional routing

Inhaltsverzeichnis

1	Einleitung	1
2	Stand der Wissenschaft und Technik	3
2.1	Anforderungen an wandlungsfähige Materialflusssysteme.....	3
2.1.1	Sicherstellung des Betriebes	3
2.1.2	Erfüllung der Leistungskenngrößen	6
2.1.3	Ermöglichung der Wandlungsfähigkeit.....	7
2.2	Modulare Fördersysteme.....	10
2.2.1	Großskalige Fördersysteme.....	11
2.2.2	Kleinskalige Fördersysteme.....	21
2.3	Fahrerlose Transportfahrzeuge	25
2.3.1	Hardwarebauformen	26
2.3.2	Übersicht über Routingalgorithmen für FTF.....	26
2.3.3	Zeitfensterbasierte Routingalgorithmen für FTF	28
2.4	Vergleichende Übersicht der betrachteten Algorithmen	32
2.5	Fazit.....	36
3	Zielsetzung und Vorgehensweise	39
4	Beschreibung des Referenzfördersystems	43
4.1	Hardware	43
4.2	Steuerung.....	47
5	Modellbildung vom Referenzfördersystem	51
5.1	Gründe für die Modellbildung	51
5.2	Modell von den Transporteinheiten	52
5.3	Modell von den Fördermodulen.....	54
5.4	Integration der Modelle in eine Simulationsumgebung.....	57

6	Beschreibung des Routingalgorithmus	59
6.1	Algorithmus für großskalige Fördermodule	59
6.1.1	Übersicht	59
6.1.2	Erstellung einer neuen Reservierungsanfrage	63
6.1.3	Verarbeitung von empfangenen Nachrichten	66
6.1.4	Anwendungsbeispiel	68
6.1.5	Begründung der getroffenen Designentscheidungen	73
6.2	Modifikation des Algorithmus für kleinskalige Fördermodule	75
6.2.1	Beschreibung der Modifikation	75
6.2.2	Anwendungsbeispiel	82
6.3	Analyse des Algorithmus	84
6.3.1	Beweis der Konfliktfreiheit	84
6.3.2	Laufzeitanalyse	87
6.3.3	Verhalten im Fehlerfall	88
7	Simulationsgestützte Untersuchung des Systemverhaltens	91
7.1	Versuchsplanung	91
7.2	Systemverhalten von quadratischen Sortern	95
7.3	Systemverhalten von rechteckigen Sortern.....	102
7.4	Systemverhalten von quadratischen Kreuzungen.....	106
7.5	Fazit	108
8	Empfehlungen für den industriellen Einsatz	111
8.1	Weiterentwicklung der netkoPs-Fördermatrix	111
8.2	Anwendungsszenario.....	113
9	Zusammenfassung und Ausblick	115
	Abbildungsverzeichnis.....	119
	Tabellenverzeichnis.....	121
	Literaturverzeichnis.....	123
	Anhang	141

Abkürzungen und Formelzeichen

Abkürzungen

Abkürzung	Bedeutung
AG	Aktiengesellschaft
CAN	Controller Area Network
Co. KG	Compagnie Kommanditgesellschaft
CPU	Central Processing Unit
FTF	Fahrerloses Transportfahrzeug
et al.	et alii/aliae/alia (Deutsch: und andere)
GmbH	Gesellschaft mit beschränkter Haftung
GPIO	General Purpose Input/Output
GPS	Global Positioning System
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ITA	Institut für Transport- und Automatisierungstechnik
KI	Künstliche Intelligenz
Ltd.	Limited Company
MCS	Magic Carpet System
PTP	Precision Time Protocol

Abkürzung	Bedeutung
QR	Quick Response
RFID	Radio-Frequency Identification
RAM	Random-Access Memory
TE	Transporteinheit
VRP	Vehicle Routing Problem

Formelzeichen

Symbol	Einheit	Bedeutung
λ_i	%	Relativer Durchsatz des Moduls i
λ_{tot}	TE/h	Gesamtdurchsatz eines Fördersystems
μ	-	Reibungskoeffizient
σ_i	%	Belegungsgrad des Moduls i
F_{max}	N	Maximale übertragbare Kraft auf eine Transporteinheit
g	m/s ²	Fallbeschleunigung der Erde (9,81 m/s ²)
j	-	Iterationszähler für die Berechnung von LZ_D und LZ_S
k_{com}	s	Verhältnis aller kommunikationsbezogener Zeiten von [Sei] zu allen kommunikationsbezogenen Zeiten von [Soh]
k_{ges}	s	Proportionalitätsfaktor, um die Durchsätze von [Sei] und [Soh] miteinander zu vergleichen.
k_{pro}	%	Prozentuale Gewichtung zwischen den transportbezogenen Zeiten und den kommunikationsbezogenen Zeiten
k_{trans}	s	Verhältnis aller transportbezogener Zeiten von [Sei] zu allen transportbezogenen Zeiten von [Soh]
LZ_D	s	Lebensdauer einer Reservierungsanfrage basierend auf der Dauer vergangener Reservierungsanfragen
LZ_S	s	Lebensdauer einer Reservierungsanfrage basierend auf dem Layout des Fördersystems
m	kg	Masse einer Transporteinheit
m_o	-	Anzahl der empfangenen HINFÄLLIG- und BESTÄTIGUNGS-Nachrichten
m_{rd}	-	Anzahl der empfangenen ABLEHNUNGS- und ANFRAGE-Nachrichten
m_s	-	Anzahl der empfangenen ABLAUFPLAN-ANFRAGE-, ABLAUFPLAN- und ABLAUFPLAN-BESTÄTIGUNGS-Nachrichten

Symbol	Einheit	Bedeutung
N_{all}	-	Anzahl der Transporteinheiten, die im Intervall t_{int} eine Senke erreicht haben
N_i	-	Anzahl der Transporteinheiten, die im Intervall t_{int} über das Modul i transportiert worden sind
N_t	TE	Im Transport befindliche Transporteinheiten im Gesamtsystem
N_{tot}	TE	Wartende und im Transport befindliche Transporteinheiten im Gesamtsystem
N_{wait}	TE	Wartende Transporteinheiten im Gesamtsystem
n	-	Anzahl der Kreuzungen im Gesamtsystem
n_m	-	Anzahl der Module in einer Modulnachbarschaft
n_s	-	Sicherheitsfaktor zur Berechnung von LZ_D
n_{Ziel}	-	Anzahl der Module zwischen einer Quelle und einer Senke
t_{com}	s	Benötigte Zeit, um eine Nachricht zu einem Nachbarmodul zu senden
\bar{t}_{wait}	s	Durchschnittliche Wartezeit aller Transporteinheiten im System
t_i	s	Zeitintervall in dem das Modul i belegt ist
t_{int}	s	Länge eines Messintervalls
t_k	s	Durchlaufzeit der Transporteinheit k
\bar{t}_{tot}	s	Durchschnittliche Zeit zum Erreichen der Senke durch eine Transporteinheit im Gesamtsystem
\bar{t}_{Ziel}	s	Benötigte durchschnittliche Zeit einer Quelle, um eine Routenreservierung erfolgreich abzuschließen
v	-	Anzahl der FTF im Gesamtsystem

Kapitel 1

Einleitung

Die Ermöglichung einer kundenindividuellen Massenproduktion (Engl.: mass customization) ist ein wesentliches Ziel der Industrie 4.0. Durch diese Form der Massenproduktion steigt die Anzahl von Produktvarianten drastisch an. Gleichzeitig werden durch einen stetig wachsenden elektronischen Geschäftsverkehr (Engl.: e-commerce) Innovationszyklen und Produktlebensdauern verkürzt. Diese veränderte Marktsituation stellt neue Anforderungen an die Produktionskette: Sie muss in Zukunft wandlungsfähiger werden, um auf wechselnde Produktionsbedingungen reagieren zu können. [BMB13, For18]

Als Teil der Produktionskette gilt diese Forderung nach Wandlungsfähigkeit auch für die Materialflusssysteme. Die bisher bestehenden, konventionellen Materialflusssysteme werden diesen neuen Anforderungen jedoch nicht gerecht [Sch09]. Um die neuartigen Anforderungen an die Materialflusssysteme zu erfüllen, wurden modulare Fördersysteme mit dezentraler Steuerung entwickelt: Jedes einzelne dieser Fördermodule verfügt über eine eigene Steuerung und kann mit jedem anderen Fördermodul kommunizieren und kombiniert werden, wodurch die Wandlungsfähigkeit im Vergleich zu konventionellen Materialflusssystemen deutlich erhöht wird. Der Verkabelungsaufwand wird reduziert, da die einzelnen Module nicht mehr mit einer zentralen Steuerung verbunden werden müssen, sondern nur noch mit ihren direkt angrenzenden Modulen. Weiterhin wird die Skalierbarkeit erhöht, da durch das Hinzufügen eines weiteren Moduls auch gleichzeitig die verfügbare Rechenleistung erhöht wird.

Für modulare Fördermodule wurden bereits dezentrale Steuerungsalgorithmen entwickelt, die nach einer Neuordnung der Module das geänderte Layout erkennen. Alleine durch den Austausch von Nachrichten mit den benachbarten Modulen sind die Steuerungen in der Lage, selbstständig Routen für Transporteinheiten durch das Lay-

out zu reservieren. Die bereits entwickelten Steuerungsalgorithmen sind darüber hinaus fähig Kollisionen, Verklemmungen, Livelocks und das Verhungern von Quellen zu verhindern. Jedoch ist keiner der existierenden Algorithmen zeitfensterbasiert. Durch einen zeitfensterbasierten Steuerungsalgorithmus wird es möglich, unmittelbar die wichtigste Zielgröße eines Fördersystems zu optimieren: Den Durchsatz.

Im Kapitel 2 werden zunächst die „klassischen“ Anforderungen zusammen mit den neuartigen Anforderungen, die aus den Zielen der Industrie 4.0 abgeleitet sind, erfasst. Anschließend wird der Stand der Wissenschaft und Technik von den modularen Fördersystemen dargestellt. Insbesondere die Unterschiede der verwendeten Steuerungsalgorithmen werden herausgearbeitet. Im nächsten Schritt wird der Stand der Wissenschaft und Technik zu den fahrerlosen Transportfahrzeugen (FTF) untersucht. Dies hat zwei Gründe: Da die FTF auch über eine hohe Wandlungsfähigkeit verfügen, stehen sie in direkter Konkurrenz zu den modularen Fördersystemen. Weiterhin wird überprüft, ob die Steuerungsalgorithmen, die für FTF entwickelt worden sind, auf die modularen Fördersysteme übertragbar sind.

Aufbauend aus den Erkenntnissen aus dem Stand der Wissenschaft und Technik wird in dem Kapitel 3 das Forschungsziel dargestellt. Dieses besteht darin, einen neuartigen zeitfensterbasierten Steuerungsalgorithmus für die modularen Fördersysteme zu entwickeln. Der Algorithmus soll hierbei unter anderem auch für das am ITA entwickelte netkoPs-Fördersystem nutzbar sein, welches in dem Kapitel 4 vorgestellt wird. Von diesem Fördersystem wird ein Softwaremodell erstellt (siehe Kapitel 5), mit dem die Eigenschaften des in dieser Arbeit entwickelten zeitfensterbasierten Steuerungsalgorithmus untersucht werden. Die Beschreibung des Algorithmus erfolgt in dem Kapitel 6. Anschließend wird mit Hilfe des entwickelten Softwaremodells das Systemverhalten von Fördersystemen, die durch den neuartigen Algorithmus angesteuert werden, untersucht (Kapitel 7). Abschließend werden in Kapitel 8 Weiterentwicklungsempfehlungen gegeben und in Kapitel 9 die wichtigsten Inhalte dieser Arbeit zusammengefasst.

Kapitel 2

Stand der Wissenschaft und Technik

2.1 Anforderungen an wandlungsfähige Materialflusssysteme

In diesem Kapitel werden drei Arten von Anforderungen vorgestellt. Die erste und wichtigste Art von Anforderung ist, dass ein Stillstand des Materialflusssystems verhindert wird. Die zweite Art von Anforderungen besteht darin, dass das Materialflusssystem die vom Anwendungsfall geforderten Leistungskenngrößen erfüllt. Nur bei einer Erfüllung dieser Leistungskenngrößen ist ein ökonomischer Betrieb des Systems gewährleistet. Die dritte Art der Anforderungen betrifft die Wandlungsfähigkeit des Systems. Die ersten beiden Arten von Anforderungen müssen von jedem Materialflusssystem erfüllt werden. Die dritte Art nur von wandlungsfähigen Materialflusssystemen.

2.1.1 Sicherstellung des Betriebes

Zur Sicherstellung des Betriebes müssen Kollisionen, Verklemmungen, Livelocks und das Verhungern von Transporteinheiten verhindert werden. Zunächst wird die Verhinderung von Kollisionen und Verklemmungen untersucht und anschließend die Verhinderung von Livelocks und dem Verhungern.

Eine Kollision entsteht, wenn zwei Transporteinheiten zum gleichen Zeitpunkt den gleichen Ort im Fördersystem beanspruchen. In [Tan15] definieren TANENBAUM und BOS Verklemmungen folgendermaßen: „Eine Menge von Prozessen befindet sich in einem Deadlock, wenn jeder dieser Prozesse auf ein Ereignis wartet, das nur ein anderer Prozess aus dieser Menge verursachen kann.“ Von MAYER und SEIBOLD wurden in [May09] und [Sei16b] diese Definition auf Fördersysteme übertragen: Transporteinheiten wurden als Prozesse definiert und die einzelnen Förderer des Systems werden als Ressourcen betrachtet. Während des Transportes einer Transporteinheit (= Prozess)

werden nun nacheinander Ressourcen (= Förderer) belegt und wieder freigegeben. Beide Autoren kommen zu dem Schluss, dass es nur zwei mögliche Arten von Verklemmungen in einem Fördersystem gibt: Verklemmungen entstehen entweder auf Grund von entgegengesetzten Routen oder auf Grund von Transporteinheiten, die in einer Kreisstruktur zyklisch aufeinander warten (siehe Abbildung 2.1).

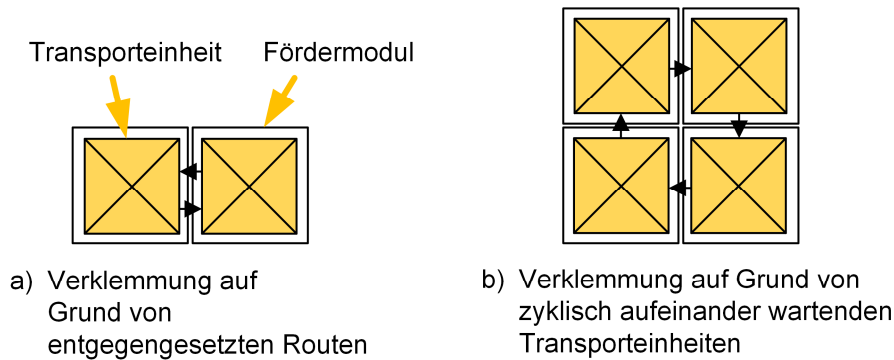


Abbildung 2.1: Zwei mögliche Arten von Verklemmungen in Fördersystemen, nach [Sei16b]

Kollisionen und Verklemmungen in Materialflusssystemen sind miteinander verwandte Effekte. Wenn eine Verklemmungssituation nicht erkannt wird und stattdessen die Transporteinheiten ihren Weg fortsetzen, kommt es zu einer Kollision. Jedoch geht nicht jeder Kollision eine Verklemmung voraus. Die Verhinderung von Kollisionen ist, im Vergleich zu der Verhinderung von Verklemmungen, einfach: Das Fördersystem muss lediglich sicherstellen, dass Transporteinheiten niemals zum gleichen Zeitpunkt den gleichen Ort beanspruchen.

COFFMAN ET AL. haben in [Cof71] alle notwendigen Bedingungen für das Auftreten von Verklemmungen ermittelt. Aufbauend auf dieser Erkenntnis haben TANENBAUM und BOS in [Tan15] vier grundsätzliche Strategien erarbeitet, um mit Verklemmungen umzugehen. Diese sind:

1. **Ignorieren des Problems:** Wenn man das Problem ignoriert, wird man vielleicht auch vom Problem ignoriert.
2. **Detektion und Korrektur:** Verklemmungen dürfen auftreten, diese werden anschließend detektiert und korrigiert.
3. **Dynamisches Vermeiden:** Verklemmungen werden vermieden durch die sorgfältige Reservierung von Ressourcen.
4. **Vorbeugung:** Das System wird so aufgebaut, dass Verklemmungen nicht auftreten können.

Die Auswirkung der Anwendung jeder dieser vier Strategien auf die Verhinderung von Verklemmungen in Fördersystemen wird im Folgenden dargestellt.

Die „Ignorieren des Problems“ Strategie ist für automatisierte Fördersysteme nicht geeignet, da das Fördersystem teilweise oder vollständig funktionsunfähig wird, bis die Verklemmung durch einen manuellen Eingriff gelöst wird.

Die „Detektion und Korrektur“ Strategie kann auch zu einer teilweisen oder vollständigen Funktionsunfähigkeit des Fördersystems führen, die wiederum nur durch einen manuellen Eingriff aufgelöst werden kann. Dies wird an den beiden Beispielen aus der Abbildung 2.1 verdeutlicht: Sowohl im Fall a) als auch im Fall b) können die entstandenen Verklemmungen gelöst werden, indem eine Transporteinheit zurücksetzt. Bei a) muss die Transporteinheit zurücksetzen, bis der Engpass frei ist und bei b) muss eine Transporteinheit kurzzeitig die Kreisstruktur verlassen. KRÜHN hat jedoch in [Krü15] gezeigt, dass durch eventuell nachrückende Transporteinheiten ein Zurücksetzen unmöglich ist und somit wiederum ein manueller Eingriff notwendig ist.

Die Strategie „Dynamisches Vermeiden“ bedeutet in Fördersystemen, dass ein Routingalgorithmus verwendet wird, der das Auftreten von Verklemmungen verhindert. Die Funktionsweise von verschiedenen Routingalgorithmen wird in den nachfolgenden Kapiteln dargestellt.

Die Strategie „Vorbeugen“ wird häufig in konventionellen Fördersystemen verwendet. Der in Abbildung 2.1 dargestellte Fall a) kann z.B. nicht auftreten, wenn alle Förderer des Fördersystems nur unidirektionales Fördern erlauben (also nur in eine Richtung fördern). Der in Abbildung 2.1 dargestellte Fall b) kann durch die mechanische Konstruktion des Förderers vermieden werden. Beispielsweise kann es bei den Ring- und Drehsortern zu keiner Verklemmung kommen, selbst wenn die gesamte Kapazität des Sorters genutzt wird, da die einzelnen Transporteinheiten auf kreisförmigen Routenabschnitten bewegt werden.

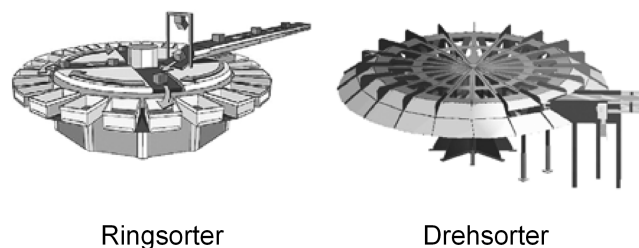


Abbildung 2.2: Ring- und Drehsorter [Jod12]

Falls ein Fördersystem bidirektionales Fördern bereitstellt, kann die Strategie „Vorbeugen“ nicht angewendet werden. Es eignet sich nur die Strategie des „Dynamisches Vermeidens“ zur Verhinderung von Verklemmungen.

Neben der Verhinderung von Kollisionen und Verklemmungen müssen Livelocks und das Verhungern (Engl.: Starvation) von Transporteinheiten verhindert werden. Ein Livelock ist eingetreten, wenn eine Transporteinheit immer wieder zyklisch den gleichen Weg abfährt, ohne ihrem Ziel näherzukommen [Sch15]. Livelocks können durch Routingalgorithmen verhindert werden, indem bei der Routenreservierung sichergestellt wird, dass jeder Förderer nur einmal befahren wird.

Das Verhungern tritt auf, wenn Transporteinheiten über zwei Richtungen auf eine Kreuzung einfahren können. Wenn die erste Richtung eine höhere Priorität besitzt als die zweite Richtung und dauerhaft Transporteinheiten von der ersten Richtung auf die Kreuzung fahren, werden Transporteinheiten aus der zweiten Richtung niemals die Kreuzung befahren. Transporteinheiten aus der zweiten Richtung „verhungern“ [Sei16b]. Das Verhungern kann verhindert werden, indem bei der Routenreservierung Fairnessregeln an Kreuzungen beachtet werden. Eine leicht zu implementierende Fairnessregel besteht darin, dass ältere Transporteinheiten gegenüber neuen Transporteinheiten bevorzugt werden.

2.1.2 Erfüllung der Leistungskenngrößen

Neben der Verhinderung von Kollisionen, Verklemmungen, Livelocks und dem Verhungern müssen alle Materialflusssysteme die vom Anwendungsfall geforderten Leistungskenngrößen erfüllen. Die Leistungskenngrößen lassen sich in drei Gruppen unterteilen: systembezogene Kenngrößen, fördererbezogene Kenngrößen und transporteinheitenbezogene Kenngrößen.

Die Systemkenngrößen werden ermittelt, wenn das Fördersystem einen stationären Zustand (Engl. steady state) erreicht hat. Dies bedeutet, dass sich die Zustandsgrößen des Systems während einer Zeitspanne nur insignifikant ändern. Ein Fördersystem erreicht einen stationären Zustand, wenn genauso viele Transporteinheiten eingeschleust, wie ausgeschleust werden. Dies ist zum Beispiel der Fall, wenn das Fördersystem seine Kapazität erreicht hat. Die wichtigste Systemkenngröße, sobald ein stationärer Zustand erreicht worden ist, ist der Durchsatz λ_{tot} . Dieser ist definiert als die Anzahl der Transporteinheiten, die innerhalb eines Zeitintervalls ihre Senke erreicht haben. Zwei weitere wichtige Kenngrößen sind die Anzahl der Transporteinheiten im System N_{tot} und die durchschnittliche Zeit \bar{t}_{tot} , die die Transporteinheiten benötigen, um die Senken zu erreichen. Die Anzahl der Transporteinheiten setzt sich hierbei zusammen aus der Anzahl der wartenden Transporteinheiten N_{wait} und der Anzahl der

Transporteinheiten, bei denen der Transportprozess schon begonnen hat N_t . Der Zusammenhang zwischen λ_{tot} , N_{tot} und \bar{t}_{tot} wird durch den Satz von LITTLE [Lit61] ausgedrückt:

$$\lambda_{tot} = \frac{N_{tot}}{\bar{t}_{tot}} \quad (2.1)$$

Eine weitere Systemkenngroße ist die durchschnittliche Wartezeit aller Transporteinheiten \bar{t}_{wait} , bevor der Transportprozess beginnt. Falls die durchschnittliche Wartezeit ~ 0 ist, bedeutet dies, dass die Kapazität des Fördersystems noch nicht erschöpft ist. Wenn die zu erwartende durchschnittliche Wartezeit bestimmt ist, kann die benötigte Kapazität der Eingangspuffer berechnet werden. Falls vor dem Beginn des Förderprozesses eine Routenreservierung erfolgen muss, muss dieser Routenreservierungsprozess kürzer sein als die durchschnittliche Wartezeit, um unnötige Wartezeiten auf den Transporteinheitenquellen zu vermeiden.

Zu den Kennzahlen für die Förderer zählt die relative Auslastung σ_i und der relative Durchsatz λ_i . Die relative Auslastung ist folgendermaßen definiert:

$$\sigma_i = \frac{t_i}{t_{int}} \quad (2.2)$$

Hierbei ist t_{int} die Länge des betrachteten Zeitintervalls und t_i die Zeit, die der Förderer i mit einer Transporteinheit belegt ist. Der relative Durchsatz λ_i ist das Verhältnis zwischen dem Durchsatz des einzelnen Förderers i zum Durchsatz des gesamten Systems (siehe Formel 2.3). Der relative Durchsatz kann z.B. dazu genutzt werden, den Verschleiß eines Förderers vorherzusagen.

$$\lambda_i = \frac{N_i}{N_{all}} \quad (2.3)$$

Hierbei ist N_m die Anzahl der Transporteinheiten, die im Intervall t_{int} über den Förderer i transportiert worden sind, und N_{all} ist die Anzahl aller Transporteinheiten, die im Intervall t_{int} eine Senke erreicht haben.

Zu den transporteinheitenbezogenen Kenngrößen zählt die Durchlaufzeit t_k . Die Durchlaufzeit für eine Transporteinheit k setzt sich hierbei zusammen aus der Zeit, die für die Reservierung einer Route benötigt wird und der Zeit, die für den anschließenden Transport benötigt wird.

2.1.3 Ermöglichung der Wandlungsfähigkeit

Bevor die Anforderungen an ein wandlungsfähiges Fördersystem dargestellt werden, muss zunächst der Begriff der Wandlungsfähigkeit definiert werden. Wandlungsfähigkeit ist in der Literatur nicht einheitlich definiert und wird zum Teil als Synonym für

Flexibilität, Agilität und Rekonfigurierbarkeit genutzt oder scharf von diesen Begriffen abgegrenzt. Das Problem der uneinheitlichen Definition wird weiter verstärkt durch unterschiedliche Übersetzungen der Begriffe aus der englischsprachigen Literatur. [Mö108]

Im Rahmen dieser Arbeit wird die von NYHUIS ET AL. in [Nyh08] getroffene Definition der Flexibilität und Wandlungsfähigkeit verwendet: „Flexibilität beschreibt hierbei die Fähigkeit eines Produktionssystems, sich schnell und nur mit sehr geringem finanziellen Aufwand an geänderte Einflussfaktoren anzupassen. Die Veränderungen und somit die möglichen erreichbaren Systemzustände sind im Kontext der Flexibilität durch vorgehaltene Maßnahmenbündel definiert und durch zum Zeitpunkt der Planung festgelegte Korridore begrenzt. Die Wandlungsfähigkeit hingegen wird als Potential verstanden, auch jenseits der vorgehaltenen Korridore organisatorische und technische Veränderungen bei Bedarf reaktionsfähig durchführen zu können.“

Die Bedeutung der Wandlungsfähigkeit von Produktionssystemen wird in Zukunft ansteigen, da die zugrundeliegenden Effekte (Globalisierung, kundenindividuelle Massenproduktion und kürzere Produktlebenszyklen) weiter zunehmen werden. [Fur10, Kle13]

Damit ein Produktionssystem, zu denen auch die Fördersysteme zählen, als wandlungsfähig gilt, muss es verschiedene Anforderungen erfüllen. Auch an dieser Stelle gibt es in der Literatur keine einheitlichen Definitionen dieser Anforderungen. Im deutschsprachigen Raum werden entweder die von NYHUIS ET AL. in [Nyh08] aufgestellten Definitionen verwendet oder die im Rahmen des SFB-467 erarbeiteten Definitionen [Wes09]. Im englischsprachigen Raum werden häufig die Definitionen von SETHI ET AL. verwendet [Set90]. Diese haben eine Übersicht über verschiedenen Flexibilitätarten von Produktionssystemen erarbeitet, indem sie die Ergebnisse von 97 Veröffentlichungen zusammengefasst haben. Besonders hervorzuheben sind weiterhin die Definitionen von FURMANS ET AL. [Fur10], da diese Anforderungen definiert haben, die sich konkret auf Fördersysteme beziehen. Im Rahmen dieser Arbeit wird auf Grund der klaren Abgrenzung zwischen den einzelnen Definitionen die Einteilung von NYHUIS ET AL. verwendet in Kombination mit den für Fördersysteme konkretisierten Anforderungen von FURMANS ET AL.

Insgesamt gibt es somit nach NYHUIS ET AL. fünf Anforderungen, die an ein wandlungsfähiges Fördersystem gestellt werden: Universalität, Mobilität, Skalierbarkeit, Modularität und Kompatibilität. Diese werde im Folgenden näher dargestellt. Falls möglich, werden hierbei quantitative Größen, mit denen die Erfüllung dieser Anforderungen gemessen werden können, benannt.

Universalität bedeutet, dass ein Fördersystem für unterschiedliche Aufgaben eingesetzt werden kann. Im Rahmen dieser Arbeit wird die Universalität daran gemessen, wie viele logistische Funktionen ein Fördersystem bereitstellen kann und wie viele verschiedene Größen von Transporteinheiten bewegt werden können.

In [Ven12] haben VENTZ ET AL. eine Übersicht über logistische Funktionen von Fördersystemen erarbeitet, die sich dazu eignet die Universalität quantifizierbar zu machen. Die Funktionen sind: Transportieren, Rotieren, Ausrichten, Zusammenführen, Verteilen, Stauen, Vereinzeln, Zwischenspeichern und Sequenzieren. Je mehr logistische Funktionen unterstützt werden, desto höher ist die Universalität des Fördersystems. Neben der Verfügbarkeit von logistischen Funktionen wird die Universalität auch daran gemessen, ob das Fördersystem in der Lage ist die logistischen Funktionen nur unidirektional oder auch bidirektional auszuführen.

Weiterhin wird die Universalität daran gemessen, wie viele verschiedene Arten von Transporteinheiten durch das Fördersystem transportiert werden können. Da es eine große Anzahl an verschiedenen Transporteinheiten gibt, werden im Rahmen dieser Dissertation die Kleinladungsträger, die in der VDA 4500 beschrieben sind, als Referenz-Transporteinheiten ausgewählt. Die VDA 4500 wurde vom Verband der Automobilindustrie erarbeitet, um die Logistikkette in der Automobilhersteller- und Zuliefererindustrie zu standardisieren. Auf Grund der Bedeutung der deutschen Automobilindustrie haben sich die genormten Kleinladungsträger auch in vielen anderen europäischen Industriebereichen durchgesetzt. Insgesamt werden in der Norm fünf verschiedene Größen definiert (siehe Tabelle 2.1). Je mehr Größen von VDA 4500 Kleinladungsträgern transportiert werden können, desto höher ist die Universalität.

Tabelle 2.1: Kleinladungsträger nach VDA 4500 [VDA17]

Bezeichnung	Nennmaße in mm
R-KLT 6429 / RL-KLT 6280	600×400×280
R-KLT 6415 / RL-KLT 6147	600×400×147
R-KLT 4329 / RL-KLT 4280	400×300×280
R-KLT 4315 / RL-KLT 4147	400×300×147
R-KLT 3215 / RL-KLT 3147	300×200×147

Die nächste Anforderung zur Erfüllung der Wandlungsfähigkeit ist die Mobilität. Im Rahmen dieser Arbeit werden zwei Stufen von Mobilität unterschieden: Fördersysteme der ersten Stufe verfügen über Rollen und können durch externe Krafteinwirkung bewegt werden. Fördersysteme der zweiten Stufe verfügen über angetriebene Räder und

können sich selbstständig innerhalb der Produktionsstätte bewegen. Wenn ein Fördersystem über eine hohe Mobilität verfügt, können Layoutänderungen schnell durchgeführt werden, um das Fördersystem an die geänderten Produktionsbedingungen anzupassen.

Die nächsten drei Anforderungen zur Erfüllung der Wandlungsfähigkeit sind miteinander verknüpft. Diese sind die Skalierbarkeit, die Modularität und die Kompatibilität. Skalierbarkeit bedeutet, dass die Kapazität eines Fördersystems durch das Hinzufügen oder Entfernen von Förderern angepasst werden kann. Damit ein Fördersystem skalierbar ist, muss sowohl die Hardware als auch die Software skalieren. Dies kann z.B. durch dezentrale Steuerung erreicht werden: Jeder Förderer verfügt über eine eigene dezentrale Steuerung und somit wird mit zunehmender Förderkapazität auch immer die verfügbare Rechenkapazität gesteigert. Wenn diese Förderer modular aufgebaut sind, ist die Skalierbarkeit leichter zu realisieren. Weiterhin müssen die Module sowohl mit den Förderern des Fördersystems als auch mit angrenzenden Systemen kompatibel sein. Dies wird durch einheitliche Schnittstellen erreicht. Die Einheitlichkeit der Schnittstellen gilt hierbei sowohl für die Transportgüter, die elektrische Energieversorgung als auch die Informationen.

Im Folgenden werden zwei Arten von wandlungsfähigen Fördersystemen vorgestellt. Dies sind die modularen Fördersysteme und die fahrerlosen Transportsysteme.

2.2 Modulare Fördersysteme

Im Rahmen dieser Arbeit werden die modularen Fördersysteme in zwei Klassen unterteilt: Großskalige und kleinskalige Fördersysteme. Ein Fördersystem wurde von VENTZ in [Ven16] als kleinskalig definiert, wenn entlang jeder Kante einer Transporteinheit mindestens drei Förderer benötigt werden, um die Transporteinheit zu bewegen. Werden weniger als drei Förderer benötigt, wird das Fördersystem als großskalig bezeichnet. Im Rahmen dieser Arbeit wird der Begriff der Kleinskaligkeit noch enger gefasst: Ein Förderer ist kleinskalig, wenn selbst die kleinste zu bewegende Transporteinheit größer ist als ein Förderer. Der Förderer ist somit kleinskalig in Bezug auf die Transporteinheit. Im Gegensatz hierzu sind Förderer großskalig, wenn selbst die größte Transporteinheit kleiner ist als ein Förderer. Die Erfüllung der Anforderungen durch die Algorithmen erfolgt anschließend in dem Kapitel 2.4 durch eine vergleichende Übersicht.

2.2.1 Großskalige Fördersysteme

Zu den großskaligen Fördersystemen zählt das FlexConveyor-System, welches die Grundlage für viele weitere Systeme darstellt, die im Folgenden noch dargestellt werden. Das FlexConveyor-System ist aus einzelnen quadratische Modulen (siehe Abbildung 2.3) aufgebaut. Jedes Modul besteht aus einem Rollenförderer mit integriertem Hubriemenausschleuser. Weiterhin sind in jedem Modul vier Lichtschranken und eine eigene dezentrale Steuereinheit integriert. Module, die als Quellen oder Senken eingesetzt werden sollen, verfügen zusätzlich noch über ein RFID-Lesegerät. Dieses wird genutzt, um RFID-Label an den Transporteinheiten auszulesen und somit das Ziel der Transporteinheiten zu bestimmen. Die einzelnen Steuerungen der Module können nur mit den Steuerungen der direkt angrenzenden benachbarten Module kommunizieren. Sowohl hardware- als auch softwareseitig ist das System so aufgebaut, dass nur Transporteinheiten bewegt werden können, die kleiner sind als die einzelnen Module. Nur bei der Übergabe einer Transporteinheit von einem Modul an das benachbarte Modul stehen die Transporteinheiten kurzzeitig mit zwei Modulen in Kontakt. [May09]

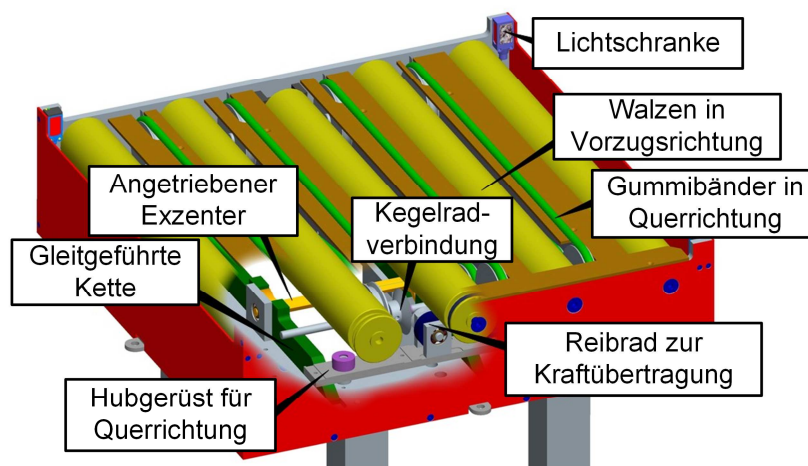


Abbildung 2.3: FlexConveyor-Modul, nach [May09]

Für die FlexConveyor-Module wurden verschiedene Steuerungsalgorithmen entwickelt, die im Folgenden dargestellt werden. Zunächst wird der Algorithmus von MAYER vorgestellt, der in [May09] beschrieben wird. Dieser kann in drei Phasen unterteilt werden: Initialisierungsphase, Transportplanungsphase und Transportphase. In der Initialisierungsphase werden die einzelnen Module zu einem neuen Layout zusammengefügt. Anschließend ermittelt jedes Modul die Anzahl und die IDs seiner benachbarten Module. Daraufhin werden Routingtabellen erstellt. Die Erstellung eines neuen Eintra-

ges in der Routingtabelle beginnt ausgehend von den Senken. Ein Eintrag in der Routingtabelle besteht aus den folgenden drei Informationen: ID der Senke, Distanz zur Senke und ID des Nachbarmoduls mit der geringsten Distanz zur Senke.

Ein Beispiel: Die Senke A erstellt initial einen neuen Eintrag in ihrer Routingtabelle: Als ID der Senke und für die ID des Nachbarmoduls mit der geringsten Distanz zu Senke wird die eigene ID abgespeichert. Weiterhin wird als Distanz 0 gespeichert. Dieser neu erstellte Routingeintrag wird an alle Nachbarmodule gesendet. Die benachbarten Module erstellen ebenfalls einen neuen Eintrag in ihrer Routingtabelle für die Senke A. Als Distanz wird eine 1 gespeichert, da die Nachbarmodule ein Modul weit entfernt sind von der Senke. Als ID des Nachbarmoduls mit der geringsten Distanz zu Senke wird die ID der Senke abgespeichert. Nachdem die Nachbarmodule von der Senke ihre Einträge erstellt haben, senden sie wiederum die neu erstellten Einträge an ihre Nachbarn. Auf diese Weise breitet sich der Eintrag für eine Senke im System aus. Falls ein Modul von zwei verschiedenen Nachbarmodulen einen Routingeintrag für die gleiche Senke erhält, wird der Routingeintrag mit der geringen Distanz übernommen. Die ID des Nachbarmoduls mit der geringsten Distanz zur Senke wird ebenfalls angepasst. Nachdem die Routingtabelle konvergiert ist, ist die Initialisierungsphase abgeschlossen und die Transportplanungsphase kann beginnen. Die Routingtabellen werden als statisch angesehen, da die Einträge innerhalb der Tabellen während des Betriebes nicht mehr angepasst werden.

Sobald eine Transporteinheit auf einer Quelle eingeschleust wird, beginnt die Transportplanungsphase. Die Quellen lesen zunächst das RFID-Label der zu transportierenden Transporteinheit aus und bestimmen so das Ziel. Die Routenplanung erfolgt zweistufig: Im ersten Schritt wird die Route von der Quelle zu der Senke lediglich geplant. Eine Route setzt sich aus mehreren Routenabschnitten zusammen: Jedes Modul entlang der Route von der Quelle bis zur Senke reserviert einen eigenen Routenabschnitt. Die Reservierung erfolgt auf Grund der Informationen in der Routingtabelle. Auf einem Modul dürfen sich mehrere Routensegmente gleichzeitig befinden, die sich im Zustand "in Planung" befinden.

Im zweiten Schritt wird die nun geplante Route ausgehend von der Senke zu der Quelle reserviert. Dieser zweistufige Prozess ist notwendig, um Kollisionen und Verklemmungen auf geraden Streckenabschnitten (einschließlich Engpässe) zu verhindern. Auf einem Modul dürfen niemals zwei reservierte und einander entgegengerichtete Routensegmente vorhanden sein. Die Verhinderung von Verklemmungen in Kreisverkehren erfolgt durch MAYERS Algorithmus erst während der Transportphase.

Kurz vor der Aufnahme einer Transporteinheit durch ein Modul prüft das Modul, ob es die Transporteinheit aufnehmen darf, ohne dass es zu einer Verklemmung kommt. Hierzu sendet es einen Deadlock-Token in die Richtung der geplanten Route der Transporteinheit. Der empfangene Token wird von jedem Modul in die Richtung des Routenabschnittes von der Transporteinheit weitergesendet, der gerade das Modul belegt. Dies ist beispielhaft in der Abbildung 2.4 dargestellt: Da das Modul 4 von der Transporteinheit I angefragt wird, sendet es einen Token in die Richtung des reservierten Routenabschnittes von I. Somit wird der Token zu dem Modul 5 gesendet. Da Modul 5 bereits von Transporteinheit II belegt ist, wird ein Token in die Richtung des reservierten Routenabschnittes von Transporteinheit II gesendet. In diesem Beispiel wird solange der Token weitergesendet, bis dieser wieder Modul 4 erreicht. Modul 4 erkennt nun, dass eine Verklemmungsgefahr besteht und erteilt der Transporteinheit I keine Freigabe. Falls der Deadlock-Token im Kreis gesendet wird und wieder das Ursprungsmodul erreicht, darf keine Transporteinheit aufgenommen werden, da es ansonsten zu einer Verklemmung kommt. Das Modul wartet eine kurze Zeit, bevor ein neuer Deadlock-Token generiert wird.

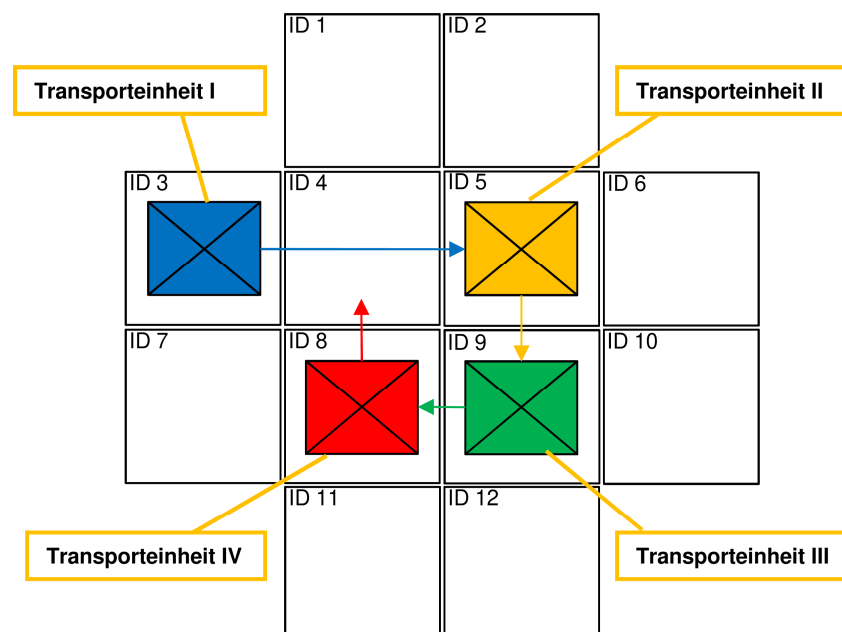


Abbildung 2.4: Verklemmungssituation in einem Kreisverkehr, nach [May09]

Wird hingegen ein Deadlock-Token an ein Modul gesendet, welches von keiner Transporteinheit belegt wird, wird ein Deadlock-Clearance-Token entgegen der Routenrichtung zu dem Modul zurückgesendet, welches den ersten Deadlock-Token ausgesendet hat. Sobald dieses Modul den Deadlock-Clearance-Token erhält, wird die wartende Transporteinheit aufgenommen, da keine Verklemmung droht.

Zusammengefasst führen diese Regeln dazu, dass in einen Kreisverkehr nur so lange Transporteinheiten eingeschleust werden können, bis nur noch ein Modul frei ist. Sobald nur noch ein Modul frei ist, kann im Kreisverkehr kein Deadlock-Clearance-Token mehr erzeugt werden.

Der Algorithmus von MAYER ist nicht zeitfensterbasiert. Dies kann in bestimmten Layouts dazu führen, dass es konzeptionell unmöglich ist die optimale Route zu reservieren. MAYER hat dies selbst an dem folgenden Szenario dargestellt (siehe Abbildung 2.5).

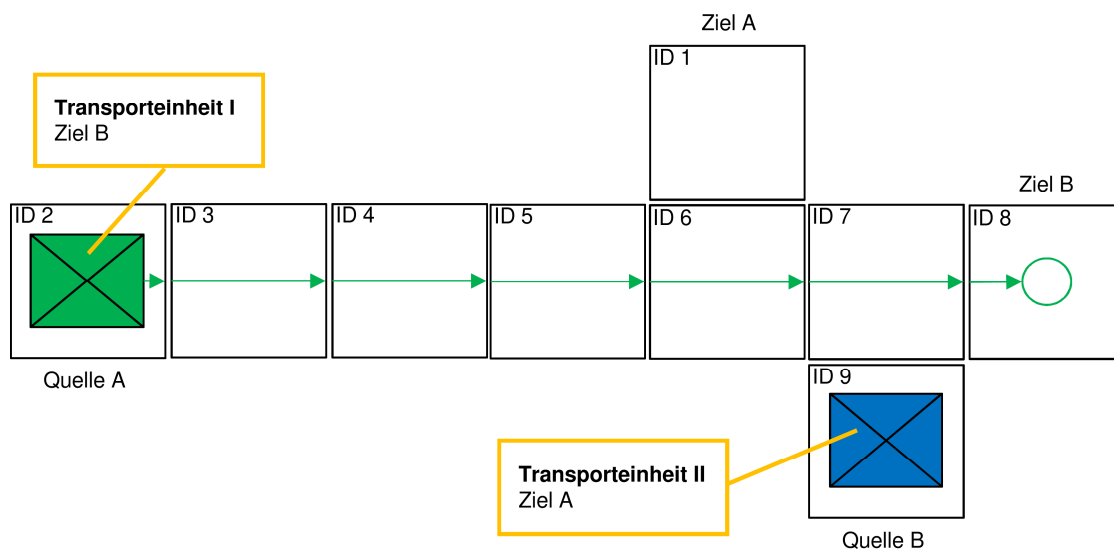


Abbildung 2.5: Szenario, in dem die Benutzung des Algorithmus von MAYER nachteilig ist, angelehnt an [May09]

In dem Szenario ist für die Transporteinheit I bereits eine Route vollständig zu dem Ziel B reserviert. Auf der Quelle B wird nun die Transporteinheit II eingeschleust. Da in den Routingeinträgen keine Zeitfenster abgespeichert werden, wissen die Module 6 und 7 nicht, wann die Transporteinheit I transportiert wird. Wenn nun das Modul 9 eine Anfrage-Nachricht an das Modul 7 sendet, muss das Modul 7 diese Anfrage-Nachricht ablehnen, um Kollisionen oder Verklümmungen zu verhindern. Erst nach dem Transport der Transporteinheit I kann die Reservierung einer Route für die Transporteinheit II erfolgen. Somit ist die Dauer des Transportes (Dauer der Routenreservierung und Dauer des Fördervorganges) der Transporteinheit II von der Dauer des Fördervorganges von der Transporteinheit I abhängig. In dem Beispiel ist der Vorteil des Einsatzes von Zeitfenstern ersichtlich: Da die Transporteinheit I noch weit von der Engstelle entfernt ist, kann die Transporteinheit II die Engstelle überqueren, bevor die Transporteinheit die Engstelle erreicht. MAYER selbst kommt zu folgendem Schluss: „Eine zeitdiskrete Reservierung könnte jedoch in bestimmten Systemen von Vorteil sein und sollte daher in zukünftigen Forschungsprojekten weiter analysiert werden.“ (Engl.:

„However, a time-discrete reservation could be of advantage in certain systems and therefore should be analyzed further in future research projects.”).

Der Algorithmus von SEIBOLD, der in [Sei16b] beschrieben ist, baut auf den Erkenntnissen von MAYER auf. Die Initialisierungsphase ist, bis auf vernachlässigbare Details, identisch. Die Vermeidung von Verklemmungen auf Grund von entgegengesetzten Routen erfolgt genau wie bei MAYER durch die zweistufige Routenreservierung. Im Gegensatz zu MAYER erfolgt jedoch die Vermeidung von Verklemmungen in Kreisverkehren bereits in der Phase der Routenreservierung. Hierzu setzt SEIBOLD das Prinzip der logischen Zeit ein. Die logische Zeit unterscheidet sich von der physikalischen Zeit dadurch, dass die logische Zeit immer nur dann voranschreitet, wenn ein Ereignis auftritt. Zwischen dem Auftreten von zwei Ereignissen kann eine beliebig lange physikalische Zeitspanne liegen. Das Prinzip der logischen Zeit wurde zuerst von LAMPORT in [Lam78] beschrieben. Die Steuerung jedes einzelnen Förderers verfügt über eine eigene logische Zeitbasis. Diese logische Zeit schreitet immer dann voran, wenn eine Transporteinheit auf den Förderer herauf- oder hinunterbefördert wird. Falls die Transporteinheit eine längere physikalische Zeitspanne auf einem Förderer warten muss, bevor der Transportvorgang fortgesetzt werden kann, schreitet die logische Zeit folglich nicht voran.

Eine Route setzt sich aus mehreren Routenabschnitten zusammen. Auf jedem Modul entlang der Route wird hierbei ein Routenabschnitt reserviert. Jeder Routenabschnitt verfügt über die Transporteinheiten-ID und aus welcher Richtung ein- und ausgefördert wird. Zusätzlich wird ein logisches Zeitfenster festgelegt: Dieses beginnt mit dem Einschleusen der Transporteinheit und endet mit dem Ausschleusen. SEIBOLDS Algorithmus verhindert Kollisionen und Verklemmungen dadurch, dass die logischen Zeitfenster sich nicht überlappen dürfen, also niemals das gleiche Zeitfenster auf dem gleichen Förderer für zwei verschiedene Transporteinheiten vergeben wird. Für eine detaillierte Beschreibung wird auf die Arbeit von SEIBOLD verwiesen.

Da die logische Zeit nicht kontinuierlich voranschreitet, ist es möglich, dass zwei benachbarte Förderer über eine unterschiedliche logische Uhrzeit verfügen. Wenn nun ein Förderer eine Routinganfrage an einen benachbarten Förderer sendet, sendet dieser auch immer seine Uhrzeit mit. Liegt die Uhrzeit des empfangenen Förderers in der Vergangenheit, setzt der empfangene Förderer seine Uhrzeit nach vorne, um seine Uhrzeit an die Uhrzeit des sendenden Förderers anzugleichen. Somit wird eine Reservierung in der Vergangenheit verhindert. Dies ist notwendig, um Verklemmungen zu

vermeiden. Nachteilig ist, dass es durch das Verbot der Reservierung in der Vergangenheit auch durch den Algorithmus von SEIBOLD nicht möglich ist, die optimale Route in dem Beispielszenario von Mayer (siehe Abbildung 2.5) zu reservieren.

Der Wissenschaftler GUE hat in wechselnder Zusammenarbeit mit anderen Wissenschaftlern mehrere Routingalgorithmen für FlexConveyor-Module entwickelt. Der Fokus der Routingalgorithmen liegt jedoch hierbei weniger auf dem Transport, sondern auf dem Lagern von Transporteinheiten. Die verschiedenen von GUE ET AL. entwickelten Algorithmen besitzen unterschiedliche Zielfunktionen. Die Zielfunktion des GridSequence-Systems (siehe [Gue12]) besteht darin, Transporteinheiten in einer vorgegebenen Sequenz aus dem Lager auszuschleusen. Im GridStore-System (siehe [Gue14]) besteht die Zielfunktion darin, immer nur einzelne Transporteinheiten auszuschleusen. Da die Zielfunktionen beider Systeme sehr ähnlich sind, unterscheiden sich die verwendeten Steuerungsalgorithmen nur in Details. Stellvertretend für beide Systeme wird der GridStore-Steuerungsalgorithmus vorgestellt.

Auf Grund der hohen Dichte von Transporteinheiten im Lager ist die Verklemmungsgefahr ohne einen Routingalgorithmus hoch. Der von GUE entwickelte Algorithmus verhindert das Entstehen von Verklemmungen durch die Einführung von Restriktionen. Die erste Restriktion ist, dass es jeweils nur einen einzigen Ein- und Ausschleusebereich geben darf. Der Ein- und Ausschleusebereich ist hierbei an entgegengesetzten Seiten des Lagers. Die nächste Restriktion besteht darin, dass keine Transporteinheiten entgegen der Richtung zum Ausschleusebereich bewegt werden dürfen. Weiterhin dürfen nur Transporteinheiten, die ausgeschleust werden müssen, in Richtung des Ausschleusebereiches bewegt werden. Alle anderen Transporteinheiten dürfen sich nur orthogonal zu den vorherigen Richtungen bewegen. Bewegungen in Richtung des Ausschleusebereiches werden von GUE als Zeilenbewegungen bezeichnet und Bewegungen orthogonal dazu als Reihenbewegungen.

Im Gegensatz zu den Algorithmen von MAYER und SEIBOLD sind bidirektionale Ausführungen der logistischen Funktionen durch den Algorithmus von GUE nicht möglich. Die Auswirkungen der Restriktionen können anschaulich als Graph visualisiert werden: In Abbildung 2.6 sind alle möglichen Routen dargestellt, die eine Transporteinheit durch das System nehmen kann.

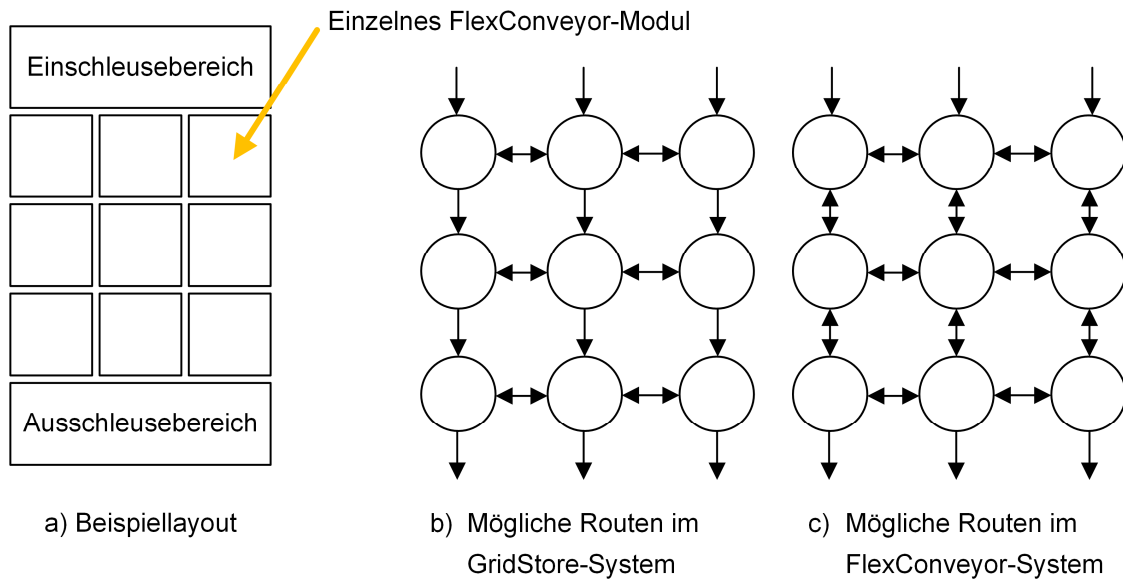


Abbildung 2.6: Vergleich der möglichen Routen des GridStore-Systems und des FlexConveyor-Systems

Das GridStore-System wurde weiterentwickelt zu dem GridPick-System [Ulu14]: Die Weiterentwicklung besteht darin, dass die Transporteinheiten im Ausschleusebereich nicht aus dem System entfernt werden, sondern nach der Entnahme einzelner Güter in das Lagerinnere zurückfahren. Somit ist das System in der Lage, bidirektionales Transportieren zu ermöglichen und besitzt dadurch eine größere Flexibilität als das GridStore-System. Die verfügbaren logistischen Funktionen sind identisch zu denen des GridStore-Systems.

Im GridStore-System wurde die Verklammerungsvermeidung sichergestellt durch die Restriktion, dass zwischen den Reihen nur in eine Richtung transportiert werden darf. Da diese Restriktion im GridPick-System entfällt, ist der Steuerungsalgorithmus für das GridPick-System wesentlich komplexer. Eine genaue Erläuterung der Funktionsweise des Steuerungsalgorithmus ist in [Ulu14] dargestellt. ULUDAĞ beweist die Verklammerungsfreiheit in einem Lager, welches maximal aus 16 FlexConveyor-Modulen bestehen kann, durch Ausprobieren aller möglichen Zustände (Engl. brute-force search). Da jedoch der Zustandsraum exponentiell ansteigt, konnte ULUDAĞ für Layouts mit einer größeren Anzahl als 16 FlexConveyor-Module nicht alle möglichen Zustände ausprobieren. Die Verklammerungsfreiheit für größere Lager wird von ULUDAĞ vermutet, aber nicht bewiesen.

Aufbauend auf den hier vorgestellten Forschungsergebnissen, wurde die Firma flexlog GmbH gegründet, die als Produkt weiterentwickelte FlexConveyor-Module vertreibt. Das Produktportfolio der flexlog GmbH schließt neben den FlexConveyor-Modulen auch noch Ausschleuser und Bandförderer ein (siehe Abbildung 2.7). Das Gesamtsys-

tem wird unter dem Produktnamen „FlexTechnology“ vertrieben. Jeder einzelne Fördertyp verfügt über eine eigene Steuerung und über einheitliche Schnittstellen, mit denen jeder Fördertyp mit jedem anderen Fördertyp verbunden werden kann [Fle20]. Über die einheitlichen Schnittstellen werden die einzelnen Förderer sowohl mit Energie als auch mit Informationen versorgt. Es muss nur ein Förderer des Fördersystems mit einer Spannungsquelle verbunden sein, um das gesamte Fördersystem mit Energie zu versorgen. Weiterhin erfolgt über die einheitliche Schnittstelle auch die Erkennung, welcher benachbarte Förderer angeschlossen ist.

Das Ergänzen von Bandförderern und Ausschleusern zu den FlexConveyor-Modulen stellt eine wirtschaftlich sinnvolle Kombination dar: Die FlexConveyor-Module und Ausschleuser werden als Kreuzungspunkte genutzt und die Bandförderer eignen sich, um wirtschaftlich größere Transportstrecken zu überwinden. Die Bandförderer sind auch steuerungstechnisch interessant: Während die Steuerung der FlexConveyor-Module auf das Transportieren von Transporteinheiten ausgelegt ist, die genauso groß sind wie eines der FlexConveyor-Module, ist dies bei den Bandförderern nicht mehr der Fall: Ein einzelner Bandförderer kann mehrere Transporteinheiten gleichzeitig transportieren. Da, im Gegensatz zu den FlexConveyor-Modulen, die Transporteinheiten auf den Bandförderern nun nicht mehr individuell bewegt werden können, muss dies in der Steuerung berücksichtigt werden. Leider hat die flexlog GmbH keine Veröffentlichungen zu ihren Steuerungsalgorithmen bereitgestellt, sodass öffentlich nicht bekannt ist, wie die Steuerung für die Bandförderer funktioniert.

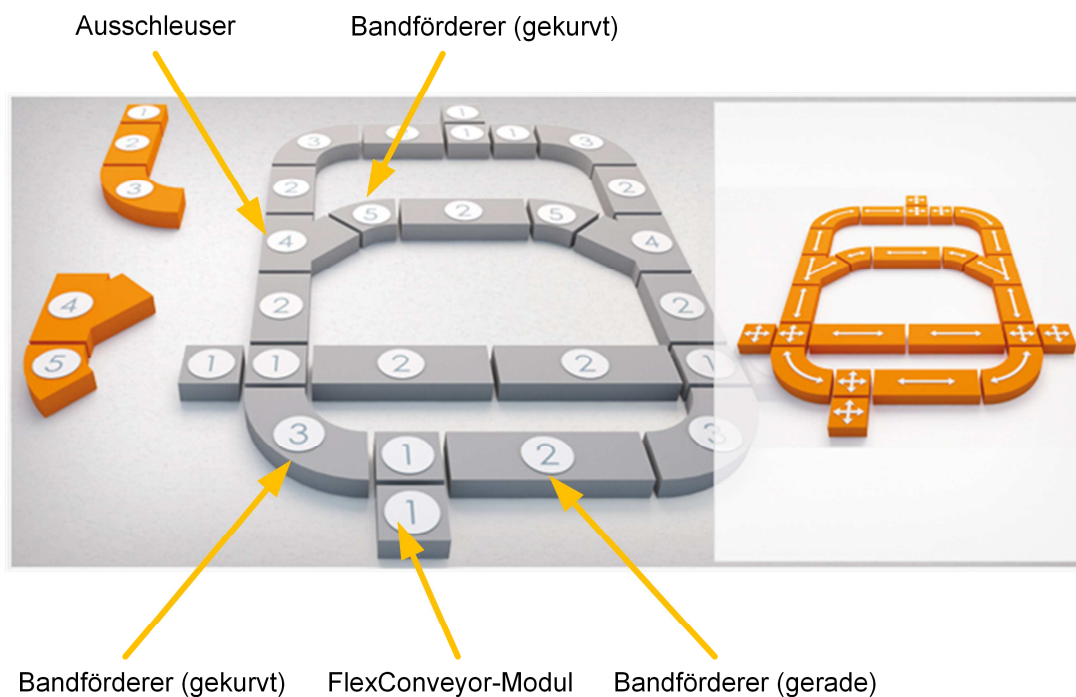
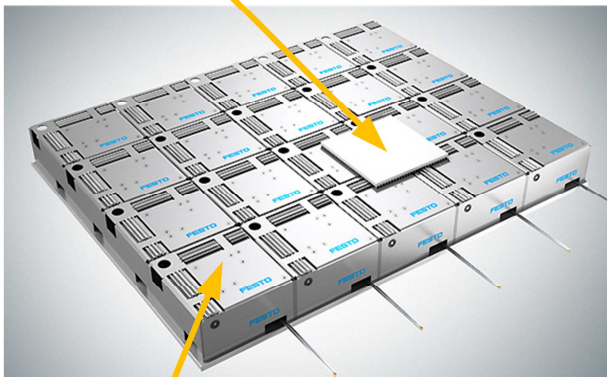


Abbildung 2.7: FlexTechnology, nach [Fle20]

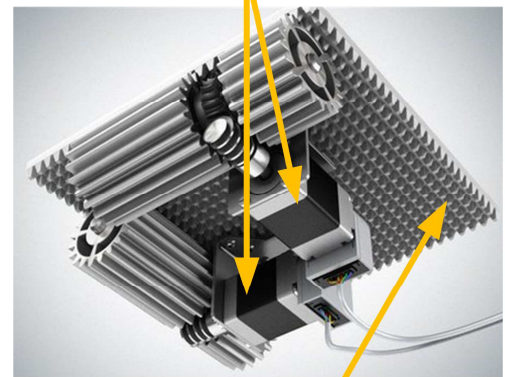
Neben der flexlog GmbH haben auch andere Firmen ähnliche Konzepte verfolgt: Die Firma Festo AG & Co. KG hat ein System namens Motion Cube vorgestellt (siehe Abbildung 2.8) [Wek17]. Das Motion Cube System besteht aus einzelnen Modulen, die über zwei Motoren verfügen, die rechtwinklig zueinander angeordnet sind. An den Wellenenden der Motoren sind Zahnräder angebracht, mit denen Platten frei auf dem System bewegt werden können. Die Unterseite der Platten ist mit omnidirektionalen Zahnrädern ausgestattet, die in [Tad12b] und [Tad12a] genauer dargestellt werden. Keinerlei Informationen wurde bisher über die verwendeten Steuerungsalgorithmen veröffentlicht. Die Firma Festo hat aus ungeklärten Gründen alle Veröffentlichungen zu dem Motion Cube System zurückgezogen.

Platte mit omnidirektionalen Zahnrändern



Einzelnes Modul

Motoren eines einzelnen Moduls



Unterseite der Platte mit omnidirektionalen Zahnrädern

Abbildung 2.8: Festo Motion Cube, nach [Wek17]

Ein weiteres großskaliges Fördersystem ist das Karis Pro System. Dieses unterscheidet sich von den bisher vorgestellten Fördersystemen dadurch, dass verschiedene Betriebsarten möglich sind. Die erste Betriebsart des Karis Pro Systems besteht darin, dass ein Einzelelement nur eine Transporteinheit transportiert. Wenn jedoch Transporteinheiten bewegt werden müssen, die größer als ein Modul sind, schließen sich mehrere Module zu einem Unstetigcluster zusammen und transportieren zusammen die Transporteinheit. Falls eine größere Anzahl von Transporteinheiten transportiert werden muss, schließen sich die einzelnen Module zu einem Stetigcluster zusammen. Da jedes Modul auf der Oberseite über eine Bewegungsvorrichtung verfügt, können Transporteinheiten bewegt werden, ohne dass sich die Einzelelemente bewegen müssen.

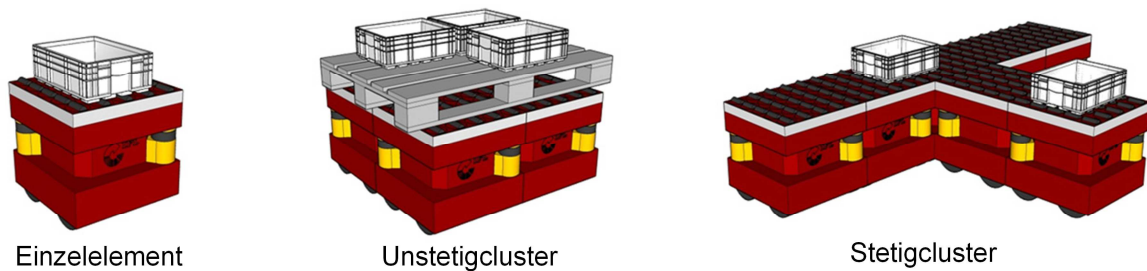


Abbildung 2.9: *Karis Pro* [Sto12]

Obwohl das Karis Pro System in drei verschiedenen Betriebsarten genutzt werden kann, wurde nur für den Stetigcluster der Steuerungsalgorithmus veröffentlicht (siehe [Sto12]). Steuerungsalgorithmen für die anderen Betriebsarten werden im Abschlussbericht zu dem Forschungsprojekt nicht detailliert erläutert [IFL16].

Der Steuerungsalgorithmus für die Stetigcluster beschränkt sich darauf, zwischen einer einzigen Quelle und einer einzigen Senke die günstigste Anordnung von Karis Pro Modulen aufzustellen. Hierfür wurden zwei Algorithmen entwickelt, die sich stark an dem A*-Algorithmus (siehe [Har68]) oder an Random-Walk Algorithmen (siehe z.B. [Mad96]) orientieren. Da es nur eine Quelle und eine Senke gibt, können keine Verklemmungen auftreten. Auf Grund des geringen Funktionsumfangs der beiden Algorithmen zur Steuerung der Stetigcluster werden diese innerhalb dieser Dissertation nicht weiter betrachtet.

Von der Firma FL Technology GmbH wird das fluide Logistik System beworben, welches eine hohe konzeptionelle Ähnlichkeit zu dem Motion Cube und dem Karis Pro System aufweist [FLT20]. Genau wie das Motion Cube System besteht das fluide Logistik System aus zwei Komponenten: den Planar-Einheiten und den Movern. Zunächst muss der Arbeitsbereich mit Planar-Einheiten ausgestattet werden. Auf diesen Planar-Einheiten können sich die Mover bewegen. Die Bewegung erfolgt, indem zwischen den Planar-Einheiten und den Movern ein elektromagnetisches Feld erzeugt wird. Genau wie bei dem Karis Pro System ist es möglich, dass sich mehrere Mover zu einem Verbund zusammenschließen, um Transporteinheiten zu bewegen, die größer sind als ein einzelner Mover. Das System ist echtzeitfähig, dezentral gesteuert und beliebig erweiterbar. Die Steuerungsalgorithmen wurden von der Firma FL Technology GmbH nicht veröffentlicht. Das fluide Logistik System kann eingesetzt werden, um Transporteinheiten zu bewegen, zu sortieren oder deren Reihenfolge zu ändern.

Das AutoStore-System [Aut18] besteht aus mehreren Schächten, in denen Transporteinheiten aufeinander gelagert werden. Auf der Oberseite der Schächte fahren kleine Fahrzeuge entlang, die einzelne Transporteinheiten aus den Schächten anheben kön-

nen. Falls nun eine Transporteinheit ausgeschleust werden muss, schichten die Fahrzeuge selbstständig alle Transporteinheiten um, um die angeforderte Transporteinheit ausschleusen zu können. Der Vorteil des Systems im Vergleich zu konventionellen Lagern ist, dass die Packungsdichte höher ist. Der Nachteil ist, dass die Auslagerungsvorgänge länger dauern. Der Hersteller gibt an, dass das System beliebig erweitert werden kann. Von dem AutoStore-System sind keine Steuerungsalgorithmen bekannt, aber auf Grund der konzeptionellen Ähnlichkeit zu dem GridStore- und GridPick-System, ist es möglich, dass die von GUE entwickelten Steuerungsalgorithmen für das AutoStore-System verwendet und weiterentwickelt wurden.



a) AutoStore Lager



b) Detailansicht einer Auslagerstation

Abbildung 2.10: AutoStore, nach [Aut18]

2.2.2 Kleinskalige Fördersysteme

Alle in diesem Kapitel vorgestellten kleinskaligen Module zeichnen sich dadurch aus, dass jedes einzelne kleinskalige Modul in der Lage ist, einen Kraftvektor, der tangential zu der Unterseite der Transporteinheit liegt, zu erzeugen. Zur Erzeugung dieses Kraftvektors verfügen die meisten kleinskaligen Module häufig über zwei Antriebe: Ein Förderantrieb und einem Schwenkantrieb, der den Förderantrieb ausrichtet. Der Schwenkantrieb bestimmt die Richtung des Kraftvektors und der Förderantrieb bestimmt die Größe des Kraftvektors. Aus diesem Aufbau ergibt sich eine hohe theoretische Prozessflexibilität: Wie in [Ven12] gezeigt wurde, können durch kleinskalige Fördersysteme alle logistischen Funktionen (siehe Kapitel 2.1.3) realisiert werden. Die geometrische Ausprägung kann sich zwischen den einzelnen Modulen unterscheiden (z.B. quadratische vs. sechseckige Ausprägung). VENTZ hat weiterhin festgestellt, dass sich eine quadratische Ausprägung mit dem geringsten Aufwand in eine bestehende

Fördertechnik integrieren lässt. Kleinskalige Fördersysteme verfügen über Funktions-
 emergenz: Während eine kleine Anzahl von Modulen nur wenige logistische Funktio-
 nen wie beispielsweise das Rotieren und Transportieren bereitstellen, können bei einer
 größeren Anzahl an Modulen auch komplexere logistische Funktionen wie beispiels-
 weise das Sequenzieren realisiert werden. Das Sequenzieren wird möglich, wenn
 Transporteinheiten zwischengespeichert werden können, bis diese in der geforderten
 Sequenz abtransportiert werden müssen. Diese Zwischenspeicherplätze werden
 durch das Hinzufügen von Modulen ermöglicht.

Innerhalb dieses Kapitels werden zunächst die unterschiedlichen Hardwarebauformen
 von kleinskaligen Förderelementen vorgestellt und anschließend die veröffentlichten
 Routingalgorithmen.

Im cogniLog-Forschungsprojekt wurde ein Modul entwickelt (siehe Abbildung 2.11),
 welches den Kraftvektor über eine Pendelscheibe erzeugt. Ein Schwenkmotor ist für
 die Ausrichtung der Scheibe verantwortlich und ein Fördermotor treibt die Pendel-
 scheibe an. Weiterhin verfügt jedes Modul über einen Lichttaster mit dem ermittelt wird,
 ob das Modul gerade von einer Transporteinheit überdeckt ist. Innerhalb des Projektes
 wurde ein Fördermatrix-Prototyp realisiert, der 6 x 6 Module aufnehmen kann. Hierbei
 wurden 16 angetriebene Module eingesetzt und 20 passive Module verwendet. Der
 Aufbau der Hardware wird in den folgenden Veröffentlichungen ausführlich dargestellt:
 [Krü15, Krü16].

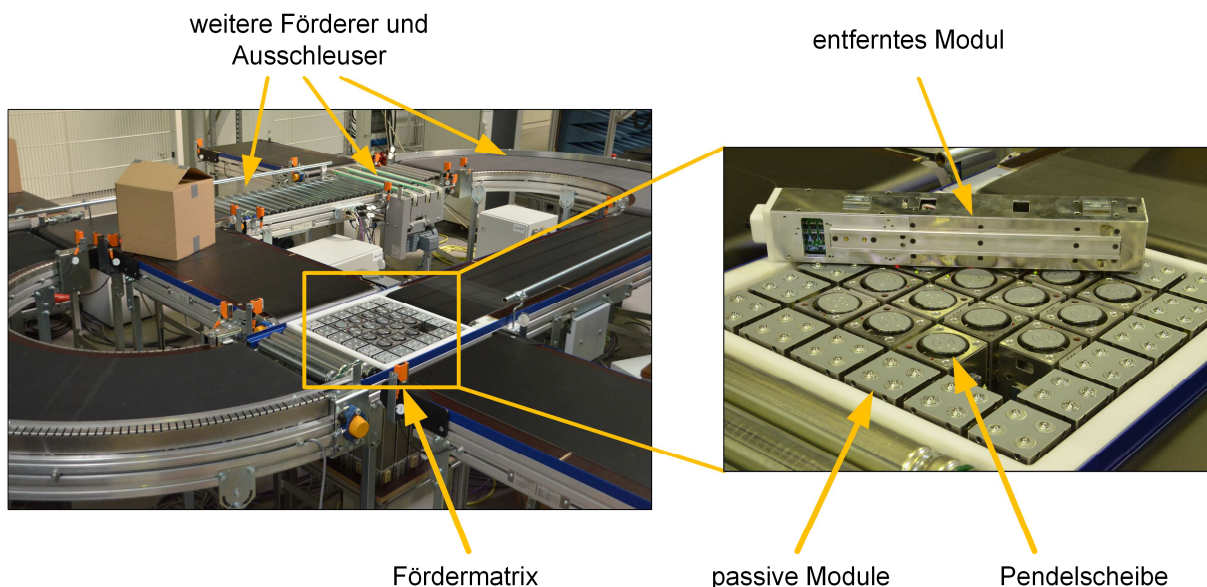


Abbildung 2.11: *cogniLog-Module, nach [Krü15]*

Aufbauend auf den Ergebnissen vom cogniLog, wurden die Module im netkoPs-Fors-
 chungsvorhaben weiterentwickelt. Statt einer Pendelscheibe, wird nur eine angetrie-

bene Motorrolle geschwenkt. Die Hardware der netkoPs-Fördermodule dient im Rahmen dieser Dissertation als Referenzfördersysteme und wird ausführlicher in Kapitel 4 dargestellt.

Die Firma Itoh Denki Ltd. hat ebenfalls ein kleinskaliges, flexibles Fördersystem entwickelt: Das „Magic Carpet System“ (MCS). Das MCS-System besteht aus einzelnen Modulen (siehe Abbildung 2.12). Jedes Modul verfügt über vier Drehteller, die unabhängig voneinander gedreht werden können. Auf jedem Drehteller ist ein Antrieb vorhanden. Die einzelnen Module verfügen über keine eigene Sensorik.

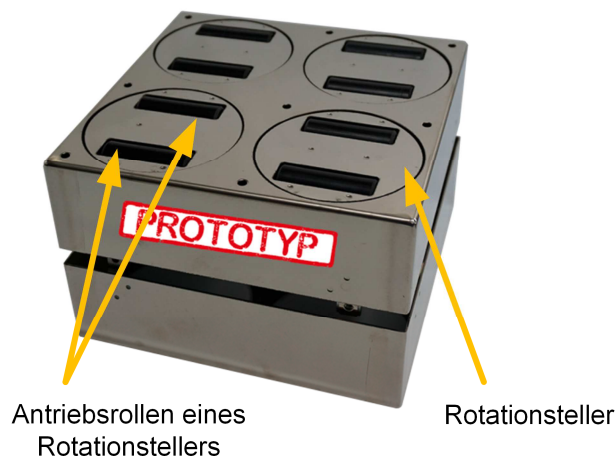


Abbildung 2.12: MCS-Modul, [Ito17]

In dem Celluveyor-Projekt wurden von URIARTE ET AL. kleinskalige, hexagonale Module entwickelt (siehe Abbildung 2.13). Ein einzelnes Modul verfügt über drei omnidirektionalen Räder. Jedes dieser einzelnen Räder wird mit einem eigenen Motor angetrieben. Im Gegensatz zu den netkoPs- oder cogniLog-Modulen verfügen die Celluveyor-Module über keinen eigenen Sensor. Stattdessen muss die Position der Transporteinheiten durch ein externes System wie beispielsweise eine Kamera erfasst werden. [Uri15]

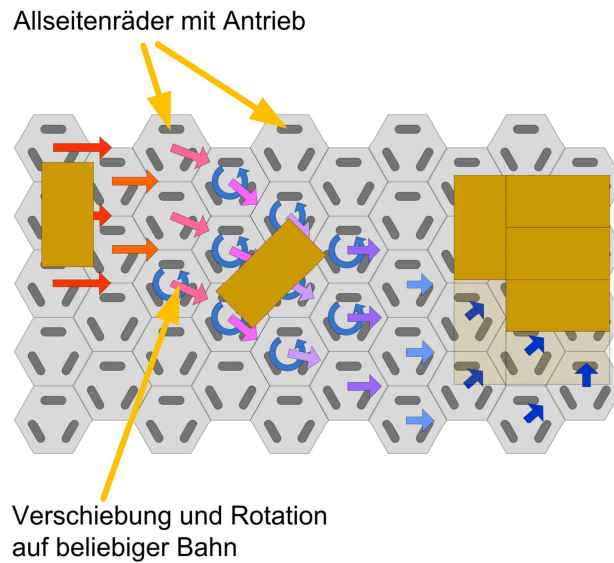


Abbildung 2.13: Celluveyor-Module, nach [Uri13]

Sowohl für das MCS-Modul als auch für das Celluveyor-Modul wurden keine Steuerungsalgorithmen veröffentlicht, weil beide als kommerzielle Produkte vertrieben werden sollen. Die Hersteller beider Systeme haben kein wirtschaftliches Interesse daran, die Funktionsweise der Steuerungsalgorithmen zu offenbaren. Der einzig bekannte Steuerungsalgorithmus für kleinskalige Fördersysteme, der in der Lage ist Verklemmungen zu verhindern, wurde von KRÜHN in [Krü15] veröffentlicht.

KRÜHNS Routingalgorithmus hat als Zielfunktion das Transportieren. Um diese Zielfunktion zu realisieren, werden verschiedene Hilfsfunktionen genutzt. Weiterhin hat SOHRT in [Soh14] einen Algorithmus zum Zwischenspeichern von Transporteinheiten entwickelt, der auf den Arbeiten von KRÜHN aufbaut. KRÜHN hat weiterhin ein Konzept entwickelt, wie das Zwischenspeichern genutzt werden kann, um das Sequenzieren auf kleinskaligen Fördermodulen zu ermöglichen. Die Prozessflexibilität der kleinskaligen Fördermodule ist somit hoch.

KRÜHNS Routingalgorithmus baut auf den Arbeiten von MAYER auf. Um die von MAYER entwickelten Routingalgorithmen zur Steuerung der kleinskaligen Fördermodule zu nutzen, hat KRÜHN das Konzept der Modulnachbarschaften eingeführt (siehe Abbildung 2.14). Für jede Transporteinheit wird eine individuelle Nachbarschaft erzeugt. Die Form der Nachbarschaft entspricht dabei der Form der Transporteinheit. Sobald die Transportdurchführungsphase beginnt, „wandert“ die transporteinheitenindividuelle Nachbarschaft mit der Transporteinheit durch das System. Module werden während des Transportvorgangs dynamisch zu der Nachbarschaft hinzugefügt, wenn sie benötigt werden und entfernt, sobald sie nicht mehr benötigt werden. Die Nachbarschaften von KRÜHN sind aus steuerungstechnischer Sicht äquivalent zu einem einzelnen

FlexConveyor-Modul. Hierdurch gelang es KRÜHN die Algorithmen von MAYER als Basis für seine Algorithmen zu nutzen. Durch das Konzept der transporteinheitenindividuellen Nachbarschaften ist die Produktflexibilität der kleinskaligen Fördererlemente sehr hoch: Solange nur ausreichend viele Module verfügbar sind, können Transporteinheiten mit beliebigen Formen transportiert werden.

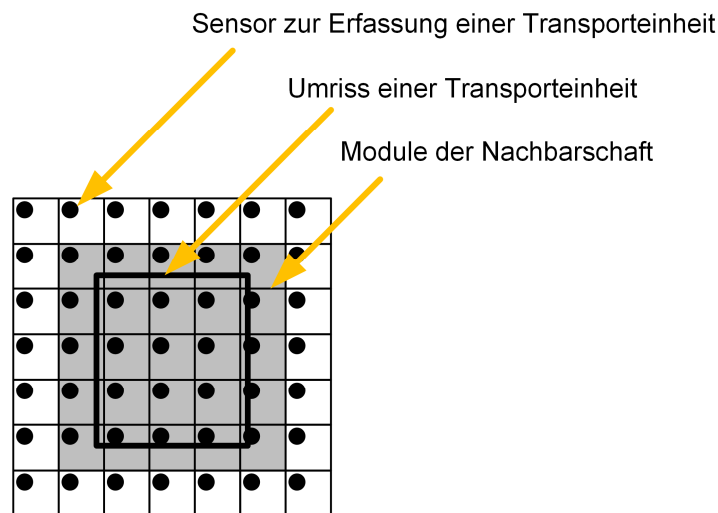


Abbildung 2.14: Modulnachbarschaft, nach [Krü15]

Die Erkennung und Vermeidung von Verklemmungen von KRÜHN basiert auf den Algorithmen von MAYER und verwendet daher auch Deadlock-Token. Im Gegensatz zu MAYER werden die Routingtabellen dynamisch angepasst, indem sowohl geplante als auch reservierte Routenabschnitte die Werte in den Routingtabellen verändern. Hierdurch gelingt es KRÜHN eine hohe Routingflexibilität zu erreichen: Im Gegensatz zu MAYER und SEIBOLD ist die Planung von Alternativrouten nicht nur vor dem Beginn der Transportdurchführungsphase möglich, sondern auch währenddessen.

Die Erweiterungsflexibilität von KRÜHNS Algorithmus ist ebenfalls hoch, da auch bei KRÜHNS Algorithmus alle Entscheidungen räumlich und zeitlich lokal getroffen werden. Es gibt somit keine theoretische Obergrenze für den Algorithmus und das Fördersystem kann beliebig skaliert werden.

2.3 Fahrerlose Transportfahrzeuge

Im Gegensatz zu den modularen Fördersystemen werden die fahrerlosen Transportfahrzeuge (FTF) nicht anhand ihrer Größe unterschieden, da diese keinen Einfluss darauf hat, welche Routingalgorithmen genutzt werden können. Es erfolgt daher zunächst eine kurze Darstellung der verfügbaren Hardwarebauformen und anschließend eine Übersicht über die verwendeten Routingalgorithmen.

2.3.1 Hardwarebauformen

FTF gibt es in verschiedenen Hardwarebauformen (siehe Abbildung 2.15). Jeder der fünf weltweit umsatzstärksten [Hen17] Hersteller von Flurfördersystemen bietet im Jahr 2020 FTF an [Cro20, Jun20, Lin20, Roc20, Toy20]. Der Hardwareaufbau ist dabei immer ähnlich: Ein FTF kann genormte Ladungsträger (z.B.: Europoolpalette) aufnehmen, diese transportieren und dann wieder abstellen. Die FTF können je nach Bauform dabei entweder in einem Lagersystem eingesetzt werden oder als reines Transportfahrzeug.

Die Art der Navigation ist vielfältig und reicht von dem Erkennen und Verfolgen von vergebenen Bahnen auf dem Boden zu der Erkennung der Umgebung durch Kamerasysteme [Ull14]. Auf welche Art die Navigation der FTF erfolgt, ist für das Routing der Fahrzeuge nebensächlich. Daher wird im Rahmen dieser Dissertation nicht detailliert die technische Realisierung der Navigationssysteme beschrieben, sondern diese als funktionierend vorausgesetzt.



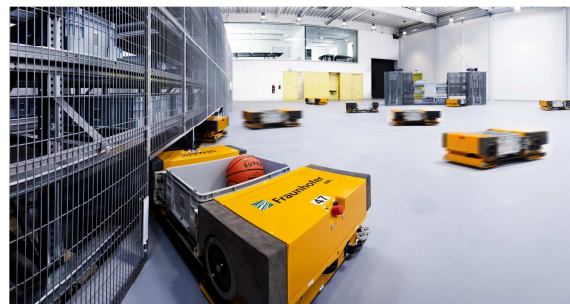
a) FTF der SSI Schaefer Fritz Schaefer GMBH



b) Eisenmann LogiMover



c) Amazon Robotics LLC



d) Zellulare Fördertechnik

Abbildung 2.15: *Verschiedene FTF-Bauformen, nach [Ama20, Fra17, Sig17, SSI17]*

2.3.2 Übersicht über Routingalgorithmen für FTF

Bevor einzelne Routingalgorithmen für FTF dargestellt werden, wird zuerst eine Übersicht über das Forschungsfeld der Steuerungen für FTF gegeben durch die Auswertung der drei aktuellsten Übersichtsveröffentlichungen (Engl. „survey paper“ oder auch

„review paper“). Diese sind: [Lea06, Vis06, Viv15]. Alle betrachteten Übersichtsveröffentlichungen gliedern die Steuerungsaufgabe von FTF in drei Hauptbereiche:

1. Arbeitsverteilung (Engl. „dispatching“)
2. Routenplanung (Engl. „routing“)
3. Ablaufplanung (Engl. „scheduling“)

Die Arbeitsverteilung, also die Zuweisung einzelner FTF zu den Transportvorgängen, wird im Rahmen dieser Dissertation nicht weiter betrachtet, da dieser Vorgang bei den modularen Fördersystemen entfällt (welcher Förderer eine Transporteinheit transportieren muss, ist bei den modularen Fördersystemen durch das Layout definiert).

Vis teilt die Algorithmen in statische und dynamische Algorithmen ein. Statische Algorithmen zeichnen sich dadurch aus, dass vor dem ersten Transportvorgang für jede mögliche Kombination von Transporteinheit, Quelle und Senke die optimale Route berechnet wird. Während des Betriebes des Fördersystems wird für ähnliche Transportvorgänge (gleiche Quelle und gleiche Senke) immer die vorherberechnete optimale Route gewählt. Bei den dynamischen Algorithmen erfolgt stattdessen die Routen- und Ablaufplanung erst kurz vor dem Beginn des Transportvorganges.

VIVALDINE ET AL. unterteilen die Algorithmen in offline- und online-Algorithmen. Bei den offline-Algorithmen sind für einen Arbeitszyklus alle Informationen vor dessen Beginn bekannt: Wann welche Transporteinheiten von welcher Quelle zu welcher Senke transportiert werden müssen und wie viele FTF dafür zur Verfügung stehen. Bei den online-Algorithmen werden diese Informationen erst unmittelbar vor dem Eintreffen einer Transporteinheit dem Fördersystem zur Verfügung gestellt. LE-ANH ET AL. verwenden die gleiche Unterteilung wie VIVALDINE ET AL. und kommen zusätzlich noch zu dem Schluss, dass offline-Algorithmen für den praktischen Einsatz häufig nicht geeignet sind, da selten alle Informationen über den Arbeitszyklus vor Beginn zur Verfügung stehen.

Die Definition der dynamischen Algorithmen von Vis deckt sich mit der Definition der online-Algorithmen, wie sie von VIVALDINE ET AL. und LE-ANH ET AL. verwendet wird. Im Folgenden werden nur dynamische bzw. online-Algorithmen betrachtet, da diese die Anforderungen eines in der industriellen Praxis eingesetzten, wandlungsfähigen Fördersystems erfüllen.

Die einzelnen Übersichtsveröffentlichungen sind in Bezug auf die Behandlung von Konfliktvermeidung unterschiedlich gegliedert. Zunächst werden von allen drei Veröffentlichungen Routingalgorithmen vorgestellt, bei denen die FTF als körperlos ange-

nommen werden. Das Ziel dieser Algorithmen besteht darin, mehrere Transportaufträge zu Touren zu gruppieren und auf die zur Verfügung stehenden FTF zu verteilen [Dan59]. Dieses Problem wird im Englischen als „Vehicle routing problem“ (kurz: VRP) bezeichnet und stellt eine Generalisierung des „Problems des Handlungsreisenden“ (Engl.: „Traveling Salesman problem“) dar. Auf Grund der Körperlosigkeit der FTF können keine Konflikte auftreten. Auf Grund der geringen Praxisrelevanz werden Algorithmen, die nur das VRP lösen, im Rahmen dieser Dissertation nicht weiter betrachtet.

In allen drei Veröffentlichungen werden zeitfensterbasierte Routingalgorithmen vorgestellt. Diese sind konzeptionell ähnlich zu dem Ansatz von SEIBOLD (siehe Kapitel 2.2.1). VIS und LE-ANH ET AL. stellen neben den zeitfensterbasierten Routingalgorithmen noch weitere Ansätze vor, die im Folgenden kurz erläutert werden.

Vis stellt Ansätze dar, bei denen das Auftreten von Verklemmungen verhindert wird durch ein geschicktes Design des Bahnlayouts, auf dem sich die FTF später bewegen. Weiterhin wird von Vis festgestellt, dass die Implementierung dieser Ansätze in der Regel zu einer Verringerung des Durchsatzes von Transporteinheiten führt. Darüber hinaus ist die Routingflexibilität dieser Ansätze gering: Häufig gibt es nur eine Bahn, die zu einem Ziel führt. Ansätze, die durch ein geschicktes Design des Bahnlayouts Verklemmungen vermeiden, werden daher im Rahmen dieser Dissertation nicht weiter betrachtet. Eine weitere Kategorie von Ansätzen, die Vis betrachtet, besteht darin, die Bahnen in einzelne Kontrollzonen einzuteilen. Die einzelnen FTF müssen sich dann vor dem Befahren dieser Kontrollzonen mit in der Nähe befindlichen FTF abstimmen. Vis beschreibt die Verwendung von Kontrollzonen als häufig verwendete Methode, wenn FTF in der industriellen Praxis eingesetzt werden. Nach Vis garantieren nicht alle Ansätze, die Kontrollzonen verwenden, eine Verklemmungsfreiheit. Die Kontrollzonen-Ansätze sind konzeptionell ähnlich zu dem Ansatz von MAYER und KRÜHN (siehe Kapitel 2.2).

Da im Rahmen dieser Arbeit ein zeitfensterbasierter Routingalgorithmus für modulare Fördermodule erforscht wird, werden im nachfolgenden Kapitel die zeitfensterbasierten Routingalgorithmen für FTF näher betrachtet.

2.3.3 Zeitfensterbasierte Routingalgorithmen für FTF

Zeitfensterbasierte Routingalgorithmen für FTF wurden sowohl für bidirektional als auch für unidirektional befahrbare Wegenetze veröffentlicht. Die Algorithmen für unidirektional befahrbare Wegenetze bieten aus Anwendungsperspektive einen geringeren Funktionsumfang und werden daher im Rahmen dieser Dissertation nicht betrachtet. Als Beispiele für Ansätze zur Steuerung von FTF auf unidirektionalen Wegenetzen

sind [Möh05, Smo10, Viv10] zu nennen. Ein Algorithmus ist zeitfensterbasiert, wenn zu jedem reservierten Routenabschnitt der Zeitraum angegeben wird, in dem der Abschnitt durch eine Transporteinheit belegt ist.

Der grundsätzliche Ablauf von zeitfensterbasierten Routingalgorithmen ist bei allen Ansätzen gleich und wird im Folgenden am Beispiel der Veröffentlichung [Kim91] von KIM & TANCHOCO erläutert, da es sich um eine der prägnantesten und am häufigsten zitierten Veröffentlichungen handelt. Der große Einfluss des Algorithmus auf nachfolgende Veröffentlichungen liegt darin begründet, dass immer die Route mit der kürzesten Transportzeit gefunden wird und somit die optimale Route. Nachteilig an dem Algorithmus ist, dass die Zeit für die Routenreservierung im schlechtmöglichsten Fall wie folgt ist: $O(v^4n^2)$. Dies bedeutet, dass die Anzahl der FTF v im System quartisch in die Reservierungszeit eingeht und die Anzahl der Kreuzungen n quadratisch eingeht. Da sich die Durchlaufzeit einer Transporteinheit aus der Reservierungszeit und der nachfolgenden Transportzeit zusammensetzt, kann der Algorithmus von KIM & TANCHOCO noch weiter verbessert werden, indem die Reservierungszeit reduziert wird. Jedoch muss beachtet werden, dass die Transportzeit den wesentlich größeren Anteil an der Durchlaufzeit ausmacht. Dies liegt darin begründet, dass die Geschwindigkeiten beider Prozesse um mehrere Größenordnungen auseinanderliegen: Während Nachrichten zwischen zwei Förderern innerhalb von Millisekunden ausgetauscht werden können, liegt die Übergabe von einer Transporteinheit von einem Förderer auf einen nachfolgenden Förderer im Sekundenbereich. Durch den steten Fortschritt im Bereich der Kommunikationstechnologie wird der Einfluss der Reservierungszeit auf die Durchlaufzeit noch weiter an Bedeutung verlieren.

Im Folgenden wird der Algorithmus von KIM & TANCHOCO beschrieben und darauf aufbauend wird eine Weiterentwicklung dargestellt, durch den der Algorithmus dazu befähigt wird, auf unvorhergesehene Abweichungen während des Transportes zu reagieren. Um Kollisionen zu verhindern definieren KIM & TANCHOCO eine Sicherheitszone um die einzelnen FTF. Statt den Abmessungen der FTF wird diese Sicherheitszone als Grundlage zur Planung der Routen genutzt.

Der Algorithmus wird auf einem zentralen Leitsystem ausgeführt, welches eine vollständige Übersicht über das Wegenetz besitzt und die Routen- und Ablaufpläne von allen FTF im System kennt. Die Planung von neuen Routen erfolgt streng sequenziell: Es wird immer nur eine Route geplant, während alle bereits reservierten Routen als statisch angesehen werden. Nachdem eine Route reserviert wurde, wird diese auf das FTF geladen und dieses beginnt mit der Transportdurchführung. Dies impliziert, dass

alle FTF über ausreichend genau zueinander synchronisierte Uhren verfügen. KIM & TANCHOCO erläutern jedoch nicht, wie die Synchronisierung der Uhren erfolgt.

Das Wegenetz wird von KIM & TANCHOCO als bidirektionaler Graph modelliert. Die Knoten des Graphen stellen die Kreuzungsstellen dar und die Kanten die Wegstrecken, welche die Kreuzungsbereiche verbinden. Zusätzlich erhält jeder Knoten einen Zeitplan, in dem aufgezeichnet wird in welchem Zeitraum die Kreuzung durch welches FTF belegt ist.

Auf den Quellen wird zunächst der frühestmögliche Zeitpunkt berechnet, zu dem der Transportvorgang begonnen werden kann. Dieser frühestmögliche Zeitpunkt ergibt sich aus der maximalen Laufzeit des Algorithmus und aus der Übertragungszeit, die benötigt wird, um den fertig berechneten Routen- und Zeitplan an das entsprechende FTF zu senden. Der Algorithmus wird in eine Hauptprozedur und eine Nebenprozedur unterteilt. Zuerst wird die Nebenprozedur beschrieben, auf der die Hauptprozedur aufbaut.

Das Ziel der Nebenprozedur besteht darin, von einem Ausgangsknoten (im Folgenden als Knoten i bezeichnet) die am frühesten verfügbaren Zeitfenster von angrenzenden Knoten (im Folgenden als Knoten j bezeichnet) zu berechnen. Hierzu werden zunächst die direkt angrenzenden Knoten ermittelt (es erfolgt also die Überprüfung der räumlichen Durchführbarkeit). Anschließend erfolgt die Überprüfung der zeitlichen Durchführbarkeit: Hierzu wird für jeden Knoten j zunächst der früheste Zeitpunkt (im Folgenden als t bezeichnet) berechnet, zu dem das FTF den Knoten j befahren darf. In t wird sowohl die Zeit zum Befahren der Strecke zwischen den Knoten i und j berücksichtigt, als auch, ob das Zeitfenster des Knoten j groß genug ist, sodass das FTF den Knoten j wieder innerhalb des freien Zeitfensters verlassen kann. Die Überprüfung kann entweder ergeben, dass das Zeitfenster des Knoten blockiert ist oder erreichbar. Nachdem die Zeitfenster berechnet worden sind, wird überprüft, ob es auf der verbundenen Kante der beiden Knoten i und j zu Kollisionen kommen kann. Die zwei Möglichkeiten der Kollision sind:

1. Ein langsames FTF wird von einem schnelleren FTF eingeholt.
2. Es gibt einen Frontalzusammenstoß von zwei sich entgegenkommenden FTF.

Beide Möglichkeiten werden überprüft durch den geschickten Vergleich der Zeitpläne der Knoten i und j . Falls ein Konflikt festgestellt wird, wird das entsprechende Zeitfenster des Knoten j als blockiert abgespeichert. Es sei darauf hingewiesen, dass die Nebenprozedur auch in der Lage ist zu überprüfen, ob ein FTF kurzzeitig auf eine Kante

ausweichen kann, um ein anderes FTF durch den Knoten fahren zu lassen. Dies bedeutet algorithmisch, dass es sich bei dem Ausgangsknoten i und dem Zielknoten j um den gleichen Knoten handelt.

Die Hauptprozedur wird iterativ ausgeführt. Zur Erläuterung werden die Abläufe am Beispiel des ersten Iterationsschrittes beschrieben. Die Menge aller Zeitfenster F wird in drei einander ausschließende Untermengen unterteilt. Die Untermenge T enthält alle Zeitfenster zu denen schon eine konfliktfreie Route vom Startknoten aus berechnet wurde. Die Untermenge U enthält alle Zeitfenster, die direkt aus der Untermenge T erreichbar sind, also alle Zeitfenster die ohne Konflikte im nächsten Schritt reserviert werden können. Die dritte Untermenge enthält alle Zeitfenster, die nicht in U oder T sind. Der Algorithmus terminiert, sobald ein Zeitfenster gefunden wird, welches sich auf dem Zielknoten befindet. Solange jedoch noch kein Zeitfenster auf dem Zielknoten gefunden wurde, wird in jedem Iterationsschritt das Zeitfenster aus U ausgewählt, welches durch die Addition auf die bisher ermittelte Gesamtzeit eine neue, kürzere mögliche Gesamtzeit ergibt. Die Auswahl des günstigsten Zeitfensters aus U erfolgt durch den Aufruf der Nebenprozedur, mit der die frühesten möglichen Zeitfenster der erreichbaren Knoten ermittelt werden. Nachdem das günstigste Zeitfenster aus U für diesen Iterationsschritt bestimmt wurde, wird dieses aus U entfernt und zu T hinzugefügt. Jedes Mal, wenn ein Zeitfenster aus U nach T verschoben wird, wird in zwei Listen gespeichert, welche Kante genutzt wurde, um von einem Knoten zum nächsten zu kommen und welches Zeitfenster auf dem nachfolgenden Knoten verwendet wurde. Sobald ein Zeitfenster auf dem Zielknoten erreicht wurde, werden diese beiden Listen benutzt, um die Route abschließend vom Zielknoten zurück zum Startknoten zu reservieren und der Algorithmus terminiert erfolgreich (die Listen werden zur Rückverfolgung der bisher ermittelten Zeitfenster und Strecken genutzt). Falls beim Starten des nächsten Iterationsschrittes keine Zeitfenster mehr in U vorhanden sind, terminiert der Algorithmus ebenfalls, da es nicht möglich ist, eine Route vom Start- zum Zielknoten zu reservieren. Die hier beschriebene Hauptprozedur basiert auf dem Finden der kürzesten Route in einem ungerichteten Graph, wie es von DIJKSTRA in [Dij59] beschrieben wurde. Die Optimierungen für DIJKSTRAS Algorithmus können daher auch auf diese Hauptprozedur angewendet werden.

Der Algorithmus von KIM & TANCHOCO wurde entscheidend von MAZA & CASTAGNA in [Maz05] weiterentwickelt, indem die Robustheit erhöht wurde. Falls das FTF-System in Bereichen eingesetzt wird, in denen sich auch Menschen aufhalten können, kann es zu unplanmäßigen Verzögerungen kommen (weil z.B. ein FTF stoppen muss, um nicht mit einem Menschen zusammenzustoßen). Durch diese Verzögerungen wird der

vorher berechnete Zeitplan ungültig, sodass es im weiteren Betrieb zu Konflikten kommen kann. MAZA & CASTAGNA lösen dieses Problem, indem bei außerplanmäßigen Verzögerungen zwar nicht mehr die Zeitfenster eingehalten werden, aber die vorher berechnete Reihenfolge der FTF an den Knoten beibehalten wird. Dieser Ansatz ist konzeptionell ähnlich zu dem Ansatz von SEIBOLD, der in [Sei16b] beschrieben wurde.

Die Praxisrelevanz des Algorithmus wurde hierbei von MÖHRING ET AL. in [Möh05] überprüft, indem dieser zur Routenreservierung im Container Terminal Altenwerder im Hamburger Hafen genutzt wurde. Das Wegenetz bestand dabei aus 30.000 Kanten und insgesamt mussten 42 Fahrzeuge gleichzeitig geroutet werden. Der Algorithmus terminierte dabei immer schneller als 0,5 s. Der zentrale Leitreechner hatte die folgende Hardware verbaut: AMD-Athlon 2100+ (1,7 MHz) mit 512 MB RAM. Es ist somit davon auszugehen, dass die Reservierungszeit keinen relevanten Einfluss auf die Durchlaufzeit hat und somit die Optimierung der Reservierungszeit für den Praxiseinsatz nicht erforderlich ist.

2.4 Vergleichende Übersicht der betrachteten Algorithmen

Ziel dieses Kapitels ist es, einen kompakten Überblick über die bereits existierenden Algorithmen zu liefern, die in den vorherigen Kapiteln vorgestellt worden sind. Mit Hilfe dieses Überblickes wird anschließend die wissenschaftliche Fragestellung dieser Dissertation identifiziert. Die in den vorangegangenen Kapiteln beschriebenen Steuerungsalgorithmen werden hierzu zunächst vergleichend anhand der in dem Kapitel 2.1 beschriebenen Anforderungen gegenübergestellt. Zunächst wird in Tabelle 2.2 dargestellt, wie die einzelnen Algorithmen den Betrieb sicherstellen.

Tabelle 2.2: Sicherstellung des Betriebes durch die einzelnen Algorithmen

Eigenschaft	Algorithmus				
	MAYER	KRÜHN	SEIBOLD	KIM & TANCHOCO	GUE ET AL.
Verhinderung von Kollisionen?	Ja	Ja	Ja	Ja	Ja
Verhinderung von Verklemmungen?	Ja	Ja	Ja	Ja	Ja
Verhinderung von Livelocks?	Nicht untersucht	Nicht untersucht	Ja	Ja	Nicht untersucht
Verhinderung vom Verhungern?	Nicht untersucht	Ja	Ja	Ja	Nicht untersucht

Es ist nicht überraschend, dass alle Steuerungsalgorithmen die Verhinderung von Kollisionen und von Verklemmungen sicherstellen. Besonders hervorzuheben sind an dieser Stelle die Algorithmen von SEIBOLD und KIM & TANCHOCO, da beide explizit sicherstellen, dass Livelocks und das Verhungern verhindert wird. Dies bedeutet jedoch nicht, dass bei der Verwendung der anderen Algorithmen Livelocks oder das Verhungern auftritt, sondern lediglich, dass die Autoren diese Aspekte nicht untersucht haben.

Als nächstes wird die Fähigkeit der Algorithmen beurteilt, die Leistungskenngrößen zu erfüllen. Leider wurden nur im Falle von SEIBOLD die Leistungskenngrößen für einen großen Parameterraum (unterschiedliche Layouts, Anzahl der eingeschleusten Transporteinheiten pro Zeitintervall, etc.) veröffentlicht. Daher können die bisher betrachteten Algorithmen nicht direkt anhand dieser Größen verglichen werden. Stattdessen sind in der nachfolgenden Tabelle 2.3 Eigenschaften aufgelistet, die einen Einfluss auf die Leistungskenngrößen haben.

Tabelle 2.3: Beurteilung der Leistungsfähigkeit durch die Algorithmen

Eigenschaft	Algorithmus				
	MAYER	KRÜHN	SEIBOLD	KIM & TANCHOCO	GUE ET AL.
Wurden die wichtigsten Leistungs-kenngrößen ermittelt?	Nein	Teilweise	Ja	Nein	Teilweise
Maximale Dauer der Routenreservierung?	Keine Aussage	Keine Aussage	Keine Aussage	$O(v^4n^2)$	Routenreservierung erfolgt nicht
Zielgröße bei der Routenwahl?	Distanz	Distanz	Logische Zeit	Physikalische Zeit	Distanz
Wird immer die optimale Lösung gefunden?	Nein	Nein	Nein	Ja	Nein

Aus der Tabelle ist ersichtlich, dass sowohl der Algorithmus von KIM & TANCHOCO als auch der Algorithmus von SEIBOLD in Bezug auf die Leistungs-kenngrößen hervorzuheben sind. Für den Algorithmus von SEIBOLD wurden bereits für einen großen Parameterraum die Leistungs-kenngrößen ermittelt. Im Gegensatz hierzu wurden zwar für den Algorithmus von KIM & TANCHOCO keine Leistungs-kenngrößen ermittelt, aber es wurde gezeigt, dass dieser die optimale Route (also die zeitlich kürzeste Route) reserviert. Somit ist es unmöglich, dass ein anderer Routingalgorithmus Routen mit kürzeren Transportzeiten findet. Jedoch garantiert der Algorithmus von KIM & TANCHOCO nicht, dass die Zeit für die Routenreservierung die kürzest mögliche ist. Somit kann es sein, dass durch die Verwendung anderer Algorithmen die Durchlaufzeit, die sich aus der Reservierungszeit und der Transportzeit zusammensetzt, niedriger ist. Besonders hervorzuheben ist weiterhin, dass der Algorithmus von KIM & TANCHOCO als einziger Algorithmus in der Lage ist, in jedem beliebigen Layout die optimalen Lösungen zu finden. Die Anpassung des Algorithmus von KIM & TANCHOCO auf dezentral gesteuerte, modulare Fördersysteme ist daher vielversprechend in Bezug auf die Leistungs-kenngrößen. Die Beurteilung der Wandlungsfähigkeit der Algorithmen ist in der nachfolgenden Tabelle 2.4 dargestellt. Hierbei sind besonders der Algorithmus von KRÜHN und der Algorithmus von KIM & TANCHOCO herauszustellen.

Tabelle 2.4: Ermöglichung der Wandlungsfähigkeit durch die einzelnen Algorithmen

Eigenschaft	Algorithmus				
	MAYER	KRÜHN	SEIBOLD	KIM & TANCHOCO	GUE ET AL.
Fördersystem?	Großskalige Module	Kleinskalige Module	Großskalige Module	FTF	Großskalige Module
Bidirektionaler Transport möglich?	Ja	Ja	Ja	Ja	Eingeschränkt
Dezentral gesteuert?	Ja	Ja	Ja	Nein	Ja
Beliebige Layouts möglich?	Ja	Ja	Ja	Ja	Nein

Für die Besonderheit des Algorithmus von KRÜHN gibt es zwei Gründe. Der erste Grund ist, dass dieser für kleinskalige Fördermodule entwickelt wurde. Kleinskalige Fördermodule haben im Vergleich zu den großskaligen Fördermodulen und zu den FTF den Vorteil, dass sie alle logistischen Funktionen, die von VENTZ definiert wurden, erfüllen (siehe Kapitel 2.1.3 für die Definitionen der logistischen Funktionen). Somit besitzen die kleinskaligen Fördermodule zumindest auf funktionaler Ebene die höchste Wandlungsfähigkeit. Wenn also in einem Anwendungsszenario das Vorhandensein aller logistischen Funktionen vorteilhaft ist, haben die kleinskaligen Fördermodule einen Vorteil und somit auch KRÜHNS Algorithmus. Weiterhin ist es KRÜHN gelungen durch die Einführung der Modulnachbarschaft einen Algorithmus, der zunächst für großskalige Fördermodule entwickelt wurde, an kleinskalige Fördermodule anzupassen. Es ist zu überprüfen, ob auch weitere Algorithmen, die für großskalige Fördermodule entwickelt worden sind, durch das Konzept der Modulnachbarschaft für kleinskalige Fördermodule angepasst werden können.

Der Algorithmus von KIM & TANCHOCO ist herauszustellen, da er als einziger nicht für ein modulares Fördersystem entwickelt wurde. Die modularen Fördersysteme, die im Rahmen dieser Arbeit betrachtet worden sind, werden immer dezentral gesteuert. Im Gegensatz hierzu ist der Algorithmus von KIM & TANCHOCO für ein zentral gesteuertes System entwickelt worden. Die Autoren haben hierzu selbst folgende Beurteilung verfasst: "Die parallele Ausführung des Routingalgorithmus kann in bidirektionalen Systemen zu widersprüchlichen Ablaufplänen oder zu Stillständen des Systems führen". (Engl. „Parallel execution of the routing algorithm may result in conflicting travel

schedules or gridlocks in a bidirectional network."). Es ist somit zu prüfen, ob der Algorithmus von KIM & TANCHOCO angepasst werden kann, um dezentral gesteuerte, modulare Fördersysteme anzusteuern.

2.5 Fazit

In dem Kapitel 2.1 wurden drei Kategorien von Anforderungen an Fördersysteme beschrieben: Anforderungen, die erfüllt werden müssen, um den Betrieb sicherzustellen, Anforderungen, die erfüllt werden müssen, um die Leistungskenngrößen zu erreichen und Anforderungen, die erfüllt werden müssen, damit das Fördersystem wandlungsfähig ist.

Die Sicherstellung des Betriebes erfolgt, indem das Auftreten von Konflikten verhindert wird, zu denen beispielsweise Verklemmungen zählen. Bei dem Auftreten eines Konfliktes kommt es zu einem vollständigen oder teilweisen Erliegen des Transportes von Transporteinheiten. Weiterhin müssen die Fördersysteme Anforderungen an die Leistungskenngrößen erfüllen, damit ein ökonomisch sinnvoller Betrieb erreicht wird. Die wichtigste Leistungskenngröße ist hierbei der Durchsatz an Transporteinheiten.

Damit ein Fördersystem als wandlungsfähig bezeichnet werden kann, muss es verschiedene Anforderungen erfüllen. Eine wichtige Anforderung ist hierbei, ob eine Layoutänderung schnell (also innerhalb von Minuten) durchgeführt werden kann. Schnelle Layoutänderungen können nur durchgeführt werden, wenn sowohl die Hardware als auch die Routingalgorithmen dies unterstützen.

Drei wandlungsfähige Fördersysteme wurden in den vorangegangenen Kapiteln näher betrachtet: großskalige Fördermodule, kleinskalige Fördermodule und fahrerlose Transportfahrzeuge. Sowohl großskalige Fördermodule als auch kleinskalige Fördermodule werden zu einem Layout zusammengefügt und anschließend miteinander vernetzt. Während des Betriebes bleiben die Module stationär. Auf der Oberseite der Module befinden sich Antriebe, mit denen Transporteinheiten von einem Modul zu einem benachbarten Modul bewegt werden. Für jede Transporteinheit, die in das System eingeschleust wird, werden durch einen Routingalgorithmus Routen reserviert. Bei der Routenreservierung müssen Konflikte verhindert und möglichst kurze Routen gefunden werden, um die Anforderungen an die Leistungskenngrößen zu erfüllen. Die Routingalgorithmen werden dezentral ausgeführt, da jedes Modul über eine eigene Steuerung verfügt.

Im Gegensatz zu den modularen Fördersystemen bleiben fahrerlose Transportfahrzeuge nicht stationär, sondern bewegen sich zusammen mit der aufgenommenen

Transporteinheit entweder frei in einem zugewiesenen Bereich oder auf vorgegebenen Bahnen. Auch hier müssen Routingalgorithmen eingesetzt werden. Die Routingalgorithmen werden in der Regel zentral ausgeführt.

Für fahrerlose Transportfahrzeuge gibt es zeitfensterbasierte Routingalgorithmen, die in der Lage sind, in jedem Layout immer die kürzesten Routen zu finden. Keiner der untersuchten Routingalgorithmen für die dezentral gesteuerten, modularen Fördersysteme ist hierzu in der Lage.

Kapitel 3

Zielsetzung und Vorgehensweise

Das Ziel dieser Arbeit besteht darin, einen zeitenfensterbasierten Routingalgorithmus für dezentral gesteuerte, modulare Fördersysteme zu entwickeln. Dieser zeitenfensterbasierte Algorithmus muss in der Lage sein, sowohl großskalige als auch kleinskalige Fördermodule anzusteuern. Die Leistungsfähigkeit des zu entwickelnden Algorithmus muss abschließend mit der Leistungsfähigkeit von bereits existierenden Algorithmen verglichen werden. Zur Erreichung dieses Hauptziels werden fünf Teilziele definiert, die im Folgenden näher beschrieben werden. Die Bearbeitungsreihenfolge der Teilziele ist in der Abbildung 3.1 graphisch dargestellt.

Das erste Teilziel besteht darin, das Referenzfördersystem zu beschreiben. Hierbei wurde als Referenzfördersystem die netkoPs-Fördermatrix ausgewählt, da es sich um ein modernes, kleinskaliges Fördersystem handelt, welches am ITA entwickelt wurde. Die Notwendigkeit für die Beschreibung des Referenzfördersystems ergibt sich aus den folgenden zwei Gründen: Der erste Grund besteht darin, dass ein zeitenfensterbasierter Routingalgorithmus nur implementiert werden kann, wenn es möglich ist, die Uhren der einzelnen Förderer ausreichend genau zu synchronisieren. Daher muss geprüft werden, ob die netkoPs-Fördermatrizen hierzu in der Lage sind. Der zweite Grund ist, dass von dem Referenzfördersystem ein Softwaremodell abgeleitet werden muss. Ohne eine vorherige genaue Beschreibung des Referenzfördersystems kann von diesem kein Softwaremodell erstellt werden.

Folglich besteht das zweite Teilziel darin, in einer Simulationsumgebung ein Softwaremodell von den netkoPs-Fördermatrizen zu erstellen. Die Erstellung des Softwaremodells liegt darin begründet, dass am ITA nur eine begrenzte Anzahl von Fördermatrizen vorhanden ist. Es ist jedoch von wissenschaftlichem Interesse, das Systemverhalten und die Leistungsfähigkeit eines Fördersystems zu analysieren, welches aus einer gro-

Ben Anzahl an Fördermatrizen besteht. Durch die Erstellung einer Simulationsumgebung, in der beliebig viele Software-Modelle von den Fördermatrizen simuliert werden können, wird dies ermöglicht. Weiterhin kann das Softwaremodell eingesetzt werden, um die Entwicklung des zeitfensterbasierten Algorithmus zu unterstützen. Erste Versionen des Algorithmus können im Softwaremodell schnell und risikofrei getestet werden, um etwaige Fehler schneller ausfindig machen zu können und somit die Entwicklung zu beschleunigen.

Das dritte Teilziel besteht anschließend darin, den zeitfensterbasierten Algorithmus zu entwickeln. Die Entwicklung gliedert sich in zwei Teilschritte. Im ersten Teilschritt muss der Algorithmus konzeptionell erarbeitet werden. Die konzeptionelle Entwicklung erfolgt, indem die im Stand der Wissenschaft und Technik dargestellten Routingalgorithmen weiterentwickelt werden. Im zweiten Teilschritt wird der konzeptionell erarbeitete Algorithmus implementiert, sodass innerhalb der Simulationsumgebung die Softwaremodelle der netkoPs-Fördermatrizen angesteuert werden können. Der zu entwickelnde Algorithmus muss hierbei sowohl für großskalige als auch kleinskalige Fördermodule umgesetzt werden. Weiterhin muss die Verhinderung von Konflikten bewiesen werden und eine Laufzeitanalyse des Algorithmus erfolgen.

Im vierten Teilziel erfolgt die Untersuchung des Systemverhaltens von netkoPs-Fördermatrizen, die mit dem zeitfensterbasierten Algorithmus angesteuert werden. Hierbei werden mit Hilfe der Simulationsumgebung verschiedene Layouts simuliert. Pro Layout werden die einstellbaren Parameter variiert und die Leistungskenngrößen gemessen. Die hierbei erzielten Werte werden mit den Werten von den bereits bestehenden Algorithmen verglichen. Somit wird es möglich, Regeln abzuleiten, in welchem Anwendungsszenario der entwickelte zeitfensterbasierte Algorithmus idealerweise eingesetzt werden kann.

Das fünfte und letzte Teilziel besteht darin, Weiterentwicklungsempfehlungen zu formulieren. Diese werden in zwei Bereiche unterteilt: Zum einen werden Empfehlungen gegeben, wie das Referenzfördersystem weiterentwickelt werden kann und zum anderen werden die notwendigen Schritte beschrieben, sodass das Referenzfördersystem in ein Produktionsumfeld integriert werden kann.

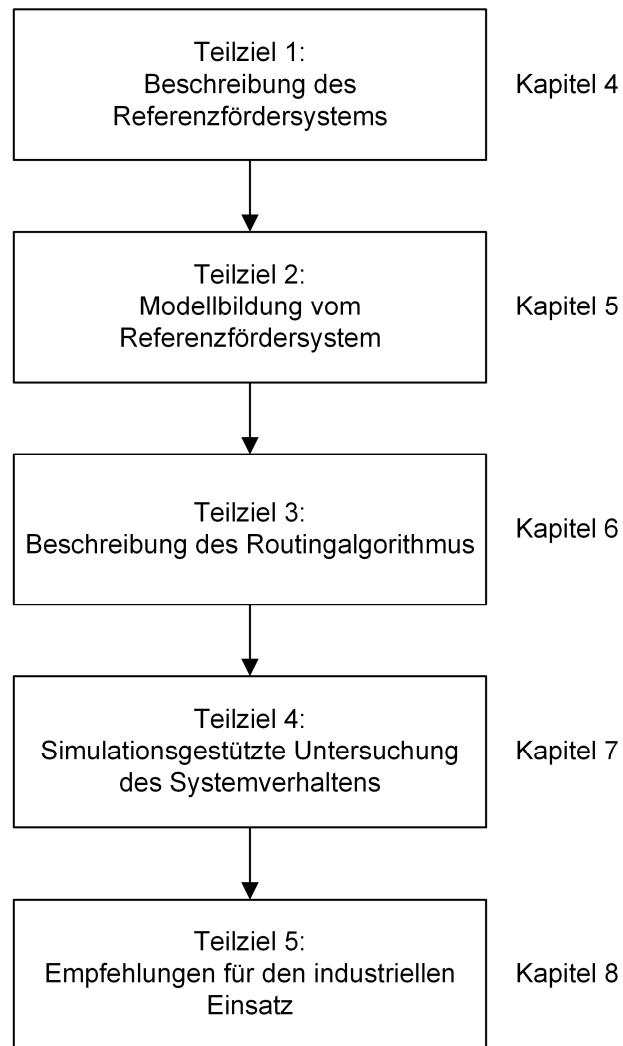


Abbildung 3.1: Bearbeitungsreihenfolge der Teilziele

Kapitel 4

Beschreibung des Referenzfördersystems

4.1 Hardware

In diesem Kapitel erfolgt die Darstellung des Referenzfördersystems, welches im Rahmen des netkoPs-Forschungsvorhabens entstand und in den folgenden Veröffentlichungen beschrieben ist: [Ove17a, Ove17b, Soh16a, Soh16b, Sti16]. In dem Forschungsvorhaben wurde die Hardware und die Steuerungen für zwei Fördertypen entwickelt: Fördermatrizen und Bandförderer. Da jeder Förderer über eine eigene Steuerung verfügt, ist das Gesamtsystem dezentral gesteuert. Auf Grund des dezentralen Steuerungsansatzes ist es möglich, dass das Layout des Fördersystems nach dem „Plug & Play“ Ansatz verändert werden kann: Eine Änderung des physischen Layouts wird durch die Steuerung erkannt und kann ohne Eingreifen eines Menschen genutzt werden. Ein mögliches Beispiellayout ist in der Abbildung 4.1 dargestellt.

Im Rahmen des netkoPs-Forschungsvorhabens wurden weiterhin für die Identifikation bzw. die Bearbeitung von Transporteinheiten Bearbeitungsmaschinen bzw. Kamerasysteme entwickelt. Für die Entwicklung eines zeitenfensterbasierten Routingalgorithmus zur Ansteuerung von Fördermatrizen ist die detaillierte Darstellung der Bandförderer, der Bearbeitungsmaschinen und der Kamerasysteme nicht erforderlich. Es wird daher an dieser Stelle auf das entsprechende Kapitel im Abschlussbericht des netkoPs-Forschungsvorhabens verwiesen [Soh17d] und auf die folgenden Veröffentlichungen: [Soh17a, Soh20b].

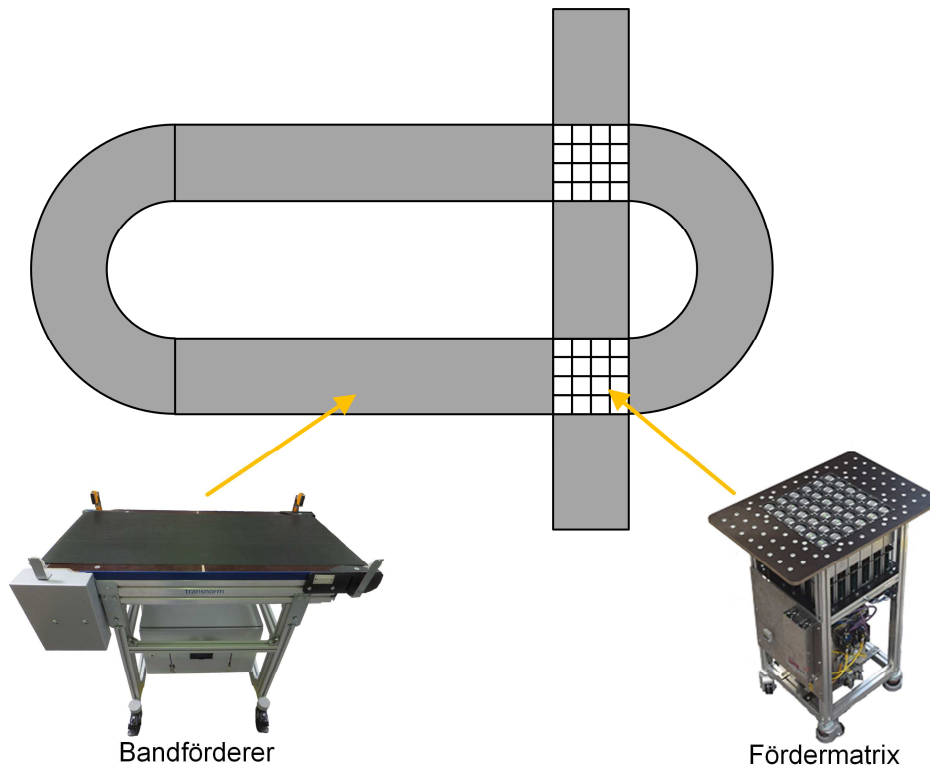


Abbildung 4.1: Beispiellayout bestehend aus Fördermatrizen und Bandförderern

Ein einzelnes netkoPs-Fördermodul (siehe Abbildung 4.2) verfügt über zwei außenlaufende Synchronmotoren. Jeder dieser Motoren wird über einen eigenen dedizierten Frequenzumrichter angesteuert. Die Motoren werden dazu genutzt, um eine Rolle zu schwenken und anzutreiben. Die Rolle kann um 360° geschwenkt werden. Beide Motoren sind baugleich und verfügen somit über die gleichen Motorkennwerte. Weiterhin verfügt jedes Modul über Lichttaster, mit dem der Belegungszustand des Fördermoduls erfasst wird. Die relevanten technischen Daten der Fördermodule sind im Anhang A1 zusammengefasst.

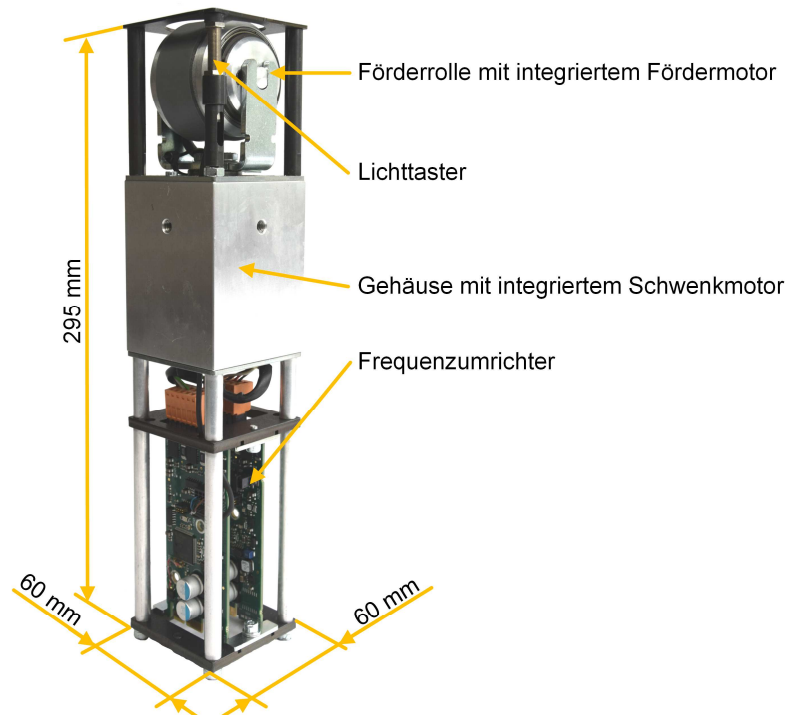


Abbildung 4.2: *netkoPs-Fördermodul*

Damit mit den einzelnen Fördermodulen Transportgüter bewegt werden können, müssen diese zu einer Fördermatrix kombiniert werden (siehe Abbildung 4.3). Der realisierte Prototyp verfügt über 36 Fördermodule und einen integrierten Schaltschrank, in dem sowohl die Leistungselektronik als auch die Steuerung für die Fördermatrix untergebracht ist.

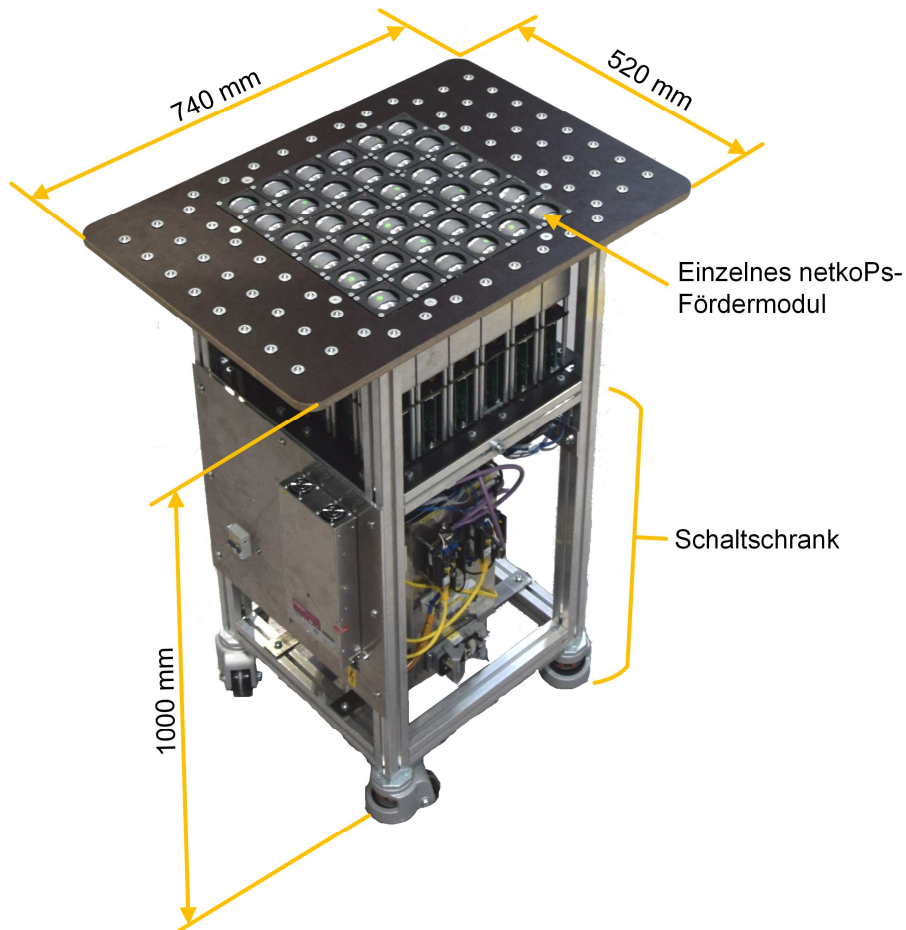


Abbildung 4.3: netkoPs-Fördermatrix

Mit der netkoPs-Fördermatrix können 8 kg schwere biegesteife Transporteinheiten mit den Abmessungen von 400 x 400 x 300 mm bewegt werden, die über eine homogene Massenverteilung verfügen. Die Matrix ist in der Lage, die Transporteinheit innerhalb von $\sim 0,25$ s auf eine Geschwindigkeit von 0,86 m/s zu beschleunigen. Dies entspricht einer Beschleunigung von $0,9 \text{ m/s}^2$. [Soh17c]

Der Reibungskoeffizient zwischen den eingesetzten Transporteinheiten und den Förderrollen beträgt $\mu \approx 0,34$. Bei einem industriellen Einsatz der Fördermatrix ist davon auszugehen, dass sowohl der Reibungskoeffizient als auch die Massenverteilung der Transporteinheiten inhomogen sein kann. Hierdurch ergeben sich Fehlerursachen. Die Fehlerursachen und ihre möglichen Kompensationsmechanismen sind im Anhang A2 dargestellt.

4.2 Steuerung

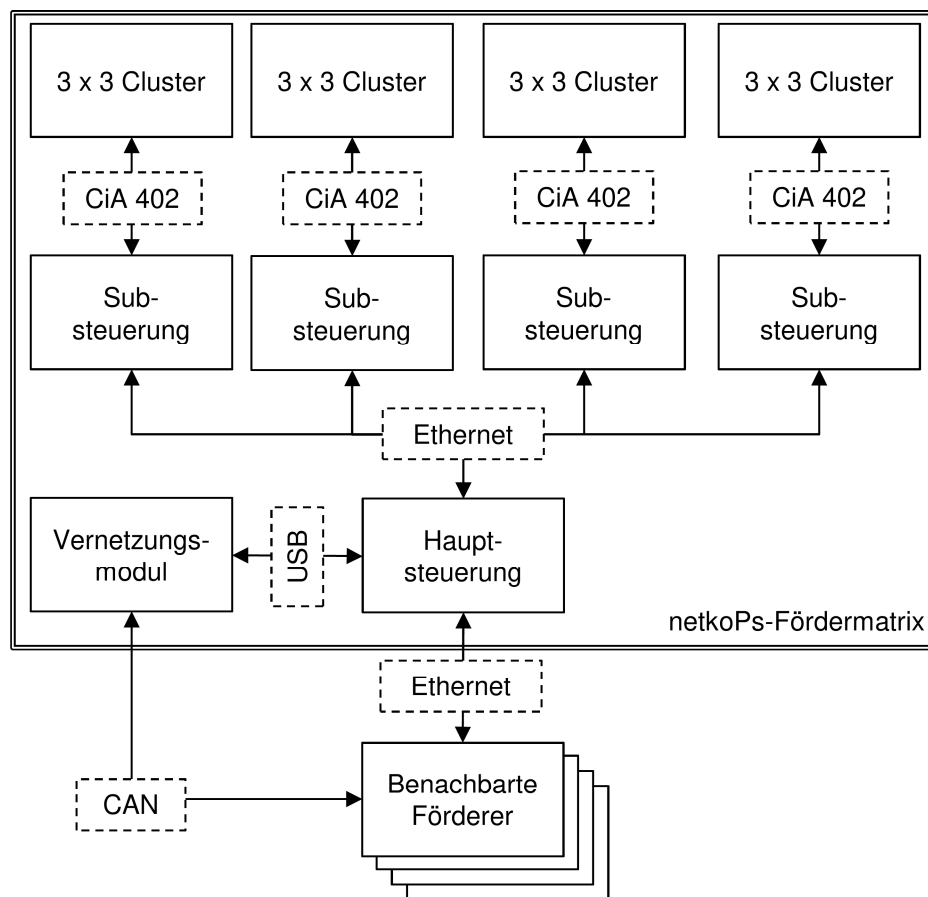


Abbildung 4.4: Aufbau der Beagle Bone basierten Steuerung der Fördermatrix

Für die netkoPs-Fördermatrix wurden zwei Steuerungen entwickelt. Zunächst wird die Steuerung beschrieben, die während der Laufzeit des Forschungsvorhabens entwickelt wurde. Der Aufbau dieser Steuerung ist in der Abbildung 4.4 dargestellt. Die einzelnen Fördermodule werden zu Clustern zusammengefasst, welche aus jeweils 9 Fördermodulen bestehen. Jedes der Fördermodule kann unabhängig voneinander bewegt werden. Weiterhin wird jeder einzelne Cluster mit einer Substeuerung über die CiA 402 Schnittstelle angesteuert [CAN20]. Bei den Substeuerungen handelt es sich um den Beagle Bone Black rev c. Die Aufgabe der Substeuerungen besteht darin, die Befehle der Hauptsteuerung, die über eine Ethernet-Schnittstelle verschickt werden, in die entsprechenden CAN-Befehle für die Fördermodule zu übersetzen. Für die Hauptsteuerung wurde ebenfalls ein Beagle Bone Black rev c verwendet. Die Hauptsteuerung wird wiederum über Ethernet mit vier benachbarten Förderern verbunden, um Daten auszutauschen. Zusätzlich ist die Hauptsteuerung über USB mit einem Vernetzungsmodul verbunden, das im Rahmen des Forschungsvorhabens von der Firma Gigatronik entwickelt wurde. Die Aufgabe des Vernetzungsmoduls besteht darin zu

detektieren, auf welcher Seite der Fördermatrix diese mit benachbarten Förderern verbunden ist. Auf der Steuerung wurde ein Routingalgorithmus implementiert, der sich an dem von KRÜHN in [Krü15] beschriebenen orientiert (siehe [Soh17b]).

Damit der Transportvorgang fehlerfrei durchgeführt werden kann, ist es notwendig, dass alle Fördermodule synchron beginnen. Weiterhin ist es notwendig, dass die Fördermodule synchron mit den angrenzenden Förderern starten. Das synchrone Starten von Antrieben geschieht über die gleichen Verfahren mit denen in Netzwerken Uhren synchronisiert werden. Die Synchronisierung von Uhren ist ein eigenes Forschungsfeld. Eine Übersicht über das Forschungsfeld liefert: [Tan16]. Verschiedene günstige Lösungen für die Synchronisierung von Uhren existieren bereits. Im Rahmen des netkoPs-Forschungsvorhabens wurde hierbei das Precision Time Protocol (PTP) ausgewählt, welches in dem IEEE 1588-2008 Standard beschrieben ist [IEE08]. Das PTP Protokoll benötigt keine besonderen Ethernet-Chips, um eingesetzt zu werden, jedoch kann die Genauigkeit erhöht werden, wenn PTP-fähige Hardware vorhanden ist. Falls PTP-fähige Hardware vorhanden ist, ist die Genauigkeit von PTP ähnlich hoch wie die von GPS. Der verwendete Beagle Bone verfügt über die entsprechende PTP-fähige Hardware [TI19]. PTP wird dann eingesetzt, wenn GPS nicht eingesetzt werden kann, aus Kostengründen oder an Orten, in denen das GPS-Signal nicht zuverlässig empfangen werden kann [Eid06]. Beides trifft auf Produktionsmaschinen zu. Die Genauigkeit von PTP liegt im Sub-Mikrosekundenbereich und somit ist diese ausreichend, um die Antriebe des Referenzfördersystems zu synchronisieren. Die maximale Abweichung der Uhren der einzelnen Steuerungen im netkoPs-System beträgt 100 nS [Ove17b].

Nachteilig an der Beagle Bone basierten Steuerung ist, dass die zeitliche Synchronisierung mit Hilfe von PTP nicht zuverlässig funktioniert hat. Der Fehlerfall bestand dabei darin, dass sich beim Einschalten der einzelnen Steuerungen in ca. 10% der Fälle die einzelnen Uhren der Steuerungen überhaupt nicht synchronisiert haben. Falls die Synchronisierung nach dem Einschalten erfolgreich durchgeführt werden konnte, blieben die Uhren während des Betriebes synchron. Die Fehlerursache liegt hierbei entweder in dem verwendeten Treiber der Hardware [TI19] oder in der verwendeten Implementierung des PTP-Protokolls [TLP20]. Das Finden und Beheben des Fehlers wurde hierbei als zeit- und kostenaufwendiger eingeschätzt, als der Austausch der verwendeten Hardware. Daher wurde eine zweite Steuerung realisiert.

Diese zweite Steuerung verwendet als CPU Modul eine Beckhoff CX 2030 [Bec20]. Dieses CPU Modul wurde mit fünf Erweiterungsklemmen ausgestattet: Vier CAN-Erweiterungsklemmen EL1859 und einer GPIO-Erweiterungsklemme. Mit den vier CAN-

Erweiterungsklemmen werden hierbei jeweils wieder die einzelnen Cluster angesteuert. Die GPIO-Klemme wird genutzt, um ein Bedienpult zu realisieren. Die Synchronisierung der einzelnen Klemmen und des CPU-Moduls erfolgt mit dem EtherCAT Protokoll, welches in der IEC 61158-3/4/5/6-1 genormt ist [IEC14]. Mit EtherCAT sind Zykluszeiten von $\leq 100 \mu\text{s}$ erreichbar. Da das CPU-Modul mit einer USB-Schnittstelle und Ethernet ausgestattet ist, kann es modifiziert werden, sodass statt eines Bedienpultes eine Verbindung mit angrenzenden Förderern möglich ist und somit der komplette Funktionsumfang der Beagle Bone basierten Steuerung erreicht wird. Der Aufbau der Beckhoff basierten Steuerung ist in der Abbildung 4.5 dargestellt.

Sowohl bei der Beagle Bone basierten Steuerung als auch bei der Beckhoff basierten Steuerung ist die Synchronisierungsgenauigkeit ausreichend, um die Antriebe synchron anzusteuern.

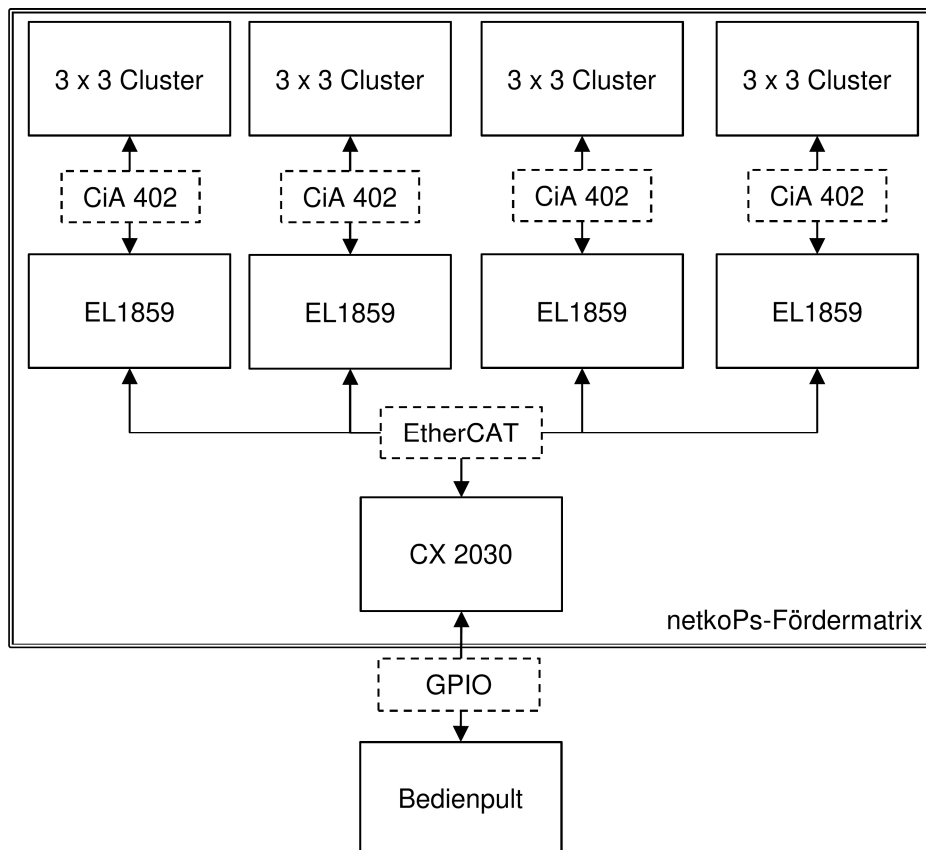


Abbildung 4.5: Aufbau der Beckhoff basierten Steuerung der Fördermatrix

Kapitel 5

Modellbildung vom Referenzfördersystem

5.1 Gründe für die Modellbildung

Da im Rahmen des netkoPs-Forschungsprojektes zwei Hardware-Prototypen der Fördermatrizen entwickelt wurden, könnte die Entwicklung des Algorithmus zur Vorhersage von Transportzeiten auch direkt auf diesen Hardware-Prototypen erfolgen. Gegen die Entwicklung des Algorithmus direkt auf den Hardware-Prototypen sprechen jedoch zwei gewichtige Gründe: Erstens ist es von wissenschaftlichem Interesse, das Systemverhalten eines Fördersystems zu analysieren, welches aus mehr als lediglich zwei Fördermatrizen besteht. Durch die Erstellung einer Simulationsumgebung, in der beliebig viele Software-Modelle von den Fördermatrizen simuliert werden können, ist es möglich, das Systemverhalten von größeren Fördersystemen zu untersuchen.

Zweitens ist die Entwicklung des Algorithmus zur Vorhersage von Transportzeiten auf Hardware-Prototypen wesentlich zeitaufwendiger als innerhalb einer Simulationsumgebung. Der höhere Zeitaufwand resultiert hauptsächlich aus der dezentralen Steuerungsarchitektur: Sowohl das Verteilen (Engl.: Deployment) von neuen Softwareversionen als auch die Fehlersuche (Engl.: Debugging) muss über das Netzwerk erfolgen. Im Gegensatz zu dem Verteilen und der Fehlersuche innerhalb einer Simulationsumgebung muss das Verteilen und die Fehlerursache zunächst netzwerkfähig gemacht werden. Dies erhöht die Komplexität der Software, wodurch wiederum mehr Möglichkeiten für Fehler entstehen. Besonders die Fehlersuche ist kritisch, da es zur Eingrenzung von Fehlern vorteilhaft ist, das gesamte System gleichzeitig anzuhalten. Dies ist auf Grund der dezentralen Steuerungsarchitektur nur mit erheblichem Mehraufwand zu realisieren.

Wegen diesen beiden Gründen erfolgt die Modellierung der Fördermatrizen als Softwaremodell in einer Simulationsumgebung. Neben den Fördermatrizen müssen darüber hinaus noch die Quellen und Senken sowie die Transporteinheiten simuliert werden. Jedes Modell ist immer eine Vereinfachung des modellierten realen Objektes.

In vorangegangenen wissenschaftlichen Arbeiten, in denen Algorithmen zur Ansteuerung von modularen Fördersystemen erforscht wurden, wurden ebenfalls Simulationsumgebungen mit Modellen von Transportgütern erstellt (vgl. [Gue12, Krü15, May09, Sei16a, Sei16b]). Die in den nachfolgenden Kapiteln beschriebenen Modelle sind angelehnt an die Modelle von KRÜHN und SEIBOLD, die in [Krü15] und [Sei16b] beschrieben wurden. In diesem Kapitel werden die bei der Modellierung getroffenen Vereinfachungen explizit genannt und es wird erläutert, welche Erweiterungen des Steuerungsalgorithmus erfolgen müssen, wenn anstelle eines Modells ein reales Transportgut bewegt werden soll.

5.2 Modell von den Transporteinheiten

Die erste Vereinfachung bei der Modellierung der Transporteinheiten besteht darin, dass die Transporteinheiten zweidimensional modelliert werden, da die Höhe für die Entwicklung des Routingalgorithmus nicht relevant ist.

Bei der Ansteuerung von kleinskaligen Fördermodulen muss die Größe und Form der Transporteinheiten berücksichtigt werden, da immer eine Nachbarschaft aus mehreren Modulen für den Transport benötigt wird. Die von KRÜHN in [Krü15] entwickelte Methode zur Bildung von Modulnachbarschaften ist in der Lage, für beliebige große rechteckige Transporteinheiten Nachbarschaften zu bilden. Für Transporteinheiten, die eine komplexere Grundfläche aufweisen, bildet KRÜHN rechteckige Hüllflächen.

Um den Implementierungsaufwand zu reduzieren, werden in dieser Arbeit nur quadratische Transporteinheiten mit einer einheitlichen Größe modelliert, da die Auswirkungen von unterschiedlich großen Transporteinheiten von KRÜHN bereits hinreichend wissenschaftlich untersucht worden sind.

Die Größe der simulierten Transporteinheiten orientiert sich an der Größe der simulierten großskaligen Fördermodule, die den FlexConveyor-Modulen nachempfunden sind. Die Abmessungen der FlexConveyor-Module sind 500 x 500 mm und somit werden die Transporteinheiten mit den Abmessungen 480 x 480 mm modelliert.

Des Weiteren werden die Transporteinheiten mit einer homogenen Massenverteilung modelliert. Somit fällt der Massenschwerpunkt mit der geometrischen Mitte der modellierten Transporteinheiten zusammen. Dies ist bei realen Transportgütern nicht

zwangsläufig gegeben. Die maximale übertragbare Kraft von allen am Transport beteiligten Fördermodulen auf das Transportgut F_{max} ist folgendermaßen definiert:

$$F_{max} = \mu \cdot m \cdot g \quad (5.1)$$

μ ist hierbei der Reibungskoeffizient zwischen Fördermodulen und Transporteinheit, m ist die Masse der Transporteinheit und g ist die Fallbeschleunigung der Erde. Im Falle einer homogenen Massenverteilung können alle am Transportprozess beteiligten Module die gleiche maximale Kraft übertragen. Bei einer inhomogenen Massenverteilung kann es dazu kommen, dass einige Module weniger Kraft übertragen können. Falls die von der Steuerung vorgegebene Kraft die maximal übertragbare Kraft übersteigt, tritt Schlupf auf. Wenn dieser Schlupf ungleichmäßig auftritt, wird ein Drehmoment erzeugt. Durch dieses Drehmoment kommt es zu einer Verdrehung der Transporteinheit. Diese Verdrehung kann dazu führen, dass das Transportgut sich verkantet und somit Teile des Fördersystems unpassierbar werden.

Genau wie die inhomogene Massenverteilung kann der Reibungskoeffizient zwischen Fördermodulen und Transporteinheit uneinheitlich sein. Diese Uneinheitlichkeit kann durch abgenutzte Laufsichten einzelner Module oder Klebeband auf den Transporteinheiten entstehen, welches einen niedrigeren Reibungskoeffizienten besitzt, als z.B. Pappe. Der hierbei auftretende Effekt ist der gleiche wie bei der inhomogenen Massenverteilung: Es kann zu unbeabsichtigten Verdrehungen der Transporteinheit kommen. Im Anhang A2 sind technische Lösungen mit ihren Vor- und Nachteilen dargestellt, um das Problem der unbeabsichtigten Verdrehung zu lösen.

Die Trägheit der Transporteinheiten wird nicht modelliert. Somit gibt es keine Beschleunigungs- oder Abbremsphasen. Wenn ein Transportvorgang beginnt, erreichen die Transportgüter sofort ihre Endgeschwindigkeit. Diese Vereinfachung wurde aus den folgenden Gründen getroffen: Auf dem Referenzfördersystem (siehe Kapitel 4) trat beim Transport kein einziges Mal Schlupf auf. Die explizite Modellierung der Beschleunigungs- und Abbremsphase ist somit nicht notwendig bei Fördergeschwindigkeiten bis 2 m/s. Bei höheren Endgeschwindigkeiten kann die Zeit für die Beschleunigung und das Abbremsen jedoch nicht vernachlässigt werden. Da die Dauer der Beschleunigungs- und Abbremsphase vor dem Beginn des Transportbeginns berechnet werden kann, können diese bei der Routenreservierung bereits eingeplant werden. Der Algorithmus, der im Rahmen dieser Arbeit entwickelt wird, berücksichtigt diese Phasen konzeptionell.

5.3 Modell von den Fördermodulen

Im Rahmen dieser Arbeit werden sowohl die kleinskaligen als auch die großskaligen Fördermodule modelliert. Die modellierten großskaligen Fördermodule werden den FlexConveyor-Modulen nachempfunden (siehe Kapitel 2.2.1): Sie haben die Abmessungen 500 x 500 mm und können nur mit direkt angrenzenden Fördermodulen kommunizieren. An jeder Kante der Fördermodule befindet sich ein Sensor, der dazu genutzt wird, Transporteinheiten zu detektieren, wenn diese das Fördermodul erreichen oder verlassen. Jedes großskalige Fördermodul verfügt über genau eine Steuerung.

Die modellierten kleinskaligen Fördermodule werden den netkoPs-Modulen nachempfunden (siehe Kapitel 4.2). Im Gegensatz zu den großskaligen Fördermodulen, gibt es zwei Möglichkeiten die Ansteuerung der Module zu modellieren. Bei der ersten Möglichkeit entspricht die Ansteuerung der tatsächlich realisierten Ansteuerung des netkoPs-Referenzfördersystems – sie wird im Folgenden als matrixbasierte Ansteuerung bezeichnet. Im Gegensatz zu der matrixbasierten Steuerung kann jedes Modul auch mit einer eigenen Steuerung modelliert werden – sie wird im Folgenden als modulbasierte Ansteuerung bezeichnet.

Die Unterscheidung zwischen der matrixbasierten und der modulbasierten Ansteuerung geschieht über den Begriff der Modulnachbarschaft, welcher in [Krü15] definiert wurde. Alle Module, die mit dem Ausführen der gleichen Aktion beschäftigt sind, also z.B. der Translation einer Transporteinheit, gehören zu der gleichen Nachbarschaft. Bei der matrixbasierten Ansteuerung sind die Nachbarschaften statisch, d.h. die Anzahl der Module in einer Nachbarschaft und somit auch die Größe der Nachbarschaft ändert sich nicht. Bei der modulbasierten Ansteuerung sind die Nachbarschaften dynamisch, d.h. Module sind nicht fest einer Nachbarschaft zugeordnet. Stattdessen werden immer wieder neue Nachbarschaften erstellt und wieder aufgelöst. Die Größe der Nachbarschaft ist hierbei abhängig von der auszuführenden Aktion.

Der Vorteil von matrixbasierten Ansteuerungen besteht darin, dass der Implementierungsaufwand geringer ist und der Reservierungsprozess vereinfacht wird, da Module immer genau einer Nachbarschaft zugeordnet sind. Im Gegensatz dazu besteht der Vorteil der modulbasierten Ansteuerung darin, dass die einzelnen Module effektiver genutzt werden, da die Größe der Nachbarschaften dynamisch für die Aufgabe gewählt wird und somit nicht mehr Module als unbedingt notwendig zu einer Nachbarschaft zusammengefasst werden.

Da in dem Referenzfördersystem eine matrixbasierte Steuerung umgesetzt wurde, wird auch das Modell der kleinskaligen Fördermodule mit einer matrixbasierten Ansteuerung realisiert. Die Untersuchung von kleinskaligen Modulen mit einer modulbasierten Ansteuerung ist jedoch wissenschaftlich interessant und wurde z.B. auch schon von KRÜHN in [Krü15] durchgeführt. Aus diesem Grund wird in dem Kapitel 6.2 ein Routingalgorithmus für eine modulbasierte Ansteuerung von kleinskaligen Fördermodulen beschrieben.

Entsprechend dem netkoPs-Referenzfördersystem wird die Anzahl der Module in der Nachbarschaft auf 36 festgelegt. Diese 36 Module werden von einer Hauptsteuerung kontrolliert. Wenn diese Hauptsteuerung eine Routinganfrage von einem benachbarten Förderer erhält, wird diese verarbeitet und anschließend, falls die Routinganfrage angenommen wurde, an benachbarte Förderer weiterleitet. Bei der Annahme einer Routinganfrage werden weiterhin entsprechende Befehle an die Fördermodule gesendet. Ein Fördersystem, welches aus mehreren Fördermatrizen mit einer matrixbasierten Ansteuerung zusammengesetzt wird, ist dezentral gesteuert, da die einzelnen Hauptsteuerungen unabhängig voneinander sind und somit dezentral Routinganfragen verarbeiten. Jedoch erfolgt die Ansteuerung der Module innerhalb einer Matrix zentral durch die entsprechende Hauptsteuerung.

Die modellierten Fördermatrizen orientieren sich an den Abmessungen der netkoPs-Fördermatrix. Ein einzelnes Modul hat die Abmessungen 60 x 60 mm. Das Spaltmaß zwischen den einzelnen Modulen beträgt ~28 mm. Somit ergibt sich eine Fläche von ~500 x 500 mm bei 36 Modulen. Dies entspricht den Abmessungen der FlexConveyor-Modulen. Genau wie die großskaligen Fördermodule können die Fördermatrizen mit Hilfe von Belegungssensoren Transporteinheiten detektieren. Die netkoPs-Module sind in der Lage, sich beliebig auszurichten. Hierdurch ist es möglich, dass Transporteinheiten sowohl rotiert als auch translatorisch bewegt werden können. Die modellierten kleinskaligen Fördermodule können jedoch Transporteinheiten nur translatorisch bewegen. Diese Vereinfachung wurde getroffen, da die Rotation nur notwendig ist, um rechteckige Transporteinheiten so auszurichten, dass diese durch eine Engstelle hindurch transportiert werden können. Da die Modelle der Transporteinheiten quadratisch sind, muss keine Rotation erfolgen und somit auch nicht die Möglichkeit der Rotation modelliert werden.

Da die Abmessungen der modellierten Fördermatrizen und der modellierten großskaligen Fördermodule identisch sind und diese über ähnliche technische Funktionen ver-

fügen (Kommunikation nur mit den direkten vier Nachbarn, Detektion von Transporteinheiten, etc.) wird das Modell der großskaligen Fördermodule ohne Anpassungen auch als Modell für die Fördermatrizen genutzt.

Die Kommunikations- und Fördergeschwindigkeit der Module kann frei gewählt werden. Eine Vereinfachung besteht darin, dass die Uhren aller Module vollständig synchron sind – es gibt somit eine einheitliche Zeitbasis in dem gesamten Fördersystem. In einem realen Fördersystem sind die Zeitbasen niemals vollständig synchron. Jedoch sind die real auftretenden Differenzen so klein, dass der Förderprozess nicht beeinflusst wird.

Jedes Modul kann mit einer Quelle oder Senke assoziiert sein. Im Falle einer Quelle oder Senke werden die Transporteinheiten über die assoziierten Fördermodule in das Fördersystem ein- bzw. ausgeschleust.

Jede Senke besitzt eine eindeutige Ziel-ID. Wenn eine Transporteinheit die Senke erreicht, die als ihr Ziel festgelegt wurde, dann wird die Transporteinheit aus dem Fördersystem ausgeschleust. Senken werden mit einer unendlich großen Aufnahmekapazität modelliert. Dies bedeutet, dass Senken immer Transporteinheiten ausschleusen können.

Quellen werden mit einer Warteschlange mit einer unendlich großen Kapazität für Transporteinheiten modelliert. Die Modellierung der Warteschlange ist notwendig, um hierdurch verschiedene Kennwerte bestimmen zu können. Z.B. kann gemessen werden, wie lange Transporteinheiten warten müssen, bevor sie über das assoziierte Modul in das Fördersystem geschleust werden. Weiterhin kann die durchschnittliche Anzahl an Transporteinheiten innerhalb der Warteschlangen gemessen werden und es kann das Verhältnis zwischen den Transporteinheiten, die in Warteschlangen warten, zu den im Transportprozess befindlichen Transporteinheiten berechnet werden.

Eine Transporteinheit aus der Warteschlange wird auf dem assoziierten Modul in das Fördersystem eingeschleust, sobald das assoziierte Modul frei ist. Zusammen mit der Transporteinheit wird dem assoziierten Modul die eindeutige ID und das Ziel der Transporteinheit mitgeteilt. Aus den von der Quelle erreichbaren Senken wird eine Senke per Zufallsgenerator als Ziel für die Transporteinheit ausgewählt. Die Wahrscheinlichkeit, mit der eine neue Transporteinheit zu der Warteschlange einer Quelle hinzugefügt wird, kann individuell pro Quelle festgelegt werden. Das mit einer Quelle assoziierte Fördermodul beginnt erst dann mit dem Routenreservierungsprozess, wenn die Transporteinheit die Warteschlange verlässt und auf dem Fördermodul platziert wird. Quellen und Senken dürfen nur mit Fördermodulen assoziiert sein, die am Rand des Fördersystems angeordnet sind.

5.4 Integration der Modelle in eine Simulationsumgebung

Die Simulationsumgebung, die im Rahmen diese Dissertation implementiert wird, baut auf dem MASON Simulationskern in Version 17 auf [Luk05, Mas19]. Die im Rahmen dieser Arbeit implementierte Simulationsumgebung verfügt über die im Folgenden beschriebenen Eigenschaften: Es ist möglich Fördermodule, Transporteinheiten, Quellen und Senken zu simulieren. Zum Starten eines Simulationslaufes werden sowohl eine Layout- als auch eine Konfigurationsdatei benötigt. In der Layoutdatei ist die räumliche Anordnung aller Förderer und aller Quellen und Senken beschrieben. In der Konfigurationsdatei werden sowohl Konfigurationsparameter für die Simulationsumgebung gesetzt als auch Steuerungsparameter für die simulierten Fördermodule. Die Senken werden durch eine farbige Umrandung besonders hervorgehoben. Jede Farbe wird hierbei immer nur einmal vergeben.

Verschiedene Funktionen wurden in die Simulationsumgebung integriert, um das Finden von Fehlerursachen zu vereinfachen. So sind die Simulationsläufe beispielsweise deterministisch. D.h.: Wenn nacheinander die Simulationsumgebung mit derselben Layoutdatei und derselben Konfigurationsdatei gestartet wird, ergibt sich ein exakt gleicher Simulationsverlauf. Durch diesen deterministischen Simulationsverlauf treten Fehler immer zum gleichen Simulationszeitpunkt auf und sind somit reproduzierbar und hierdurch leichter zu beheben. Die Reproduzierbarkeit wird erreicht, indem ein deterministischer Zufallszahlengenerator verwendet wird, mit dem auf den Quellen festgelegt wird, wann eine neue Transporteinheit erzeugt und eingeschleust wird und welches Ziel diese Transporteinheit hat. Der deterministische Zufallszahlengenerator benutzt hierbei den MT19937-Algorithmus, der in [Mat98] beschrieben ist. Die verwendete Implementierung ist in [Mas19] zu finden.

Weiterhin wurde die Simulationsumgebung so aufgebaut, dass sie mit oder ohne grafischer Benutzeroberfläche gestartet werden kann. Wenn die Simulationsumgebung ohne graphische Benutzeroberfläche gestartet wird, erhöht sich ihre Ausführungsgeschwindigkeit (pro reale Zeiteinheit können mehr Simulationsschritte durchgeführt werden). Die Simulationsumgebung mit eingeschalteter graphischer Benutzeroberfläche ist in Abbildung 5.1 dargestellt.

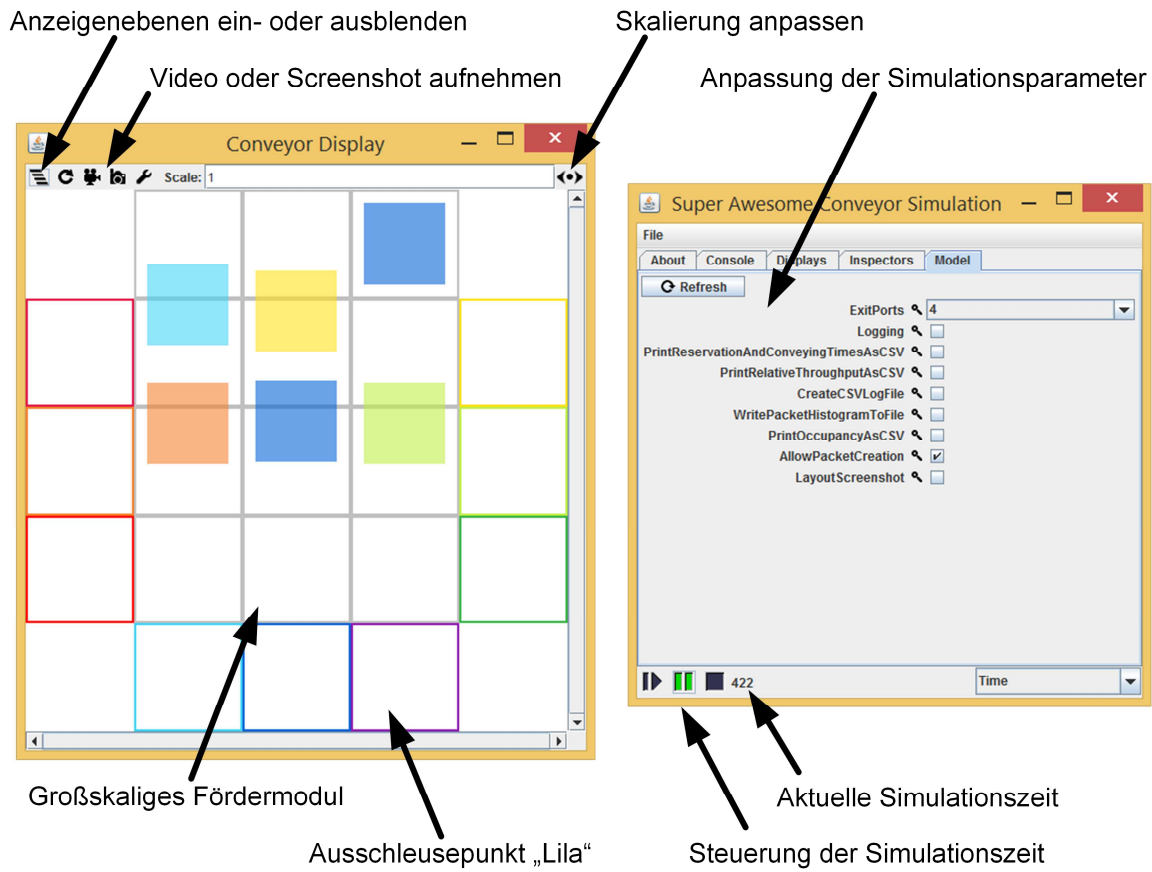


Abbildung 5.1: Screenshot der Simulationsumgebung

Kapitel 6

Beschreibung des Routingalgorithmus

6.1 Algorithmus für großskalige Fördermodule

6.1.1 Übersicht

Der hier beschriebene Algorithmus kann sowohl zur Ansteuerung von großskaligen Fördermodulen als auch zur Ansteuerung von kleinskaligen Fördermodulen mit einer matrixbasierten Ansteuerung verwendet werden (siehe Kapitel 5.3). Eine erste Version dieses Algorithmus wurde vom Autor dieser Arbeit bereits in den folgenden wissenschaftlichen Publikationen beschrieben: [Soh19, Soh20a].

Um die Lesbarkeit des Kapitels 6.1 zu erhöhen, werden konzeptionelle Designentscheidungen, die bei der Erstellung des Algorithmus getroffen wurden, erst nach der Beschreibung des Algorithmus im Kapitel 6.1.5 dargestellt. Der Algorithmus muss in der Lage sein, konfliktfreie Routen in bidirektional benutzbaren Fördersystemen zu reservieren. Hierbei muss dieser Algorithmus skalieren, um auch in Fördersystemen, die aus einer großen Anzahl von Fördermodulen bestehen, performant zu funktionieren.

Bevor eine Routenreservierung durchgeführt werden kann, müssen die Module zunächst die Initialisierungsphase durchlaufen. In dieser Phase erstellen die einzelnen Module Routingtabellen, indem Nachrichten mit ihren direkten Nachbarn ausgetauscht werden. Hierbei wird das gleiche Verfahren verwendet, welches MAYER in [May09] angewandt hat und das wiederum auf dem „Distance-Vector Protocol“ basiert, welches in Ethernet-Netzwerken eingesetzt wird und in [Tan11] beschrieben ist. Nach der Erstellung der Routingtabelle verfügen alle Module über die Information, welche Senken über sich erreicht werden können, wie weit die Senken entfernt sind und in welcher Richtung der kürzeste Weg zur Senke liegt.

Die Reservierung von Routen erfolgt, indem Nachrichten zwischen den einzelnen Fördermodulen ausgetauscht werden: Zunächst werden Routen von der Quelle zur Senke angefragt und anschließend in die Gegenrichtung bestätigt. Während der Anfragephase werden die Route und die Zeitfenster ausgehandelt, während in der Bestätigungsphase keine Veränderung der Route und der Zeitfenster erfolgt. In den Routineinträgen werden die folgenden Informationen gespeichert: eindeutige ID der Transporteinheit, Ankunftszeit, Abfahrtszeit, Priorität, Lebenszeit (Engl.: time-to-live), ID des vorherigen Moduls, ID des nachfolgenden Moduls und Zustand des Routineintrages (der Zustand kann entweder „angefragt“ oder „bestätigt“ sein).

Die eindeutige ID zur Identifizierung der Transporteinheit wird der Quelle von einem externen System mitgeteilt. Auch die Priorität kann von einem externen System mitgeteilt werden. Wenn keine Priorität mitgeteilt wird, dann erzeugt die Quelle eine Priorität basierend auf dem Einschleusezeitpunkt der Transporteinheit. Je weiter der Einschleusezeitpunkt zurückliegt, umso höher ist die Priorität der Transporteinheit. Somit steigt die Priorität mit zunehmendem Alter der Transporteinheiten. Die eindeutige ID der Transporteinheit, die Priorität und die Lebenszeit werden von der Quelle bei der Erstellung einer neuen Reservierungsanfrage abgespeichert und von den nachfolgenden Modulen nicht mehr verändert.

Die einzelnen Module überprüfen zyklisch die Lebenszeit der Reservierungsanfragen. Wenn die Lebenszeit abgelaufen ist, werden die dazugehörigen Anfragen somit auf allen Modulen gleichzeitig gelöscht. Anschließend wird von der Quelle eine neue Routinganfrage gestartet. Das Einführen einer Lebenszeit ist notwendig, da auf Grund der dezentralen Steuerung kein Modul einen vollständigen Überblick über das Fördersystem besitzt. Ohne einen vollständigen Überblick über das Fördersystem können die Quellen nur schätzen, wie lange der Reservierungsprozess dauern wird. Die Abschätzung der Lebenszeit wird im Kapitel 6.1.2 erläutert.

Abbildung 6.1 liefert eine Übersicht über die Verarbeitung der empfangenen Nachrichten. Hierbei können die Fördermodule die folgenden Nachrichten senden und empfangen: HINFÄLLIG, ABLEHNUNG, ANFRAGE und BESTÄTIGUNG. Die Verarbeitung dieser Nachrichten in der hier gewählten Reihenfolge wirkt sich positiv auf den Paketdurchsatz des Fördersystems aus. Sowohl HINFÄLLIG- als auch ABLEHNUNGS-Nachrichten löschen Routineinträge im Ablaufplan eines Moduls, sodass wieder mehr freie Zeitfenster zur Verfügung stehen. Die Sub-Algorithmen zur Verarbeitung dieser vier ersten Nachrichten werden in dem Kapitel 6.1.3 beschrieben. Der Aufbau und der Inhalt sowohl der Routineinträge als auch der verschiedenen Nachrichten wird im Anhang A3 detaillierter dargestellt.

Sobald ein Modul eine Anfrage erhält, erstellt es einen neuen Routingeintrag für das angefragte Zeitfenster. Das Modul versucht nun in den neuen Routingeintrag seinen Ablaufplan einzufügen. Falls hierbei das Einfügen nicht erfolgreich ist, wird eine ABLEHNUNGS-Nachricht an das anfragende Fördermodul geschickt. Wenn jedoch das Einfügen erfolgreich ist, wird eine ANFRAGE-Nachricht an das Fördermodul mit der kürzesten angenommenen Zielzeit (angenommene Zeit zum Erreichen des Ausschleusepunktes) geschickt.

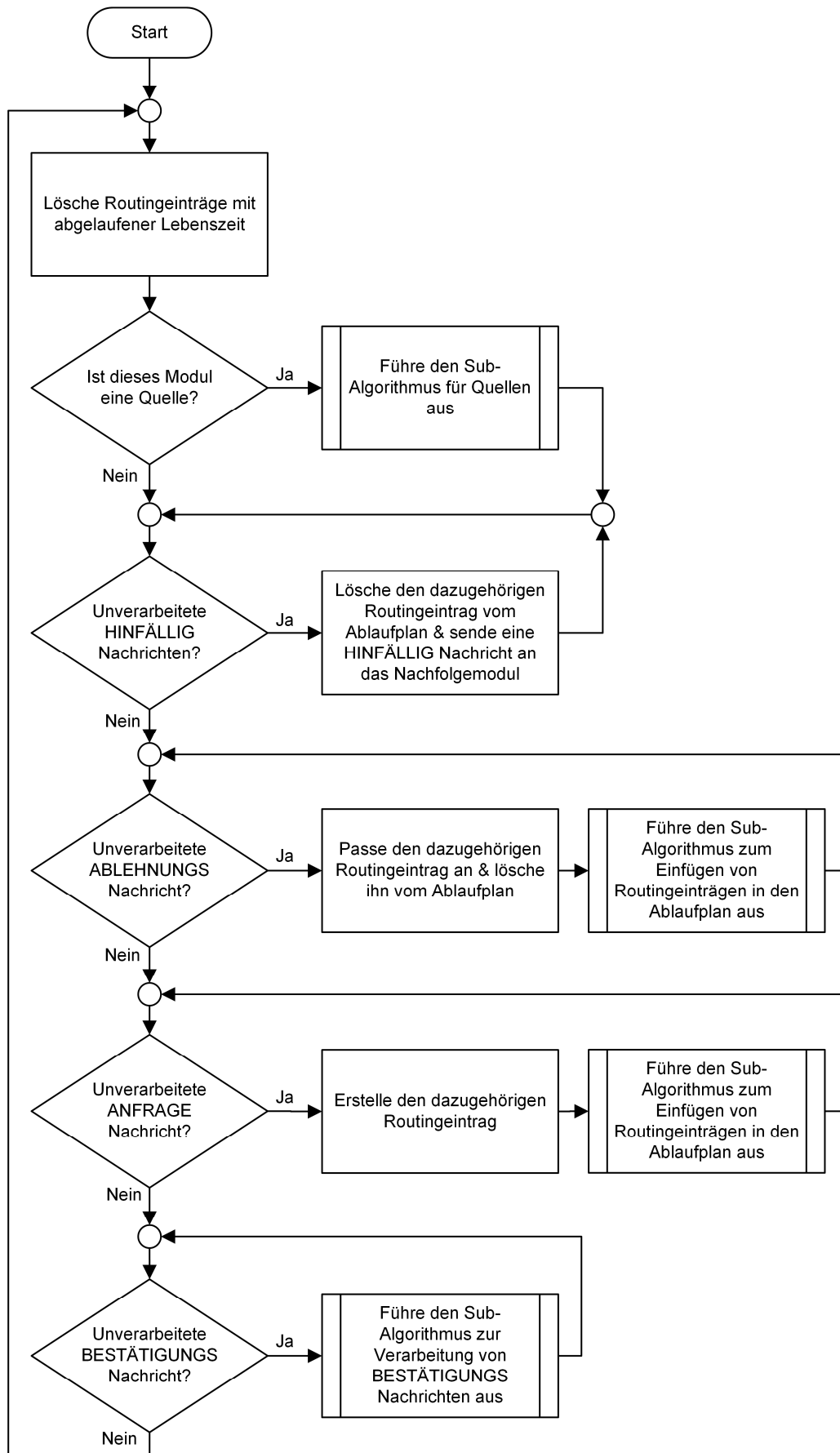


Abbildung 6.1: Algorithmus zur Verarbeitung von empfangenen Nachrichten

6.1.2 Erstellung einer neuen Reservierungsanfrage

Wenn eine Transporteinheit auf einer Quelle eingeschleust wird, bleibt das zur Quelle gehörende Fördermodul so lange blockiert, bis erfolgreich eine Route für die Transporteinheit reserviert wurde. Anschließend wird die Lebenszeit der Routinganfrage und die gewünschte Ankunftszeit auf dem nachfolgenden Modul geschätzt und eine ANFRAGE-Nachricht an das Modul mit der kürzesten Zielzeit geschickt. Jede ANFRAGE-Nachricht beinhaltet die eindeutige ID der Transporteinheit, die Priorität, die Ziel-ID der Senke, die gewünschte Ankunftszeit auf dem nachfolgenden Modul und die Lebenszeit der Anfrage.

Die gewünschte Ankunftszeit der Anfrage muss soweit in der Zukunft liegen, dass der gesamte Routenreservierungsprozess bis zu diesem Zeitpunkt erfolgreich abgeschlossen werden kann. Wie lange dies dauert, hängt von der Anzahl der Fördermodule zwischen Quelle und Senke und von der Anzahl der auftretenden zeitlichen Überlappungen mit Anfragen von anderen Quellen ab. Das Auflösen der Konflikte, die durch die zeitlichen Überlappungen entstehen können, kostet hierbei Zeit. Da Quellen keinen Überblick über den Zustand des gesamten Systems haben, kann die Anzahl der zeitlichen Überlappungen und somit die zusätzliche Zeit, die zur Auflösung der Konflikte benötigt wird, nur geschätzt werden. Somit kann die Quelle die Ankunftszeit, die Teil der ersten ANFRAGE-Nachricht ist, nur schätzen.

Wenn mehr Überlappungen auftreten als vorher abgeschätzt wurden, erreicht die BESTÄTIGUNGS-Nachricht die Quelle nach dem geschätzten Transportbeginn. Es ist somit unmöglich geworden, dass die Transporteinheit in dem vorher ausgehandelten Zeitfenster auf den entsprechenden Fördermodulen sein wird. Falls der Transportvorgang trotzdem begonnen wird, gibt es eine Differenz zwischen der vorher ausgehandelten und der tatsächlichen Zeit. Wie bereits im Kapitel 2.3.2 dargestellt, haben MAZA & CASTAGNA in [Maz05] bewiesen, dass es bei einer Differenz zwischen der ausgehandelten und der tatsächlichen Aufenthaltszeit zu keinen Konflikten kommt, wenn an Kreuzungen weiterhin die vorher ausgehandelte Reihenfolge der Transporteinheiten eingehalten wird. Obwohl ein Konflikt verhindert werden kann, ist es trotzdem vorteilhaft, dass es keine Differenz zwischen der ausgehandelten und der tatsächlichen Aufenthaltszeit gibt, da die Fördermodule bei der Reservierung von neuen Routen sich an den bisher ausgehandelten Aufenthaltszeiten orientieren. Durch eine Differenz zwischen den ausgehandelten und den tatsächlichen Zeiten werden möglicherweise nicht optimale Routen gewählt, wodurch der Durchsatz des Gesamtsystems verringert wird.

Um die Erzeugung dieser Differenz zu verhindern, wird bei der Erstellung einer Routinganfrage eine Lebenszeit berechnet. Diese Lebenszeit stellt eine Abschätzung der

Quelle dar, wie lange die Routenreservierung wahrscheinlich dauern wird. Eine Sicherheitsmarge wird zu dieser Abschätzung hinzuaddiert, um die Wahrscheinlichkeit zu erhöhen, dass die Routinganfrage innerhalb der Lebenszeit bestätigt wird. Falls die Lebenszeit zu konservativ geschätzt wird und die BESTÄTIGUNGS-Nachricht weit vor dem Ablauf der Lebenszeit bei der Quelle eintrifft, kann die Transporteinheit nicht früher starten, da bei einem zu frühen Start ebenfalls eine Differenz zwischen den tatsächlichen und den ausgehandelten Zeiten entsteht. Falls die Lebenszeit jedoch zu optimistisch geschätzt wird, erreicht die BESTÄTIGUNGS-Nachricht zu spät die Quelle, sodass die vorher ausgehandelten Aufenthaltszeiten nicht erreicht werden können. In diesem Fall werden alle schon bestehenden Anfragen gelöscht und eine neue Reservierungsanfrage wird von der Quelle erstellt.

In dieser Arbeit werden zwei Methoden zur Abschätzung der Lebenszeit vorgestellt. Bei der ersten Methode basiert die Schätzung der Lebenszeit auf der durchschnittlichen Zeit \bar{t}_{Ziel} , die vorangegangene Routenanfragen benötigt haben, um bestätigt zu werden. Jede Quelle speichert für jede erreichbare Senke die durchschnittliche Zeit ab. Da Routinganfragen, die nicht innerhalb der Lebenszeit bestätigt wurden, den Wert von \bar{t}_{Ziel} erhöhen, ist diese Methode selbstkorrigierend. In der Formel für die Berechnung der Lebenszeit nach der ersten Methode ist \bar{t}_{Ziel} der erste Term.

Es ist möglich, dass die Anzahl der Transporteinheiten im Fördersystem kurzzeitig ansteigt und somit deutlich mehr Überlappungen aufgelöst werden müssen. Hierdurch dauert der Abschluss der Routenreservierung wesentlich länger als im Durchschnitt. Um diesen Fall zu berücksichtigen, wird ein zweiter Term eingeführt: $n_s \cdot j$. j ist hierbei ein Iterationszähler, der bei 1 startet und der nach jedem erfolglosen Versuch einer Routenreservierung um 1 erhöht wird. Wenn die Routenreservierung erfolgreich abgeschlossen wird, wird der Iterationszähler wieder auf 1 zurückgesetzt. j wird anschließend mit dem Sicherheitsfaktor n_s multipliziert. Der Einfluss des Sicherheitsfaktors auf den Durchsatz wird in dem Kapitel 7 näher untersucht. Die vollständige Formel zur Berechnung der Lebenszeit LZ_D ist folgendermaßen definiert:

$$LZ_D = \bar{t}_{Ziel} + n_s \cdot j \quad (6.1)$$

Wenn die Anordnung der Fördermodule geändert wird, verlieren alle vorher berechneten durchschnittlichen Zeiten ihre Gültigkeit. Solange noch keine durchschnittlichen Zeiten verfügbar sind, wird eine zweite Methode zur Berechnung der Lebenszeit vorgestellt. Die Formel für diese zweite Methode lautet:

$$LZ_S = 2 \cdot t_{com} \cdot n_{Ziel} \cdot j \quad (6.2)$$

n_{Ziel} ist hierbei die Anzahl der Fördermodule zwischen der Quelle und der Senke, die aus der Routingtabelle entnommen wird und t_{com} ist die Zeit, die benötigt wird, um

eine Nachricht von einem Modul zu einem benachbarten Modul zu schicken. Die 2 wird benötigt, da jedes Modul entlang der Route mindestens zwei Nachrichten verarbeiten muss: Eine ANFRAGE- und eine BESTÄTIGUNGS-Nachricht. j ist genau wie in der Formel 6.1 der Iterationszähler für die fehlgeschlagenen Versuche. Sobald Werte für \bar{t}_{Ziel} verfügbar sind, wechselt die Quelle automatisch von der zweiten Methode zu der ersten Methode.

Wenn die Quelle die Lebenszeit berechnet hat, wird eine Nachricht zu demjenigen Fördermodul gesendet, welches die kürzeste projektierte Ankunftszeit zu der angefragten Senke aufweist. Diese Ankunftszeit setzt sich aus zwei Komponenten zusammen. Die erste Komponente ist die Basiszeit. Die Basiszeit wird berechnet, indem die verbleibende Distanz zu der Senke dividiert wird durch die einheitliche Fördergeschwindigkeit der Module. Die zweite Komponente der projektierten Ankunftszeit ist eine temporäre Erhöhung durch eine vorher empfangene ABHLEHNUNGS-Nachricht. Da jedes Fördermodul bei dem Senden einer ABLEHNUNGS-Nachricht den Beginn des nächsten freien Zeitfensters mitschickt, speichert das empfangene Modul ab, wann das sendende Modul frühestens wieder eine Transporteinheit aufnehmen kann. Durch diese temporäre Erhöhung ist es möglich, dass ein Fördermodul eine ANFRAGE-Nachricht nicht zu dem benachbarten Modul mit der kürzesten Distanz zum Ziel schickt, sondern zu dem Modul mit der kürzesten projektierten Ankunftszeit. Der Sub-Algorithmus zum Starten einer neuen Routinganfrage ist in der Abbildung 6.2 dargestellt. Das Einfügen der neu erstellten Routingeinträge in den Ablaufplan wird im nächsten Kapitel erläutert.

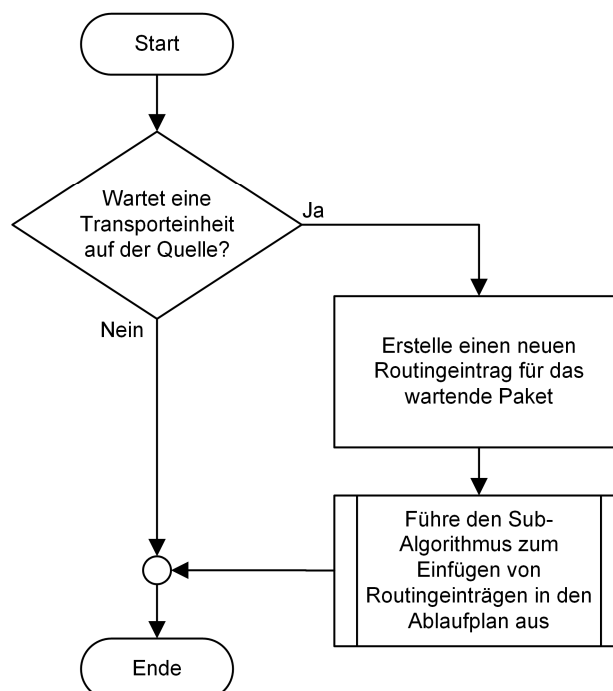


Abbildung 6.2: Sub-Algorithmus zum Starten einer neuen Routinganfrage

6.1.3 Verarbeitung von empfangenen Nachrichten

Sobald ein Fördermodul eine ANFRAGE-Nachricht erhält, erstellt es zunächst einen Routingeintrag mit der angefragten Ankunftszeit. Im nächsten Schritt wird der Ablaufplan des Moduls gefiltert: Alle Routingeinträge, die im Zustand „angefragt“ sind und eine niedrigere Priorität als die gerade erhaltene ANFRAGE-Nachricht haben, werden herausgefiltert. Anschließend wird versucht, den neu erstellen Routingeintrag in den Ablaufplan des Moduls einzufügen. Hierbei wird der Algorithmus von ALLEN, der in [All83] beschrieben ist, zum Einfügen genutzt. Falls hierbei das Einfügen nicht erfolgreich ist, wird eine ABLEHNUNGS-Nachricht an das anfragende Fördermodul geschickt. Wenn jedoch das Einfügen erfolgreich ist, wird eine ANFRAGE-Nachricht an das Fördermodul mit der kürzesten angenommenen Zeit zum Erreichen des Ausschleusepunktes geschickt. Wenn bei dem Einfügen Routingeinträge mit einer niedrigeren Priorität überschrieben worden sind, werden an das nachfolgende und das vorherige Fördermodul des überschriebenen Routingeintrages HINFÄLLIG- bzw. ABLEHNUNGS-Nachrichten geschickt.

Wenn der neu erstellte Routingeintrag erfolgreich in den Ablaufplan der Senke eingefügt wurde, sendet die Senke eine BESTÄTIGUNGS-Nachricht an das Vorgängermodul zurück. Der Sub-Algorithmus zum Einfügen von Nachrichten ist in Abbildung 6.3 dargestellt.

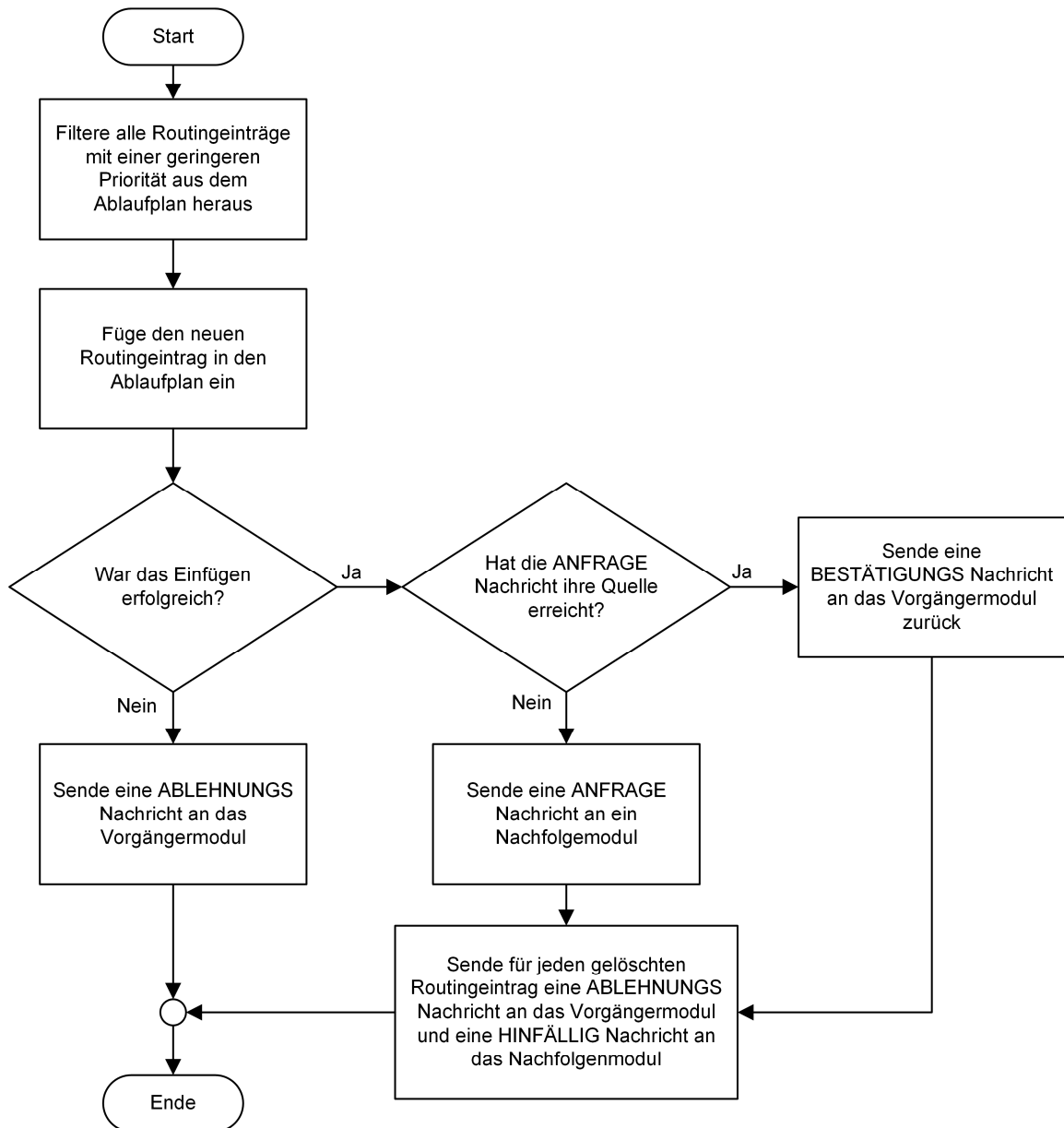


Abbildung 6.3: Einfügen einer neuen Routinganfrage in den Ablaufplan

Für jeden Routingeintrag, der durch das Einfügen des neu erstellten Routingeintrages gelöscht wurde, wird eine HINFÄLLIG-Nachricht an das Nachfolgemodul geschickt und eine ABLEHNUNGS-Nachricht an das Vorgängermodul.

Wenn ein Modul eine BESTÄTIGUNGS-Nachricht erhält, wird der dazugehörige Routingeintrag in den Zustand „bestätigt“ verändert. Falls die BESTÄTIGUNGS-Nachricht die Quelle erreicht hat, beginnt der Transportprozess, sobald der vorher geschätzte Abfahrtszeitpunkt eintritt. Der Sub-Algorithmus zur Verarbeitung der BESTÄTIGUNGS-Nachrichten ist in der Abbildung 6.4 dargestellt.

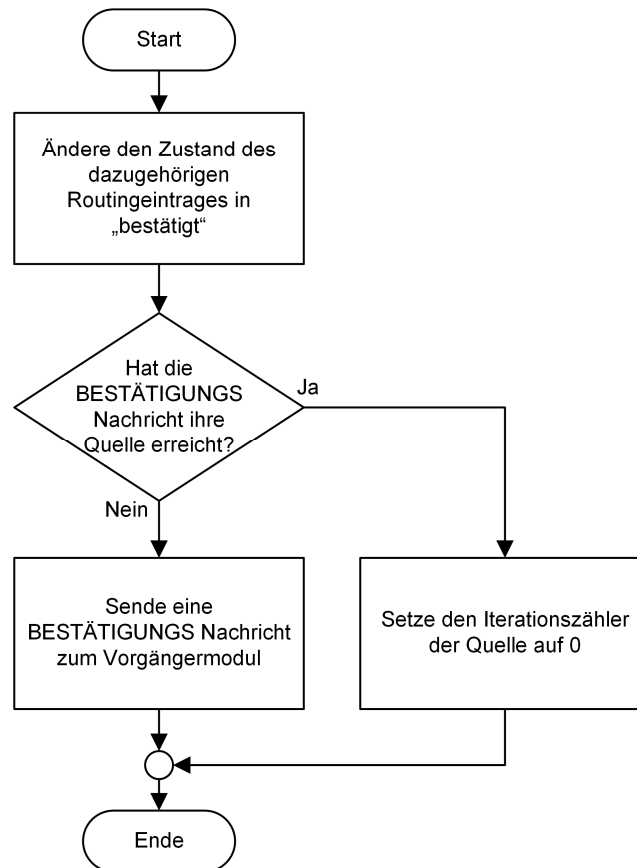


Abbildung 6.4: Sub-Algorithmus zur Verarbeitung von BESTÄTIGUNGS-Nachrichten

Wenn ein Fördermodul eine ABLEHNUNGS-Nachricht erhält, wird der dazugehörige Routingeintrag aus dem Ablaufplan entfernt. Die Abfahrtszeit des gerade entfernten Routingeintrages wird nun entsprechend der Abfahrtszeit angepasst, die in der ABLEHNUNGS-Nachricht vom Nachfolgemodul vorgeschlagen wurde. Anschließend wird der gleiche Algorithmus zum Einfügen von Routingeinträgen genutzt, der auch schon beim Verarbeiten von ANFRAGE-Nachrichten verwendet wurde (siehe Abbildung 6.3).

6.1.4 Anwendungsbeispiel

Die Funktionsweise des Algorithmus wird anhand der Abbildungen 6.5 bis 6.7 erläutert. In diesen Abbildungen ist eine Förderanlage dargestellt, die aus fünf großskaligen Fördermodulen besteht. Weiterhin werden die Ablaufpläne aller Fördermodule dargestellt. Diese Gesamtübersicht über alle Ablaufpläne besitzen die einzelnen Module nicht – jedes Fördermodul hat nur Zugriff auf den eigenen Ablaufplan.

Die ID der Fördermodule ist in der oberen linken Ecke der Fördermodule eingezeichnet. Zwei Transporteinheitenquellen sind in der Förderanlage vorhanden: Die Transporteinheitenquelle A besitzt die ID 2 und befindet sich am linken Rand der Förderanlage und die Transporteinheitenquelle B besitzt die ID 5 und befindet sich am unteren

Rand der Förderanlage. In der Förderanlage befindet sich eine Senke: Diese befindet sich am obersten Rand der Anlage und das dazugehörige Fördermodul hat die ID 1. Routingeinträge, die im Zustand „angefragt“ sind, werden mit einem einzigen Pfeil auf den Fördermodulen dargestellt. Routingeinträge im Zustand „bestätigt“ werden mit zwei Pfeilen dargestellt. Die Zuordnung der Routingeinträge zu den entsprechenden Transporteinheiten geschieht auf zwei Arten: Zum einen haben die Pfeile die gleiche Farbe wie die Transporteinheiten und zum anderen geht der erste Pfeil der Routinganfrage immer von der dazugehörigen Transporteinheit aus.

In den Ablaufplänen auf der horizontalen Achse ist die Zeit dargestellt, während entlang der vertikalen Achse die IDs der einzelnen Fördermodule angeordnet sind. Drei verschiedene Zeitfenstertypen existieren: Freie Zeitfenster f , in denen sich noch kein Routingeintrag befindet, Zeitfenster r mit einem Routingeintrag im Zustand „angefragt“ und Zeitfenster c mit einem Routingeintrag im Zustand „bestätigt“. Jedes Zeitfenster besitzt zwei Indizes: Mit dem unteren Index wird die ID des Fördermoduls gekennzeichnet und mit dem oberen Index werden die Zeitfenster vom gleichen Typ aufsteigend durchnummeriert.

In diesem Anwendungsbeispiel benötigt das Empfangen und Verarbeiten einer Nachricht 1 ms. Das Transportieren einer Transporteinheit von der geometrischen Mitte eines Fördermoduls zu einem benachbarten Fördermodul dauert ohne Stoppen 10 ms. Diese unrealistisch schnelle Transportgeschwindigkeit wurde gewählt, damit sich die Kommunikations- und Transportgeschwindigkeit in der gleichen Größenordnung befinden, wodurch diese in den Abbildungen besser dargestellt werden können.

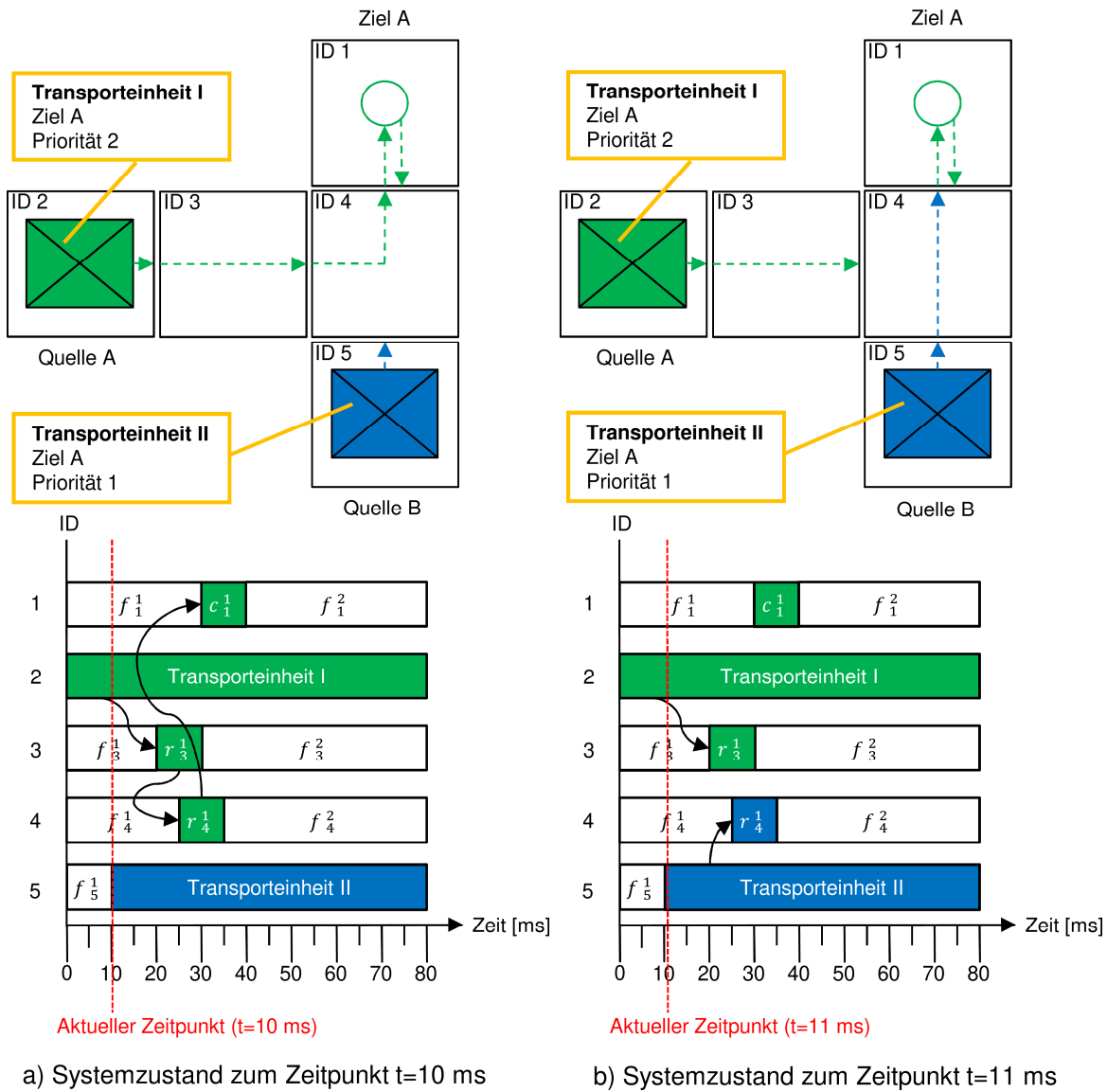
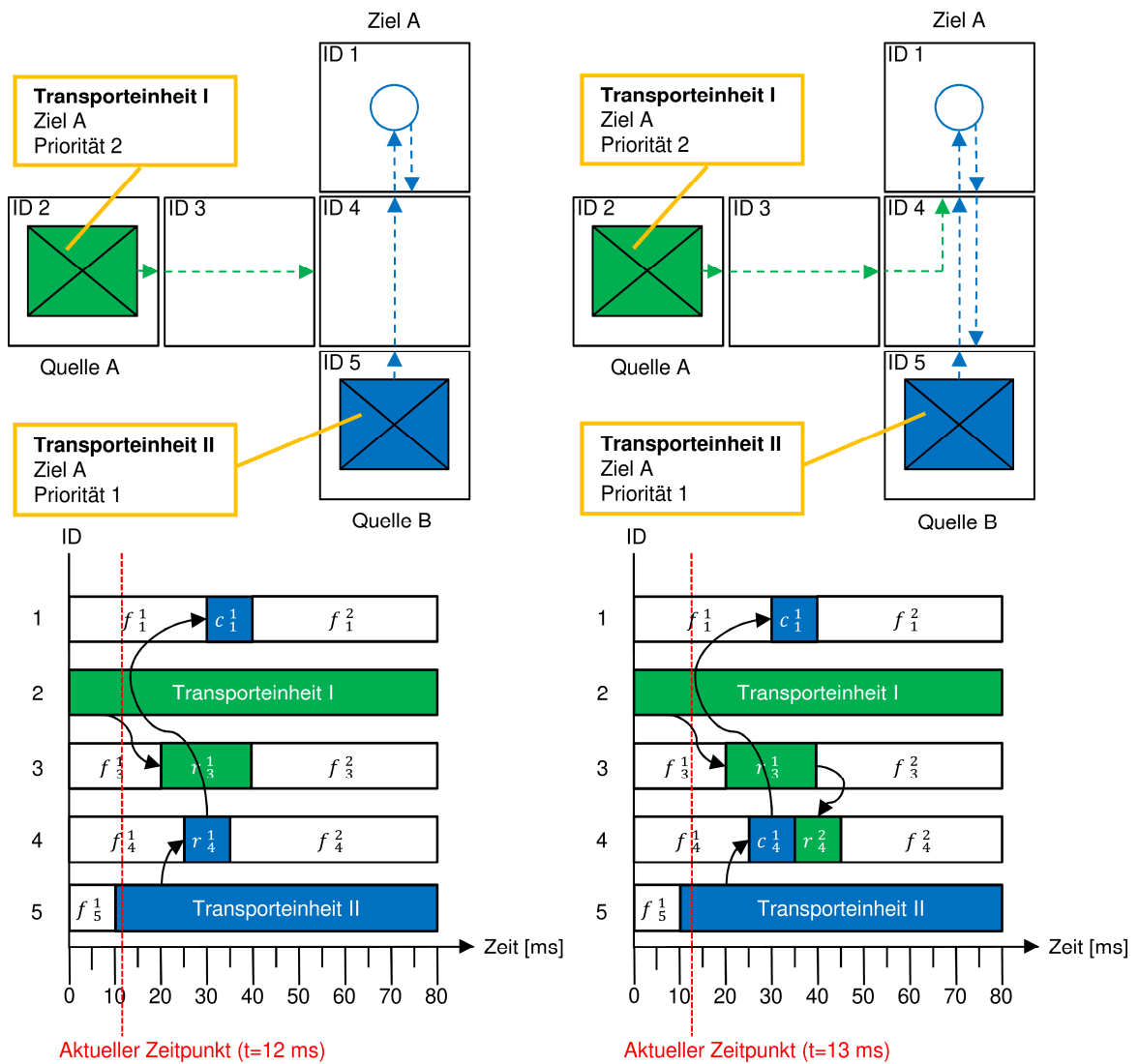


Abbildung 6.5: Systemzustände zum Zeitpunkt t=10 ms und t=11 ms

Zu dem Zeitpunkt t=10 ms (siehe Abbildung 6.5 a)) wartet auf der Quelle A die Transporteinheit I auf eine BESTÄTIGUNGS-Nachricht als Antwort auf die vorher gesendete ANFRAGE-Nachricht. Die zur Transporteinheit I gehörende ANFRAGE-Nachricht hat hierbei bereits das Ziel A erreicht, welches eine BESTÄTIGUNGS-Nachricht an das Fördermodul 4 zurückgesendet hat. Gleichzeitig wird die Transporteinheit II durch die Quelle B in das System eingeschleust. Auf Grund der neu eingeschleusten Transporteinheit II wird eine ANFRAGE-Nachricht zu dem Fördermodul mit der ID 4 gesendet. Die Transporteinheit II besitzt eine höhere Priorität als die Transporteinheit I, obwohl diese nach der Transporteinheit I eingeschleust wurde.

Zu dem Zeitpunkt t=11 ms (siehe Abbildung 6.6 b)) erhält das Fördermodul 4 sowohl eine BESTÄTIGUNGS Nachricht für die Transporteinheit I als auch eine ANFRAGE-Nachricht für die Transporteinheit II. Da ANFRAGE-Nachrichten immer vor BESTÄTI-

GUNGS-Nachrichten verarbeitet werden, wird die ANFRAGE-Nachricht für die Transporteinheit II zuerst verarbeitet. Weil sich die Zeitfenster der Routingeinträge für die Transporteinheit I und II überlappen, vergleicht der Algorithmus die Prioritäten der Transporteinheiten. Da die Transporteinheit II eine höhere Priorität besitzt als die Transporteinheit I, ersetzt der Routingeintrag der Transporteinheit II den Routingeintrag der Transporteinheit I. Das Modul 4 sendet nun drei Nachrichten aus: Eine HINFÄLLIG-Nachricht wird an das Modul 1 gesendet, um diesem mitzuteilen, dass der bereits bestätigte Routingeintrag für die Transporteinheit I gelöscht werden muss. Weiterhin wird eine ABLEHNUNGS-Nachricht an das Modul 3 gesendet. In dieser ABLEHNUNGS-Nachricht ist die Information enthalten, wann das nächste freie Zeitfenster auf dem Modul 4 verfügbar ist. Abschließend wird eine ANFRAGE-Nachricht für die Transporteinheit II an das Modul 1 gesendet.



a) Systemzustand zum Zeitpunkt t=12 ms

b) Systemzustand zum Zeitpunkt t=13 ms

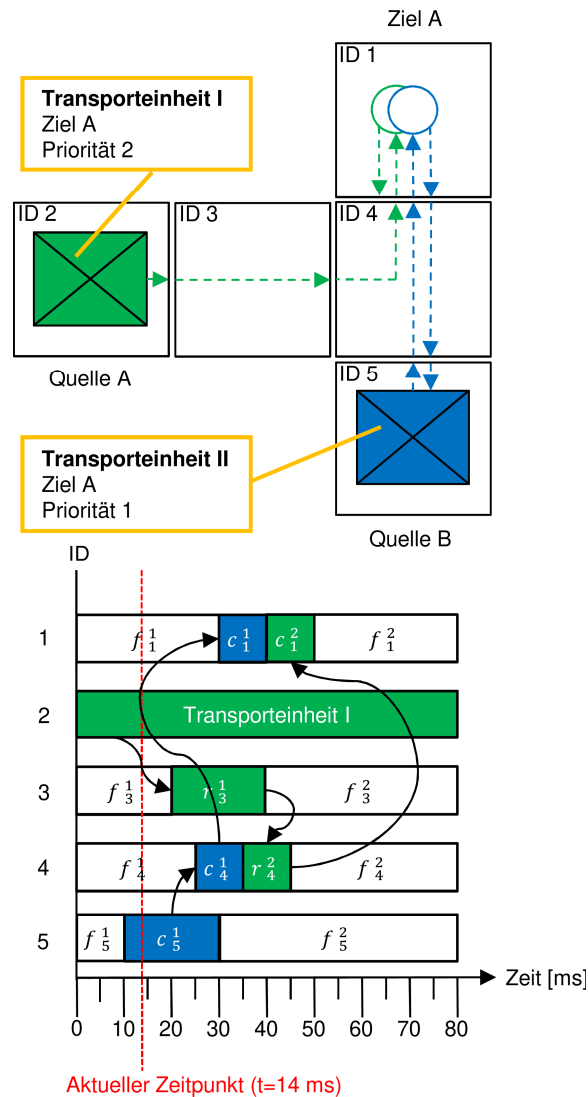
Abbildung 6.6: Systemzustände zum Zeitpunkt t=12 ms und t=13 ms

Zum Zeitpunkt $t=12$ ms (siehe Abbildung 6.6 a)) erhält das Modul 1 sowohl die HINFÄLLIG- als auch die ANFRAGE-Nachricht von dem Modul 4. Auf Grund der HINFÄLLIG-Nachricht, wird der Routingeintrag von der Transporteinheit I auf dem Modul 1 gelöscht. Anschließend wird auf Grund der empfangenen ANFRAGE-Nachricht ein neuer Routingeintrag für die Transporteinheit II auf dem Modul 1 erstellt. Da das Modul 1 die Senke für die Transporteinheit II ist, wird eine BESTÄTIGUNGS-Nachricht an das Modul 4 geschickt. Gleichzeitig erhält das Modul 3 die ABLEHNUNGS-Nachricht von dem Modul 4. Das Modul 3 passt nun das Zeitfenster des Routingeintrages von der Transporteinheit I an und sendet erneut eine ANFRAGE-Nachricht an das Modul 4.

Zum Zeitpunkt $t=13$ ms (siehe Abbildung 6.6 b)) erhält das Modul 4 zwei Nachrichten: Eine BESTÄTIGUNGS-Nachricht für die Transporteinheit II von dem Modul 1 und eine ANFRAGE-Nachricht für die Transporteinheit I von dem Modul 3. Da sich das Zeitfenster, welches in der ANFRAGE-Nachricht angefragt wird, mit keinem anderen Zeitfenster überschneidet, wird ein dazugehöriger Routingeintrag auf dem Modul 4 erstellt. Auf Grund des neu erstellten Routingeintrages sendet das Modul 4 eine ANFRAGE-Nachricht an das Modul 1. Nachdem die ANFRAGE-Nachricht verarbeitet wurde, wird als nächstes die BESTÄTIGUNGS-Nachricht verarbeitet. Die Verarbeitung besteht darin, dass der Routingeintrag für die Transporteinheit II in den Zustand „bestätigt“ wechselt. Anschließend wird eine BESTÄTIGUNGS-Nachricht an das Modul 5 gesendet.

Zum Zeitpunkt $t=14$ ms (siehe Abbildung 6.7) erhält das Modul 1 eine ANFRAGE-Nachricht von dem Modul 4. Da das angefragte Zeitfenster frei ist, wird ein entsprechender Routingeintrag erstellt. Da das Modul 1 der Ausschleusepunkt für die Transporteinheit I darstellt, befindet sich der Routingeintrag im Zustand „bestätigt“ und es wird eine BESTÄTIGUNGS-Nachricht an das Modul 4 zurückgesendet. Gleichzeitig erhält das Modul 5 die BESTÄTIGUNGS-Nachricht für die Transporteinheit II. Der dazugehörige Routingeintrag wechselt in den Zustand „bestätigt“. Da die BESTÄTIGUNGS-Nachricht die Quelle erreicht hat, wird diese nicht mehr durch die Transporteinheit II dauerhaft blockiert.

In dem hier dargestellten Beispiel sind die Auswirkungen einer zu konservativ berechneten Lebenszeit zu erkennen: Die Quellen haben beide eine Lebenszeit von 15 ms berechnet. Da jedoch die BESTÄTIGUNGS-Nachricht für die Transporteinheit II innerhalb von 4 ms die Quelle erreicht hat, muss die Transporteinheit II nun weitere 11 ms warten, bevor der Transportvorgang beginnen kann.

Abbildung 6.7: Systemzustand für den Zeitpunkt $t=14$ ms

6.1.5 Begründung der getroffenen Designentscheidungen

Bevor eine Routenreservierung durchgeführt werden kann, müssen zunächst Routingtabellen erstellt werden mit denen das momentane Layout der Förderanlage beschrieben wird. Der in dieser Arbeit beschriebene Algorithmus nutzt hierbei statische Routingtabellen, d.h. nach der initialen Erstellung der Tabellen werden diese nicht mehr verändert. Die Algorithmen von MAYER, SEIBOLD und KIM & TANCHOCO nutzen ebenfalls statische Routingtabellen, während KRÜHN in [Krü15] einen Ansatz mit dynamischen Routingtabellen beschreibt. Die Funktionsweise der dynamischen Routingtabellen von KRÜHN wird an dem folgenden Beispiel erläutert: Wenn eine Engstelle durch eine Route belegt wird, erhöhen die Module der Engstelle dynamisch ihre Kosten in die Gegenrichtung, d.h. andere Routenanfragen werden versuchen eine Reservierung dieser Engstelle in entgegengesetzter Richtung zu vermeiden. Wie stark die Kosten erhöht werden, kann hierbei anwendungsspezifisch eingestellt werden. Über diese

Einstellmöglichkeit kann festgelegt werden, ob die Reservierung von entgegengesetzten Routen in Engstellen zulässig ist. Falls das Planen von entgegengesetzten Routen zulässig ist und in dem Layout Alternativrouten zu der Engstelle existieren, kann über die Einstellmöglichkeit die maximale Länge von Alternativroute festgelegt werden, bevor die neue Route entgegen der bereits existierenden Route durch die Engstelle geplant wird. Die Erhöhung der Kosten, die durch eine belegte Engstelle entstehen, breiten sich ausgehend von der belegten Engstelle wellenartig aus.

Da die Routenreservierung von KRÜHN nicht zeitfensterbasiert ist, können Engstellen nur zwei Zustände aufweisen: Entweder es gibt keine reservierten Routen und die Kosten werden nicht erhöht oder es gibt mindestens eine reservierte Route und in die Gegenrichtung werden die Kosten erhöht. Der Algorithmus, der in dieser Arbeit beschrieben wurde, ist jedoch zeitfensterbasiert. Daher müssen auch die dynamischen Kosten zeitfensterbasiert sein. Für jeden Zeitpunkt gibt es hierbei nur zwei Zustände: Entweder eine Transporteinheit blockiert die Engstellen und diese ist nicht befahrbar oder die Engstelle ist frei und somit befahrbar. Jedes Mal, wenn ein neuer Routingeintrag erstellt wird, müssen alle benachbarten Module über den neu erstellten Routingeintrag informiert werden, damit diese die Information besitzen, ob die Engstelle befahrbar ist. Diese Information muss ebenfalls wellenartig im Fördersystem verteilt werden. Das Erstellen eines einzigen neuen Routingeintrages erzeugt daher eine große Anzahl von Nachrichten. Die Information, ob ein Modul belegt ist, wird nach dem Push-Prinzip verteilt. Um diese große Anzahl an Nachrichten zu vermeiden, wurde auf die Implementierung der dynamischen Routingtabellen verzichtet. Stattdessen teilen die Module erst nach erfolgter ANFRAGE-Nachricht mit, ob es möglich ist, zu diesem Zeitpunkt einen Routingeintrag zu erstellen. Es wird somit nach dem Pull-Prinzip gearbeitet.

Die Verwendung von dynamischen Kosten kann die Zeit verkürzen, die benötigt wird, um eine Route zu reservieren, da die freien Zeitfenster eines Modules den Nachbarn bekannt sind und nicht erst angefragt werden müssen. Die Erweiterung des in dieser Arbeit beschriebenen Algorithmus um diese Funktion sollte daher in weiterführenden Arbeiten untersucht werden.

SEIBOLD hat in [Sei16b] die Reservierung von unvollständigen Routen beschrieben. Eine unvollständige Routenreservierung liegt vor, wenn eine Route nicht bis zur Senke reserviert wird, sondern auf einem Modul endet, welches näher zur Senke liegt als die Quelle. Die Quelle schätzt die Zeit ab, wie lange die Routenreservierung bis zur Senke benötigt. Falls sich ungewöhnlich viele Transporteinheiten im Fördersystem befinden, erreicht die Routenreservierung jedoch nicht die Senke innerhalb der abgeschätzten

Zeit. In diesem Fall wird das Modul, welches bereits einen entsprechenden Routineeintrag besitzt und sich am nächsten zur Senke befindet, als Zwischenziel definiert. Ein Modul eignet sich hierbei nur als Zwischenziel, wenn es keine weiteren Routineeinträge im Ablaufplan gibt, da die Transporteinheit auf diesem Zwischenziel so lange warten muss, bis die Routenreservierung zum nächsten Zwischenziel oder zur Senke erfolgreich durchgeführt wurde. Die Transporteinheit wird nun zu diesem Zwischenziel transportiert von welchem aus der Routenreservierungsprozess fortgesetzt wird.

Bei der Auswahl eines Zwischenziels prüft der Algorithmus von SEIBOLD nur, ob es keine weiteren Routineeinträge auf dem Zwischenziel gibt. Somit ist es möglich, dass als Zwischenziel ein Modul ausgewählt wird, welches einen Engpass darstellt. Da Zwischenziele so lange blockiert sind, bis der Routenreservierungsprozess erfolgreich abgeschlossen ist, können andere Transporteinheiten in diesem Zeitraum den Engpass nicht passieren, was sich negativ auf den Durchsatz auswirkt. Auf Grund dieser Möglichkeit der negativen Auswirkung auf den Durchsatz wurde die Reservierung von unvollständigen Routen im Rahmen dieser Arbeit nicht implementiert.

Weiterhin ist es möglich, dass es zum Verhungern an Quellen kommt, wenn sich eine ungünstige Reihenfolge von Reservierungsschritten zyklisch wiederholt. Das Auftreten eines solchen Zyklus ist jedoch äußerst unwahrscheinlich, sodass die notwendigen Gegenmaßnahmen zur Verhinderung nicht im Hauptteil dieser Arbeit beschrieben werden, sondern im Anhang A4 dargestellt sind. In den meisten Layouts (siehe Kapitel 7) reicht es aus, wenn das Verhungern verhindert wird durch die Priorisierung der Transporteinheiten anhand des Einschleusezeitpunktes.

6.2 Modifikation des Algorithmus für kleinskalige Fördermodule

6.2.1 Beschreibung der Modifikation

Da der Algorithmus für kleinskalige Fördermodule eine Modifikation des Algorithmus für großskalige Fördermodule darstellt, baut dieses Kapitel auf dem Kapitel 6.1 auf. Der Algorithmus zur Ansteuerung von kleinskaligen Fördermodulen mit einer modulbasierten Ansteuerung nutzt das von KRÜHN in [Krü15] beschriebene Konzept der Modulnachbarschaften. Jede Nachricht muss um die Information erweitert werden, wie groß die dazugehörige Transporteinheit ist, da anhand dieser Information die Größe der benötigten Modulnachbarschaft bestimmt werden kann. In jeder Modulnachbarschaft gibt es immer ein Hauptmodul. Als Hauptmodul einer Nachbarschaft wird das

Modul festgelegt, welches momentan unter dem geometrischen Mittelpunkt der Transporteinheit liegt. Somit gibt es während des Transportes einer Transporteinheit mehrere Hauptmodule, von denen jedoch immer nur ein einziges aktiv ist. Jeder Routineeintrag muss zusätzlich noch um die Information erweitert werden, von welchem Modul der Routineeintrag erstellt wurde, da es möglich ist, dass Einträge sowohl vom Modul selbst erstellt werden können als auch von anderen Modulen. Drei weitere Nachrichten werden zur Koordinierung innerhalb der Modulnachbarschaft eingeführt: Die ABLAUFPLAN-ANFRAGE-, die ABLAUFPLAN- und die ABLAUFPLAN-BESTÄTIGUNGS-Nachricht. Die Bedeutung dieser Nachrichten wird im Folgenden erläutert. Die Verarbeitung von den neu eingeführten Nachrichten erfolgt vor der Verarbeitung der bereits bestehenden Nachrichten, wie in Abbildung 6.8 dargestellt ist.

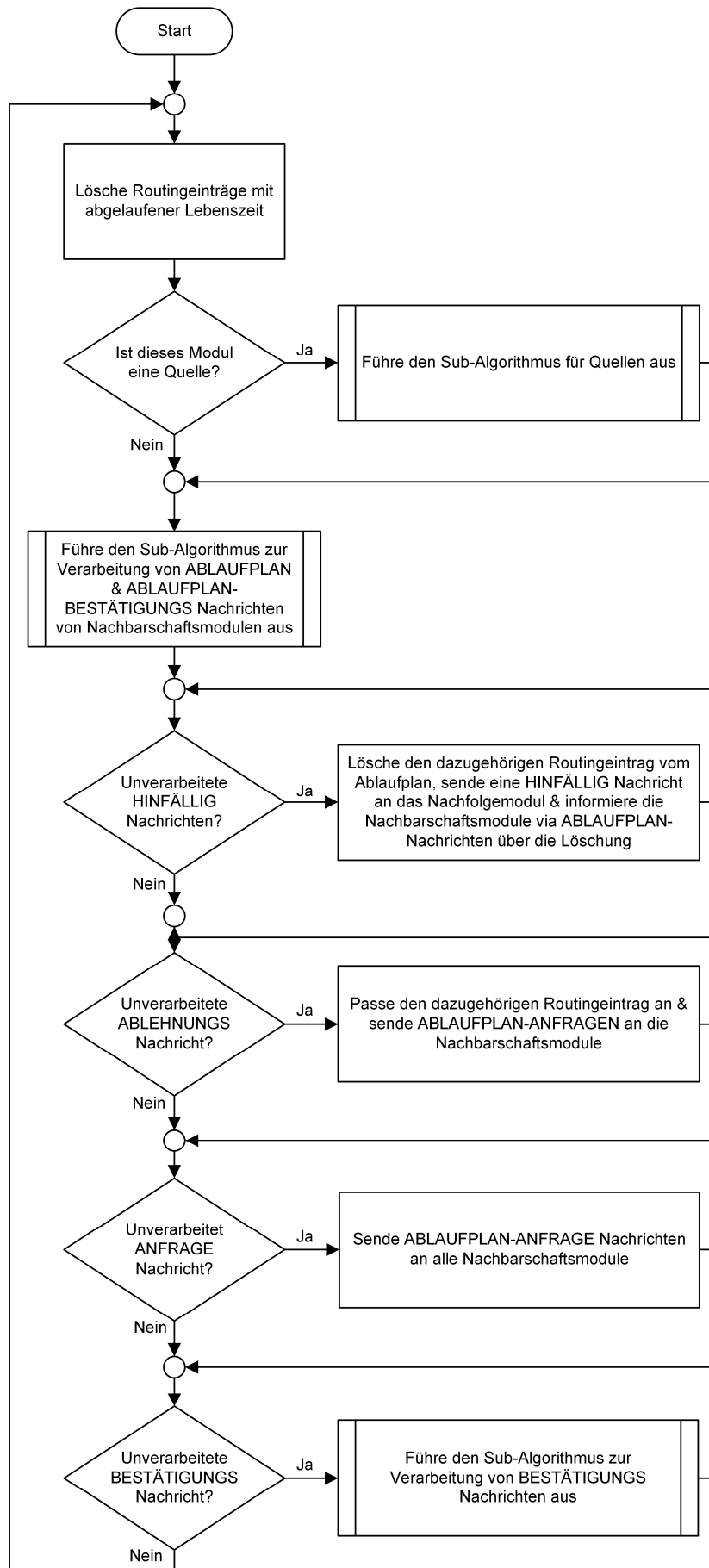


Abbildung 6.8: Verarbeitung von Nachrichten durch kleinskalige Fördermodule

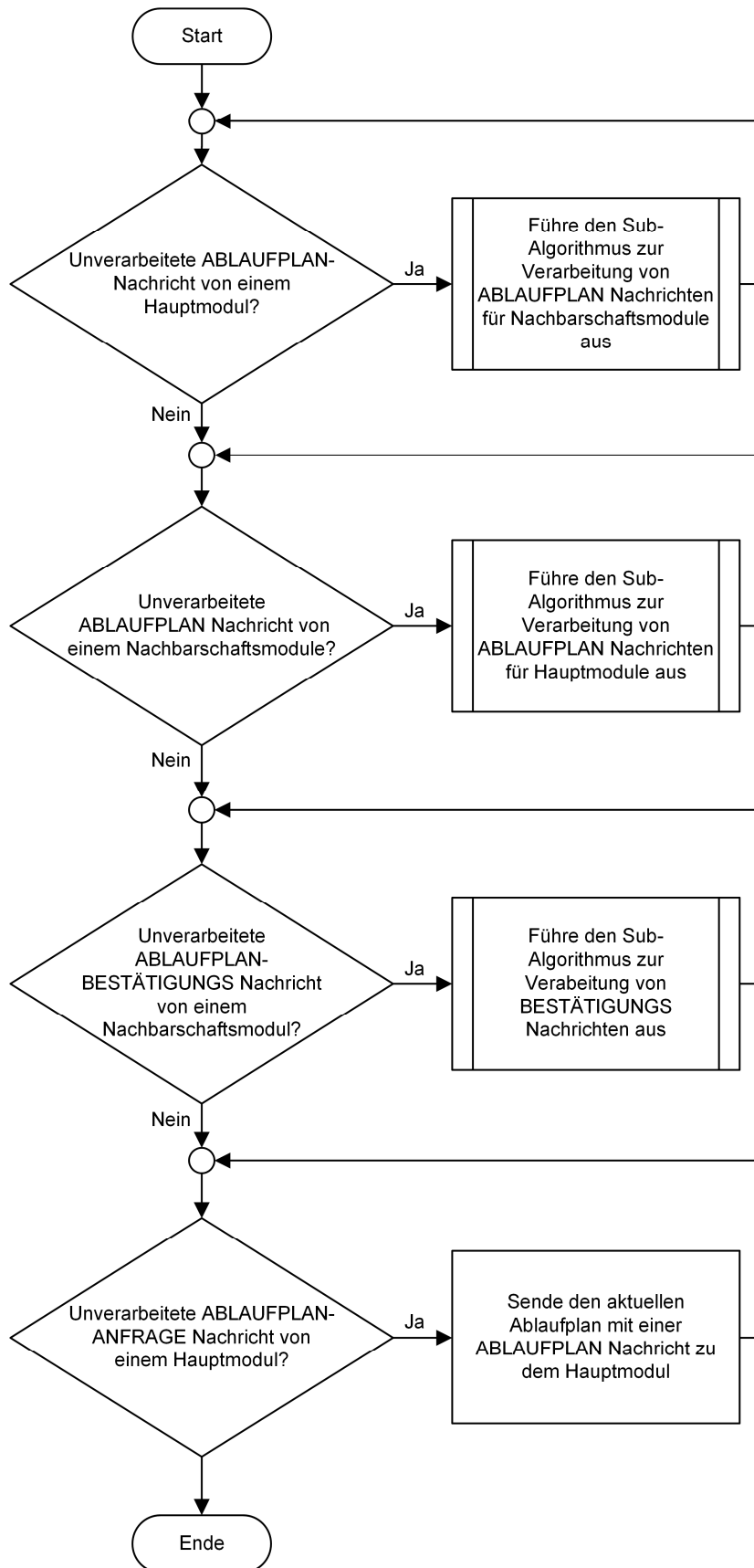


Abbildung 6.9: Verarbeitung von ABLAUFPLAN-, ABLAUFPLAN-BESTÄTIGUNGS- und ABLAUFPLAN-ANFRAGE-Nachrichten

In der Abbildung 6.9 ist die Verarbeitung der neuen Nachrichtentypen dargestellt. Bei der Verarbeitung der ABLAUFPLAN-Nachrichten wird zwischen zwei Fällen unterschieden. Beim ersten Fall erhält ein Hauptmodul eine ABLAUFPLAN-Nachricht von einem Nachbarschaftsmodul als Antwort auf eine ABLAUFPLAN-ANFRAGE-Nachricht. Im zweiten Fall erhält wiederum das Nachbarschaftsmodul seinen eigenen modifizierten Ablaufplan von einem Hauptmodul zurück, nachdem das Hauptmodul einen neuen Routingeintrag hinzugefügt hat.

Die Nutzung der neu eingeführten Nachrichtentypen wird am Beispiel der Verarbeitung einer ANFRAGE-Nachricht erläutert. Wenn ein Modul eine ANFRAGE-Nachricht erhält, nimmt es selbst die Rolle eines Hauptmoduls ein. Das Hauptmodul sendet nun eine ABLAUFPLAN-ANFRAGE-Nachricht an alle Nachbarschaftsmodule. Wie in Abbildung 6.9 dargestellt ist, antworten die Nachbarschaftsmodule mit einer ABLAUFPLAN-Nachricht.

Wenn das Hauptmodul von allen Nachbarschaftsmodulen Ablaufpläne erhalten hat, wird vom Hauptmodul basierend auf der ANFRAGE-Nachricht ein Routingeintrag erstellt und es wird versucht, den Routingeintrag in die Ablaufpläne aller Module der Nachbarschaft einzufügen. Der entsprechende Sub-Algorithmus ist in Abbildung 6.10 dargestellt. Falls es möglich ist, den Routingeintrag in alle Ablaufpläne einzufügen, werden die modifizierten Ablaufpläne zurück zu den Nachbarschaftsmodulen gesendet. Falls es nicht möglich ist, den Routingeintrag in alle Ablaufpläne einzufügen, wird eine ABLEHNUNGS-Nachricht an das Vorgängermodul gesendet.

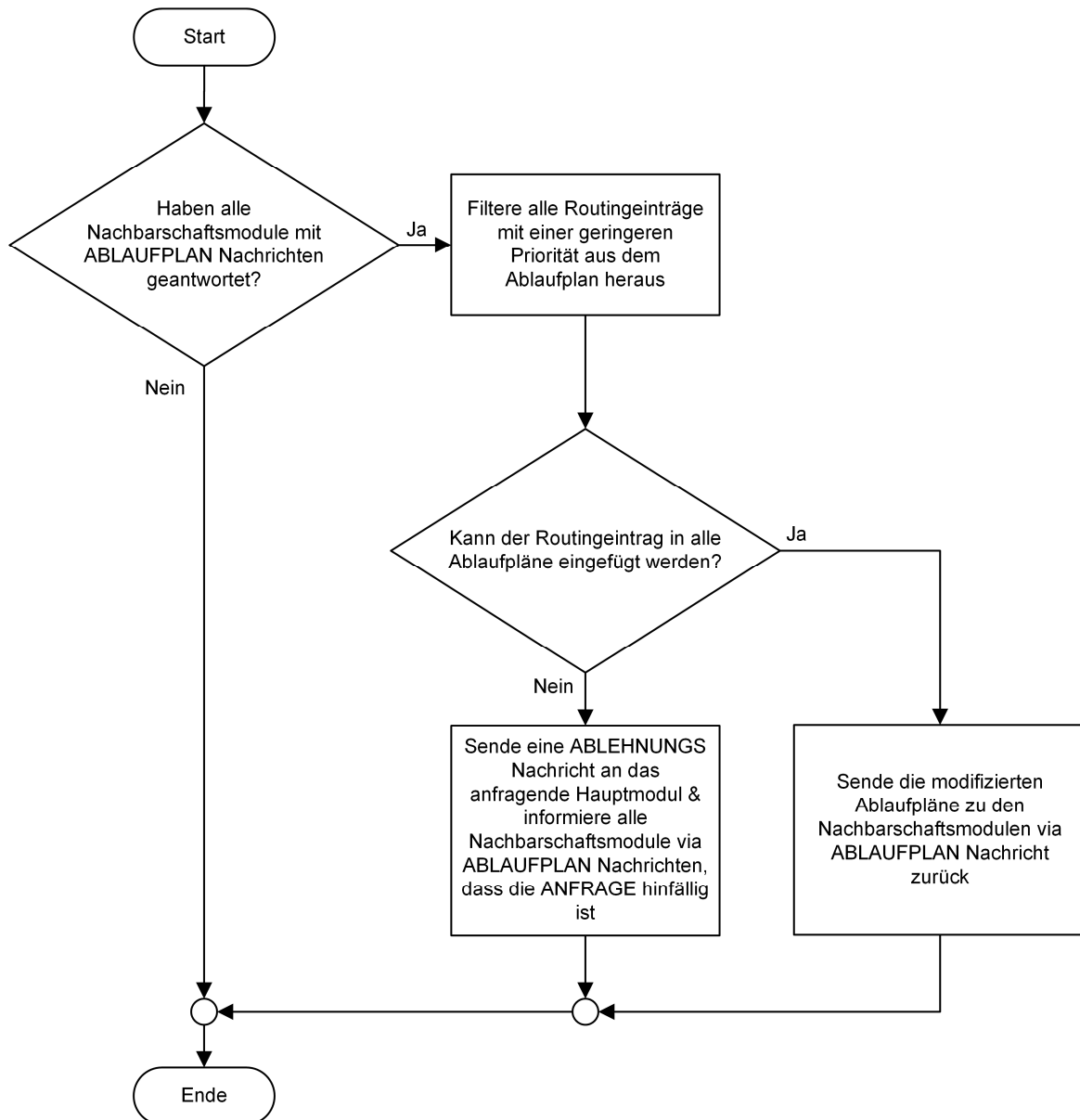


Abbildung 6.10: Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN-Nachrichten durch Hauptmodule

Die Verarbeitung der modifizierten Ablaufpläne durch die Nachbarschaftsmodule ist in der Abbildung 6.11 dargestellt. Es ist möglich, dass ein Modul von zwei verschiedenen Hauptmodulen gleichzeitig als Nachbarschaftsmodul angefragt wird. Somit kann es sein, dass die Modifikationen der Hauptmodule sich gegenseitig widersprechen. In diesem Fall vergleicht das Nachbarschaftsmodul die Prioritäten beider Routingeinträge. Hierbei wird der Eintrag mit der höheren Priorität angenommen und eine ABLAUFPLAN-BESTÄTIGUNGS-Nachricht an das Hauptmodul gesendet. An das Hauptmodul, dessen Routingeintrag nicht angenommen wurde, wird der aktualisierte Ablaufplan via ABLAUFPLAN-Nachricht geschickt. Dieser Prozess wird zu besserer Veranschaulichung in dem Kapitel 6.2.2 beispielhaft dargestellt. Wenn bei der Annahme eines mo-

difizierten Ablaufplanes Routingeinträge mit einer niedrigeren Priorität gelöscht werden, werden alle Module, die zu dem überschriebenen Routingeintrag gehören, über die Löschung informiert.

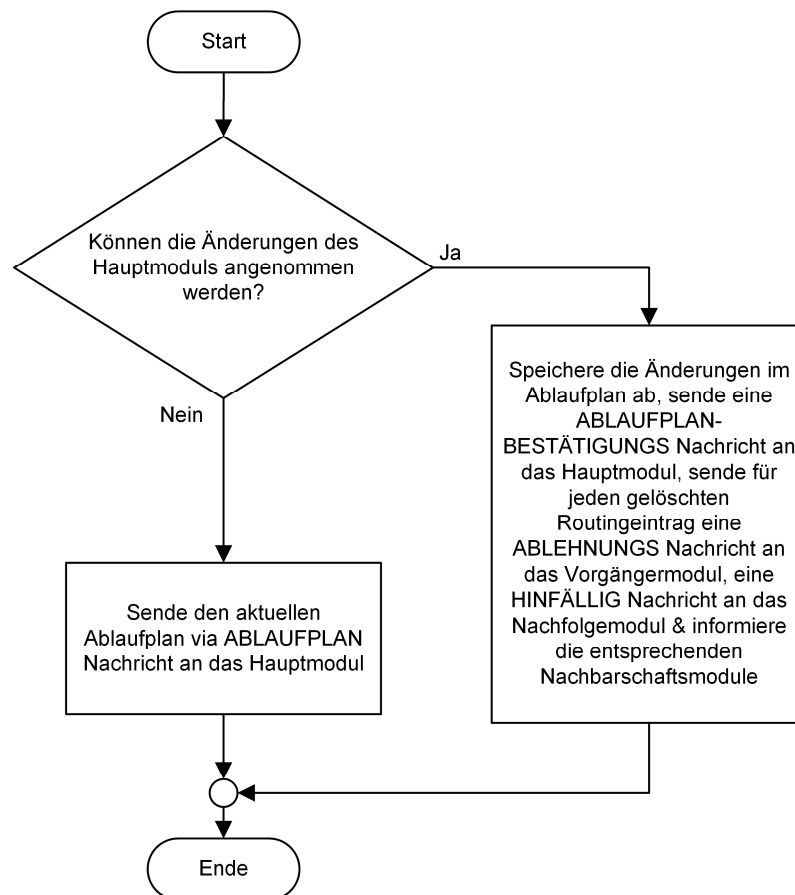


Abbildung 6.11: Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN-Nachrichten durch Nachbarschaftsmodule

Der Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN-BESTÄTIGUNGS-Nachrichten ist in Abbildung 6.12 dargestellt. Das Hauptmodul prüft zunächst, ob alle Nachbarschaftsmodule ABLAUFPLAN-BESTÄTIGUNGS-Nachrichten gesendet haben. Das Hauptmodul überprüft anschließend, ob die Routinganfrage die entsprechende Senke erreicht hat. Falls die Senke erreicht wurde, wird eine BESTÄTIGUNGS-Nachricht an das Vorgängermodul gesendet. Falls die Senke nicht erreicht wurde, wird eine ANFRAGE-Nachricht an das nachfolgende Modul gesendet. Wenn bei der endgültigen Annahme der Routinganfrage Routingeinträge mit einer geringeren Priorität gelöscht wurden, werden alle Module, die zu dem überschriebenen Routingeintrag gehören, über die Löschung informiert.

Um diese Dissertation kurz zu halten, wird das Verarbeiten von ABLEHNUNGS-, HINFÄLLIG- und BESTÄTIGUNGS-Nachrichten nicht ausführlich erläutert, da der grundsätzliche Ablauf, um Informationen in einer Modulnachbarschaft zu verteilen, identisch zu der Verarbeitung von ANFRAGE-Nachrichten ist.

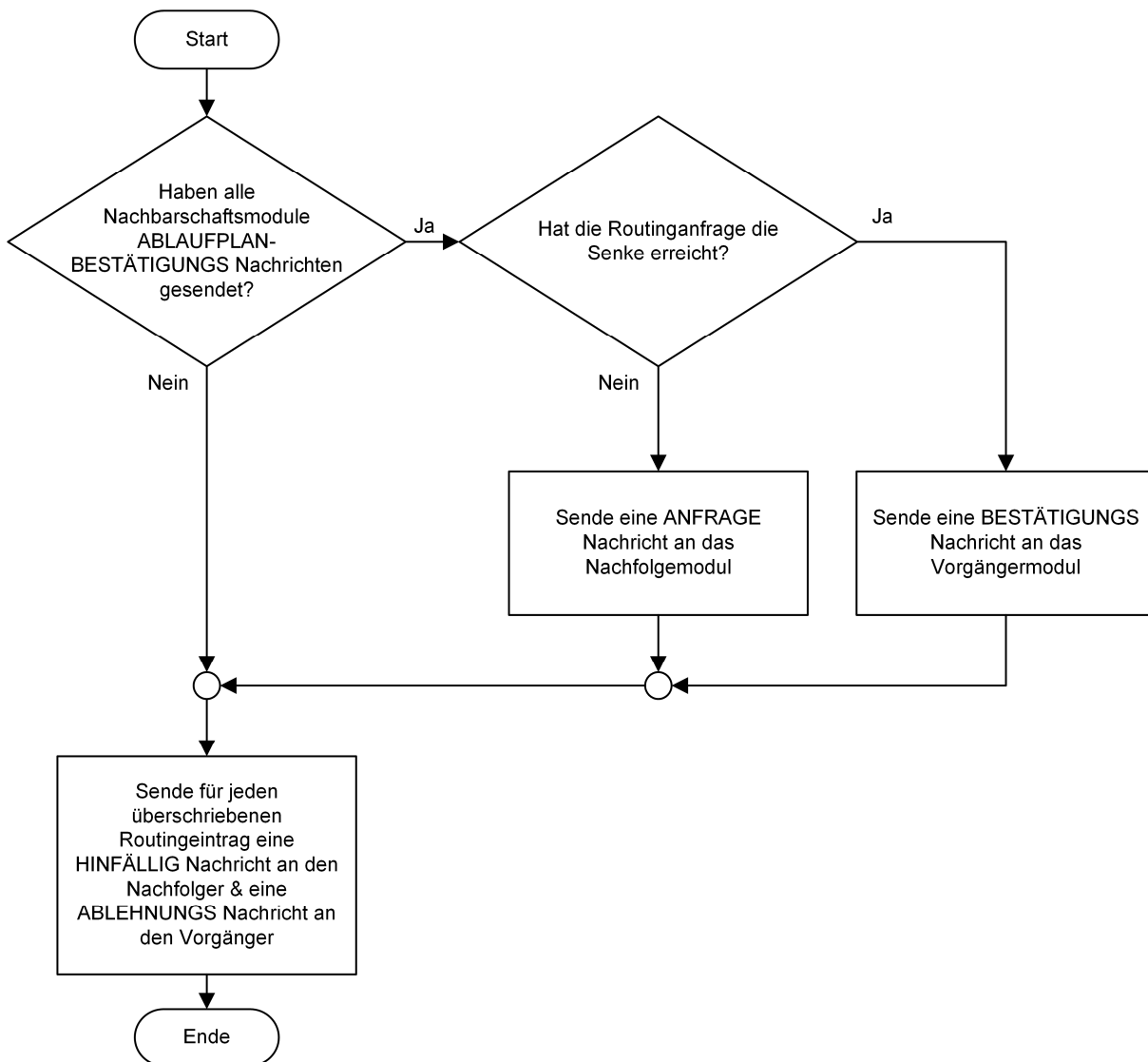


Abbildung 6.12: Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN-BESTÄTIGUNGS-Nachrichten

6.2.2 Anwendungsbeispiel

In dem Anwendungsbeispiel, welches in der Abbildung 6.13 dargestellt ist, wird das Systemverhalten gezeigt, wenn parallel zwei Hauptmodule dasselbe Nachbarschaftsmodul anfragen. Um das Anwendungsbeispiel übersichtlich zu halten, werden nur zwei Hauptmodule und zwei Nachbarschaftsmodule dargestellt.

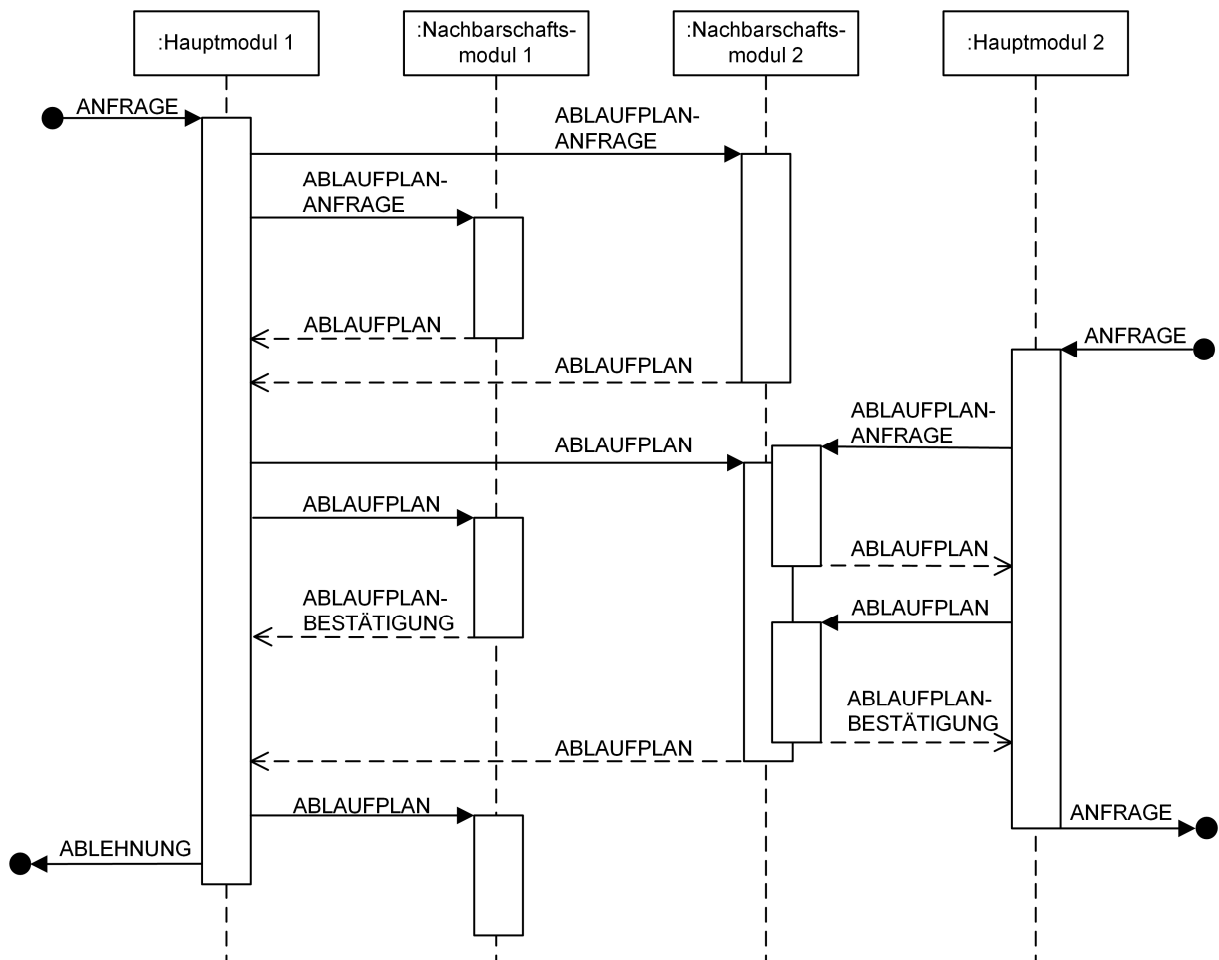


Abbildung 6.13: Sequenzdiagramm für die Verarbeitung einer ANFRAGE-Nachricht

Zur Darstellung des Anwendungsbeispiels wird ein Sequenzdiagramm verwendet, welches in dem UML Standard [ISO05] definiert ist. In Sequenzdiagrammen wird der Austausch von Nachrichten zwischen Objekten modelliert. Jedes Objekt besitzt eine vertikale Lebenslinie. Der Verlauf der Zeit in Sequenzdiagrammen wird von oben nach unten entlang dieser Lebenslinie modelliert. Prozesse, die von den Objekten ausgeführt werden, sind als unbeschriftete Rechtecke dargestellt, die auf der Lebenslinie der Objekte angeordnet werden. Die Länge der Rechtecke sagt hierbei nichts über die Dauer des Prozesses aus.

In dem ersten Schritt erhält das Hauptmodul 1 eine ANFRAGE-Nachricht, worauf es jeweils eine ABLAUFPLAN-ANFRAGE-Nachricht an die beiden benötigten Nachbarschaftsmodule sendet. Nachdem das Hauptmodul 1 die Ablaufpläne erhalten hat, wird der Routingeintrag erfolgreich in alle drei Ablaufpläne (Hauptmodule 1, Nachbarschaftsmodule 1 und 2) eingefügt. Gleichzeitig hat das Hauptmodul 2 eine ANFRAGE-Nachricht erhalten, die eine höhere Priorität aufweist als die ANFRAGE, die das Hauptmodul 1 erhalten hat. Das Hauptmodul 2 benötigt für die Durchführung des Transportvorganges das Nachbarschaftsmodul 2. Das Nachbarschaftsmodul 2 schickt nun den

Ablaufplan an das Hauptmodul 2. Das Hauptmodul 2 schließt das erfolgreiche Einfügen des Routingeintrages schneller ab als das Hauptmodul 1 und schickt den modifizierten Ablaufplan an das Nachbarschaftsmodul 2 zurück, welches den Empfang mit einer ABLAUFPLAN-BESTÄTIGUNGS-Nachricht quittiert. Anschließend schickt das Hauptmodul eine ANFRAGE-Nachricht an ein benachbartes Modul.

Das Hauptmodul 1 schickt nun nach dem Einfügen des Routingeintrages die modifizierten Ablaufpläne an die Nachbarschaftsmodule 1 und 2. Das Nachbarschaftsmodul 1 antwortet umgehend mit einer ABLAUFPLAN-BESTÄTIGUNGS-Nachricht. Da jedoch das Nachbarschaftsmodul 2 gerade damit beschäftigt ist, die Anfrage von dem Hauptmodul 2 zu bearbeiten, wird zunächst der vom Hauptmodul 1 verschickte Ablaufplan nicht verarbeitet. Da das vom Hauptmodul 1 benötigte Zeitfenster auf dem Nachbarschaftsmodul 2 nach der Verarbeitung der Anfrage des Hauptmoduls 1 nicht mehr verfügbar ist, kann keine ABLAUFPLAN-BESTÄTIGUNGS-Nachricht an das Hauptmodul 1 zurückgesendet werden. Stattdessen wird der geänderte Ablaufplan an das Hauptmodul 1 geschickt. Das Hauptmodul 1 stellt nun fest, dass der Routingeintrag nicht eingefügt werden kann. Daher wird eine ABLEHNUNGS-Nachricht an das Vorgängermodul geschickt und darüber hinaus wird dem Nachbarschaftsmodul 1 mitgeteilt, dass der bereits bestätigte Routingeintrag gelöscht werden kann.

Es ist nicht intuitiv, dass die Reservierung des Hauptmoduls 2 schneller abgeschlossen ist als die Reservierung des Hauptmoduls 1, obwohl das Hauptmodul 1 früher mit der Reservierung begonnen hat. Dieser Fall kann jedoch eintreten, wenn die Modulnachbarschaft von dem Hauptmodul 1 mehr Module enthält als die Modulnachbarschaft von dem Hauptmodul 2. Auf Grund der größeren Anzahl der Module sind mehr Kommunikationsschritte notwendig und die Reservierung dauert länger.

6.3 Analyse des Algorithmus

6.3.1 Beweis der Konfliktfreiheit

Zu den Konflikten, die auftreten können, gehören Kollisionen, Verklemmungen, Verhungern und Livelocks. Die Definitionen dieser Konflikte sind in dem Kapitel 2.1.1 dargestellt. Livelocks werden verhindert, indem bei der Suche nach einem Nachfolgemodul das Vorgängermodul ausgeschlossen wird. Eine Modifikation des Algorithmus zur Verhinderung vom Verhungern von Quellen ist im Anhang A4 beschrieben. In den meisten Layouts (siehe Kapitel 7) kann das Verhungern jedoch schon über Priorisierung von Transporteinheiten basierend auf ihrem Einschleusezeitpunkt verhindert werden.

Im Folgenden wird der Beweis erbracht, dass sowohl Kollisionen als auch Verklemmungen durch den Algorithmus verhindert werden. In Kapitel 2.1.1 wurde gezeigt, dass es nur zwei grundlegende Verklemmungssituationen in modularen Fördersystemen gibt: Entweder werden Verklemmungen erzeugt durch einander entgegengesetzte Routen oder durch Transporteinheiten, die sich in einem Kreisverkehr gegenseitig blockieren (siehe Abbildung 2.4). In beiden Fällen ist die Verklemmung aufgetreten, weil für jede Transporteinheit auf dem nachfolgenden Modul das zum weiteren Transport benötigte Zeitfenster bereits von einer anderen Transporteinheit reserviert ist. Somit werden Kollisionen und Verklemmungen vermieden, wenn sich die Zeitfenster für verschiedene Transporteinheiten niemals überlappen. Da die Zeitfenster bei der Verarbeitung von ANFRAGE-Nachrichten ausgehandelt werden, muss nur die Phase des Aushandelns betrachtet werden und nicht die Bestätigungs-Phase. Der im Folgenden dargestellte Beweis ist sowohl für kleinskalige als auch für großskalige Fördermodule gültig. Im Gegensatz zu den großskaligen Fördermodulen muss bei der Verwendung von kleinskaligen Fördermodulen beim Empfangen einer ANFRAGE-Nachricht nicht nur ein Ablaufplan auf Überlappungen überprüft werden, sondern alle Ablaufpläne innerhalb der Modulnachbarschaft (siehe Kapitel 6.2.1). Um den Beweis übersichtlich darzustellen, wird die Reservierung innerhalb der Modulnachbarschaft nicht explizit beschrieben.

Zunächst wird durch eine vollständige Fallunterscheidung der Beweis für die erste mögliche Verklemmungssituation beschrieben, bei der zwei Transporteinheiten durch entgegengesetzte Routen verklemmt sind. Die folgenden drei Fälle können auftreten:

Im ersten Fall sind beide Routingeinträge im Zustand „angefragt“. Jedes der beiden Module sendet nun eine ANFRAGE-Nachricht an das andere Modul. Beide Module erstellen auf Grund der empfangenen ANFRAGE-Nachricht einen neuen Routingeintrag, filtern ihre Ablaufpläne basierend auf der Priorität der ANFRAGE-Nachricht und versuchen anschließend, den neuen Routingeintrag in den gefilterten Ablaufplan einzufügen. Bei gleicher Priorität wird die eindeutige ID der Transporteinheiten genutzt, um zu entscheiden, welche Routinganfrage gelöscht wird. Das Modul, welches den Routingeintrag mit der niedrigeren Priorität gespeichert hatte, löscht diesen alten Routingeintrag beim Einfügen des neuen Routingeintrages auf Grund der höheren Priorität des neuen Routingeintrages. Das Modul, welches den Routingeintrag mit der höheren Priorität gespeichert hatte, fügt den neuen Routingeintrag auf Grund seiner niedrigeren Priorität nicht ein. Somit besitzen beide Routingmodule nach Verarbeitung der ANFRAGE-Nachrichten nur noch jeweils einen Routingeintrag für die Transporteinheit mit

der höheren Priorität und weder eine Kollision noch eine Verklemmung tritt beim nachfolgenden Transportvorgang auf.

Im zweiten Fall ist einer der Routingeinträge bereits im Zustand „bestätigt“, während der andere Routingeintrag im Zustand „angefragt“ ist. Nur das Modul mit dem Routingeintrag im Zustand „angefragt“ sendet eine ANFRAGE-Nachricht an das benachbarte Modul. Das Modul, welches die ANFRAGE-Nachricht empfängt, stellt fest, dass das angefragte Zeitfenster mit dem Zeitfenster des bereits vorhandenen Routingeintrages im Zustand „bestätigt“ überlappt. Da Routingeinträge im Zustand „bestätigt“ nicht durch Routingeinträge im Zustand „angefragt“ gelöscht werden können, wird eine ABLEHNUNGS-Nachricht zurückgesendet. Somit gibt es keine Überlappungen der Zeitfenster und es kommt nicht zu einer Kollision oder einer Verklemmung.

Im dritten Fall sind beide Routingeinträge bereits im Zustand „bestätigt“. Dieser Fall kann nicht auftreten und somit kann auch keine Kollision oder Verklemmung auftreten. Falls beide Routingeinträge im Zustand „angefragt“ gewesen wären, hätten die Module die Regeln des ersten Falles angewandt und somit wäre danach nur auf einem der beiden Module ein Routingeintrag im Zustand „bestätigt“ vorhanden. Falls einer der beiden Routingeinträge im Zustand „bestätigt“ und der andere Routingeintrag im Zustand „angefragt“ gewesen wäre, hätten die Module die Regeln des zweiten Falles angewendet und somit wäre es nicht zu einer Kollision oder einer Verklemmung gekommen.

Die zweite mögliche Verklemmungssituation kann durch zyklisch abhängige Routen hervorgerufen werden. Im Folgenden wird durch eine vollständige Fallunterscheidung bewiesen, dass es auch hierbei zu keiner Verklemmung oder Kollision kommen kann. Hierzu werden die zwei möglichen Fälle untersucht, die auftreten können, wenn ein Modul, welches sich bereits in einer zyklischen Abhängigkeit mit anderen Modulen befindet, eine Nachricht von außerhalb der zyklischen Abhängigkeit erhält.

Im ersten Fall besitzt das zyklisch abhängige Modul einen Routingeintrag im Zustand „bestätigt“, wenn es eine ANFRAGE-Nachricht von einem Modul erhält, welches bisher noch keine zyklische Abhängigkeit aufweist. Das zyklisch abhängige Modul sendet als Antwort auf die ANFRAGE-Nachricht eine ABHLEHNUNGS-Nachricht zurück, da Routingeinträge im Zustand „bestätigt“ durch ANFRAGE-Nachrichten nicht mehr überschrieben werden können und somit kommt es nicht zu einer Verklemmung oder einer Kollision.

Im zweiten Fall besitzt das zyklisch abhängige Modul einen Routingeintrag im Zustand „angefragt“, wenn es eine ANFRAGE-Nachricht erhält. Durch den Vergleich der Prioritäten wird entweder durch den neu erstellen Routingeintrag der bereits vorhandene

Routingseintrag gelöscht oder der bereits vorhandene Routingseintrag bleibt bestehen und es wird eine ABHLEHNUNGS-Nachricht an das anfragende Modul zurückgeschickt und somit kommt es nicht zu einer Verklemmung oder einer Kollision.

6.3.2 Laufzeitanalyse

Für die Laufzeitanalyse wird die Landau-Notation verwendet, die in [Lan09] beschrieben ist. Mit der Landau-Notation lässt sich die asymptotische Laufzeit bzw. die Komplexitätsklasse eines Algorithmus bestimmen. Die asymptotische Laufzeit ist hierbei unabhängig von der Implementierung, der verwendeten Hardware und der Eingabemenge. Anstatt also mehrere Messungen mit verschiedenen Eingabemengen auf verschiedenen Hardwarekonfigurationen durchzuführen, können mit Hilfe der Landau-Notation viel allgemeinere Aussagen getroffen werden, die auch noch für zukünftige Hardwarekonfigurationen gelten.

Ein Beispiel zur Veranschaulichung: Wenn die Laufzeit eines Algorithmus f linear wächst mit der Eingabemenge n , wird dies mit der Landau-Notation folgendermaßen angegeben:

$$f \in O(n) \tag{6.3}$$

Die Bestimmung der asymptotischen Laufzeit hat eine große praktische Relevanz, da hierdurch eine Aussage über die Skalierbarkeit des Algorithmus getroffen werden kann. Ein Algorithmus, der wie in dem obigen Beispiel über eine lineare asymptotische Laufzeit verfügt, lässt sich mit heutigen Computern fast immer beenden. Erst bei extrem großen Eingabemengen können die heutigen Computer die Ausführung in einer nicht vertretbaren Zeit abschließen. Algorithmen mit einer linearen oder sogar einer noch geringeren asymptotischen Laufzeitzeit gelten als skalierbar, da ihre Laufzeit lediglich proportional zu der Eingabemenge ansteigt.

Sowohl für den Algorithmus für die großskaligen Module als auch für die Modifikation des Algorithmus für die kleinskaligen Module wird eine Laufzeitanalyse durchgeführt. Die Laufzeit der Algorithmen ist abhängig von der Anzahl der empfangenen Nachrichten. Die Verarbeitungszeit von HINFÄLLIG- und BESTÄTIGUNGS-Nachrichten ist bei dem Algorithmus für großskalige Fördermodule konstant, da jeweils nur ein einziger Routingseintrag verändert werden muss.

Im Gegensatz dazu ist die Verarbeitungszeit von ABLEHNUNGS- und ANFRAGE-Nachrichten nicht konstant, da ein neu erstellter Routingseintrag in den Ablaufplan des Moduls eingefügt werden muss. Im ungünstigsten Fall muss beim Einfügen ein Ver-

gleich mit jedem bereits vorhandenen Routingeintrag erfolgen. Um Livelocks zu verhindern, dürfen Module keine ANFRAGE-Nachrichten an Vorgängermodule senden. Somit können im Ablaufplan eines einzigen Moduls maximal so viele Routingeinträge vorhanden sein, wie es Transporteinheiten im Fördersystem gibt. Die Laufzeit zur Verarbeitung der Nachrichten ist daher:

$$f \in O(m_o + m_{rd} \cdot N_{tot}) \quad (6.4)$$

Hierbei ist m_o die Anzahl der empfangenen HINFÄLLIG- und BESTÄTIGUNGS-Nachrichten, m_{rd} ist die Anzahl der empfangenen ABLEHNUNGS- und ANFRAGE-Nachrichten und N_{tot} ist die Anzahl der Transporteinheiten im Fördersystem.

Die Laufzeit des Algorithmus der kleinskaligen Fördermodule ist länger als die Laufzeit für großskalige Module, da mehr Nachrichten geschickt und verarbeitet werden müssen, um alle Module innerhalb einer Modulnachbarschaft zu koordinieren. Das Verarbeiten von ABLAUFPLAN-ANFRAGE-, ABLAUFPLAN-, und ABLAUFPLAN-BESTÄTIGUNGS-Nachrichten ist konstant. Im Gegensatz zu den großskaligen Fördermodulen ist jedoch die Verarbeitung von HINFÄLLIG- und BESTÄTIGUNGS-Nachrichten nicht mehr konstant, da das Hauptmodul diese Nachrichten an alle Module innerhalb der Nachbarschaft weiterleiten muss. Bei der Verarbeitung von ABLEHNUNGS- und ANFRAGE-Nachrichten müssen die Hauptmodule sowohl den dazugehörigen Routingeintrag auf ihrem eigenen Ablaufplan einfügen als auch auf dem Ablaufplan aller Nachbarschaftsmodul. Somit ergibt sich die folgende Laufzeit:

$$f \in O(m_s + n_m \cdot (m_o + m_{rd} \cdot N_{tot})) \quad (6.5)$$

Hierbei ist m_s die Anzahl der empfangenen ABLAUFPLAN-ANFRAGE-, ABLAUFPLAN- und ABLAUFPLAN-BESTÄTIGUNGS-Nachrichten, n_m ist die Anzahl der Module in einer Modulnachbarschaft, m_o ist die Anzahl der empfangenen HINFÄLLIG- und BESTÄTIGUNGS-Nachrichten, m_{rd} ist die Anzahl der empfangenen ABLEHNUNGS- und ANFRAGE-Nachrichten und N_{tot} ist die Anzahl der Transporteinheiten im Fördersystem.

Da der Algorithmus und seine Modifikation eine lineare asymptotische Laufzeit aufweisen, sind sie skalierbar.

6.3.3 Verhalten im Fehlerfall

Sowohl der Beweis der Konfliktfreiheit als auch die Laufzeitanalyse wurde für ein idealisiertes Modell der Fördermodule (siehe Kapitel 5.3) durchgeführt. Jedoch ist es wissenschaftlich auch interessant zu untersuchen, welche Fehler in einem realen System auftreten können, wie sich diese auf das Systemverhalten auswirken, wie sie detektiert

und gegebenenfalls sogar korrigiert werden können. Es ist Stand der Wissenschaft und Technik eine systematische Fehleranalyse mit Hilfe der FMEA-Methode (Engl.: „Failure Mode and Effects Analysis“) zu erstellen. Bei der FMEA wird für jeden möglichen Fehler die Schwere, die Auftretenswahrscheinlichkeit und die Entdeckungswahrscheinlichkeit ermittelt. Anschließend wird für jeden möglichen Fehler überprüft, wie dieser verhindert werden kann und entsprechenden Maßnahmen ergriffen. [Tie20]

Im Rahmen dieser Arbeit wird keine vollständige FMEA durchgeführt, da viele mögliche Fehler und deren Behebung wissenschaftlich nicht interessant sind (z.B. Schaltschrank einer Fördermatrix wurde nicht richtig angeschlossen). Stattdessen werden die zwei wichtigsten und wissenschaftlich interessantesten Fehlerklassen untersucht: Ein Modul verhält sich fehlerhaft oder ein Transportprozess wird fehlerhaft durchgeführt.

Zu dem fehlerhaften Ausführen des Transportvorganges zählt z.B. eine unbeabsichtigte Verdrehung oder das Hängenbleiben einer Transporteinheit. Das Auftreten dieser Fehler wurde vom Autor dieser Arbeit an konventionellen Fördersystemen beobachtet. Bei konventionellen Fördersystemen werden fehlerhaft ausgeführte Transportvorgänge durch Kontrollstationen (z.B. optische Scanner) detektiert, indem festgestellt wird, dass eine Transporteinheit zwar die vorherige Kontrollstation durchlaufen hat, aber nicht mehr die nachfolgenden. Da konventionelle Fördersysteme häufig nur in eine Richtung fördern können, besitzen diese keine Möglichkeit die hängengebliebenen Transporteinheiten durch eine kurzzeitige Umkehr der Förderrichtung zu lösen. Daher wird der Fehler zwar vom System detektiert, aber die Korrektur muss durch einen menschlichen Eingriff erfolgen. Der Mensch muss hierbei zur Behebung des Fehlers die gesamte Strecke zwischen den beiden Kontrollstationen untersuchen.

Da die modularen Fördersysteme über eine größere Anzahl an Sensoren verfügen als konventionelle Fördersysteme, können fehlerhafte Transportvorgänge schneller detektiert werden. Durch die Fähigkeit der modularen Fördersysteme in mehrere Richtungen fördern zu können, sind diese auch in der Lage, einmal verkantete Transporteinheiten wieder zu lösen. Daher wird die Notwendigkeit von Eingriffen durch Menschen verringert. Falls es zu einem Fehler beim Transportvorgang gekommen ist, können die vorher ausgehandelten Zeitfenster entlang der Route nicht mehr erreicht werden. Wie im Kapitel 6.1.2 beschrieben ist, werden durch den Routingalgorithmus selbst bei einer Differenz zwischen geplantem und tatsächlichem Zeitfenster Konflikte verhindert. Weiterhin können die modularen Fördersysteme um eine Regelung erweitert werden (siehe Anhang A2), um fehlerhafte Transportvorgänge gar nicht erst entstehen zu lassen.

Fehler der zweiten Fehlerklasse treten auf, wenn ein einzelnes Modul nicht mehr korrekt funktioniert. Hierbei wird zwischen zwei Fällen unterschieden: Zum einen ist es möglich, dass ein Modul gar nicht mehr funktioniert und zum anderen ist es möglich, dass ein Modul nur noch teilweise funktioniert. Falls ein Modul gar nicht mehr funktioniert, kann eine Fehlererkennung nur von einem benachbarten Modul erfolgen. Eine Möglichkeit besteht darin, dass in regelmäßigen Abständen alle Module an ihre Nachbarmodule Ping-Nachrichten senden. Falls die Nachbarmodule zu langsam antworten, geht der Sender davon aus, dass das Modul nicht mehr funktionsfähig ist. In diesem Fall benachrichtigt das noch funktionsfähige Modul einen Menschen, um eine Reparatur oder einen Austausch zu veranlassen. Falls in dem Layout Alternativrouten zum Ziel vorhanden sind, können Transporteinheiten das nicht mehr funktionsfähige Fördermodul umfahren.

Wenn ein Modul nur noch teilweise funktioniert, dann kann dies ebenfalls durch die Nachbarmodule festgestellt werden. Fehler können sowohl auf der Hardwareebene (einzelne Motoren funktionieren nicht mehr) als auch auf der Softwareebene auftreten (fehlerhafte zeitliche Synchronisierung). Im Falle von Fehlern in der Hardware können die Nachbarmodule den Fehler erst feststellen, wenn der Transportvorgang nicht mehr korrekt ausgeführt wird. Im Falle von Fehlern in der Software, können die Module zusätzlich zu den Ping-Nachrichten noch Statusinformationen von sich selbst mitschicken, wozu z.B. die eigene, lokale Zeitbasis gehört. Wenn nun ein Nachbarmodul solch eine Statusmeldung erhält und feststellt, dass der Nachbar z.B. eine größere zeitliche Abweichung aufweist, als im System zulässig ist, kann wiederum eine Fehlermeldung an einen Menschen gesendet werden, der dann das Modul näher untersuchen muss, um die volle Funktionsfähigkeit wiederherzustellen.

Kapitel 7

Simulationsgestützte Untersuchung des Systemverhaltens

7.1 Versuchsplanung

Im Kapitel 5.3 wurden die verwendeten Modelle sowohl für die großskaligen als auch die kleinskaligen Fördermodule beschrieben. Das Modell für die großskaligen Fördermodule basiert auf den FlexConveyor-Modulen und besitzt Abmessungen von 500 x 500 mm. Die FlexConveyor-Module werden ausgewählt, da in [Sei16b] von SEIBOLD bereits Leistungskenngrößen für Layouts mit FlexConveyor-Modulen ermittelt worden sind. Die Ergebnisse dieser Arbeit und die Ergebnisse von SEIBOLD können daher miteinander verglichen werden.

Die Simulation wird so eingestellt, dass ein Simulationsschritt 1 ms in der Realität entspricht. Somit beträgt die Messauflösung $\pm 0,5$ ms. Weiterhin wird festgelegt, dass die Übertragung einer Nachricht von einem großskaligen Modul zu einem benachbarten großskaligen Modul 1 ms dauert und die Fördergeschwindigkeit der großskaligen Module wird auf 2 m/s festgelegt. Da die Abmessungen eines Modules 500 x 500 mm sind, dauert der Transport einer Transporteinheit von einem Modul zu einem benachbarten Modul 250 ms. Für die großskaligen Module werden alle Transporteinheiten mit einer einheitlichen Größe modelliert (480 x 480 mm), da es aus steuerungstechnischer Perspektive irrelevant ist, ob eine Transporteinheit ein großskaliges Modul vollständig oder nur zum Teil belegt.

Das Modell der kleinskaligen Fördermodule basiert auf dem netkoPs-Referenzfördersystem. Eine Fördermatrix des netkoPs-Referenzfördersystems verfügt über 36 Module. Die Module nehmen hierbei eine Fläche von 500 x 500 mm ein. Die Fördermodule werden mit einer matrixbasierten Ansteuerung modelliert und somit gehören alle

36 Module einer Matrix zu einer statischen Modulnachbarschaft, die von einem Hauptmodul kontrolliert wird. Da alle Module einer statischen Modulnachbarschaft immer die gleiche Aktion ausführen (siehe Kapitel 5.3), ist es auch bei den kleinskaligen Fördermodulen aus steuerungstechnischer Perspektive irrelevant, ob eine Transporteinheit alle Module der Nachbarschaft belegt oder nur einen Teil davon. In dem Kapitel 6.2 wurde ein Algorithmus für kleinskalige Fördermodule mit einer modulbasierten Ansteuerung beschrieben. Der Vorteil der modulbasierten Ansteuerung ist, dass die zur Verfügung stehenden Module effektiver genutzt werden auf Grund der dynamisch gebildeten Modulnachbarschaften, deren Größe an die bevorstehende Aufgabe angepasst ist. Im Gegensatz zu der matrixbasierten Ansteuerung werden bei der modulbasierten Ansteuerung die Leistungskenngrößen durch die Größe der Transporteinheiten (bzw. das Größenverhältnis zwischen den Transporteinheiten und den Modulen) beeinflusst. Der Einfluss der Größe der Transporteinheiten bei kleinskaligen Fördermodulen mit modulbasierter Ansteuerung wurde im Rahmen dieser Arbeit nicht simulationsgestützt untersucht. Jedoch wurde im Anhang A5 der Einfluss der Größe der Transporteinheiten theoretisch betrachtet.

Die Simulation wird so eingestellt, dass die Übertragung einer Nachricht zwischen einer Fördermatrix zu einer benachbarten Fördermatrix 1 ms dauert und die Fördergeschwindigkeit auf 2 m/s festgelegt. Somit dauert der Transport einer Transporteinheit von einer Fördermatrix zu einer benachbarten Fördermatrix 250 ms. Die Modelle der großskaligen Fördermodule und der kleinskaligen Fördermodule mit einer matrixbasierten Ansteuerung sind somit aus steuerungstechnischer Perspektive identisch. Innerhalb dieses Kapitels wird daher nicht mehr zwischen großskaligen Modulen und kleinskaligen Modulen mit matrixbasierter Ansteuerung unterschieden und diese werden vereinfacht nur noch als Modul bezeichnet.

Der nächste Schritt in der Versuchsplanung besteht darin festzulegen, welche Layouts untersucht werden müssen. Da der entwickelte Algorithmus in der Lage ist, in jedem beliebigen Layout Routen zu reservieren, muss eine Auswahl getroffen werden. Die Priorität wird hier auf die für den späteren Praxiseinsatz relevantesten Layouts gelegt. Um diese Priorität festzulegen, müssen Auswahlkriterien aufgestellt werden. Diese Auswahlkriterien leiten sich von den Hardwareeigenschaften der Module ab. Die Module sind auf Grund der komplexeren Technik teurer als konventionelle Fördersysteme, besitzen aber gleichzeitig einen höheren Funktionsumfang. So sind alle im Rahmen dieser Arbeit vorgestellten Module in der Lage, in mindestens vier Richtungen zu fördern. Aus diesen besonderen Eigenschaften der Fördermodule leiten sich die folgenden Auswahlkriterien ab: Fördermodule sollten in quadratischen oder rechteckigen

Layouts eingesetzt werden, da hier ihre Fähigkeit in alle Richtungen zu fördern vorteilhaft genutzt werden kann. Folgerichtig ist somit ein Einsatz von Fördermodule nicht sinnvoll, wenn nur entlang einer Richtung gefördert werden kann. Daher macht es ökonomisch keinen Sinn z.B. Förderbänder durch Fördermodule zu ersetzen. Weiterhin spricht für die Verwendung der Fördermodule in rechteckigen oder quadratischen Layouts, dass in diesen Layouts die Module höher ausgelastet werden können. Auf Grund der höheren Herstellungskosten sollten die Fördermodule höher ausgelastet werden, um sich schneller zu amortisieren. Im Rahmen dieser Arbeit werden daher nur quadratische oder rechteckige Layouts untersucht.

Die Layouts werden in zwei verschiedene Klassen unterteilt: Sorter- und Kreuzungslayouts. Vorangegangene Forschungsarbeiten haben ebenfalls den Einsatz von modularen Fördersystemen in Sorter- und Kreuzungslayouts untersucht [Krü15, Sei16a]. Sorterlayouts besitzen nur an einem Rand Quellen für Transporteinheiten. An allen anderen Rändern werden Senken angeordnet. Im Gegensatz hierzu können bei den Kreuzungslayouts die Quellen und Senken beliebig an den Rändern des Layouts angeordnet werden.

Nachdem sowohl die Form der zu untersuchenden Layouts (quadratisch und rechteckig) als auch die Anordnung der Senken und Quellen festgelegt wurde, muss nun die Größe der zu untersuchenden Layouts definiert werden. In jedem Layout, in dem Quellen und Senken nur an den Rändern zugelassen sind, stellen Quellen und Senken Engpässe dar. Dies liegt darin begründet, dass die Gesamtanzahl der Module bei einer Vergrößerung des Layouts quadratisch ansteigt, während die Anzahl der Quellen und Senken lediglich linear ansteigt. Somit sinkt die Auslastung der Module und der Betrieb der Anlage wird unwirtschaftlicher. Aus diesem Grund wird die maximale Anzahl der Module eines Layouts auf 256 begrenzt.

In dem folgenden Kapitel werden insgesamt 17 verschiedene Layouts simuliert. Hierbei werden für die einzelnen Layouts zusätzlich noch die Steuerungsparameter variiert. In sieben Layouts werden hierbei Kreuzungen simuliert, während in zehn Layouts Sorter simuliert werden. Drei der simulierten Sorter-Layouts wurden hierbei schon von SEIBOLD in [Sei16a] simuliert. Somit sind die Ergebnisse dieser drei Layouts vergleichbar mit den Ergebnissen von SEIBOLD. Die Resultate für zwei Sorter-Layouts und für ein Kreuzungs-Layout werden im Hauptteil dieses Kapitels detaillierter dargestellt. Die Ergebnisse aller übrigen Layouts sind in den Anhängen A6 bis A8 näher beschrieben. Einen Überblick über die simulierten Layouts liefert die Tabelle 7.1.

Tabelle 7.1: Übersicht über alle simulierten Layouts

Name des Layouts	Anzahl der Module	Anzahl der Quellen	Anzahl der Senken
Inter_3x3	9	6	6
Inter_4x4	16	8	8
Inter_6x6	36	6	18
Inter_8x8	64	16	16
Inter_12x12	144	12	36
Inter_14x14	196	14	42
Inter_16x16	256	32	32
Sorter_3x3	9	6	6
Sorter_4x4	16	8	8
Sorter_6x6	36	6	18
Sorter_8x8	64	16	16
Sorter_12x12	144	12	36
Sorter_14x14	196	14	42
Sorter_16x16	256	32	32
Sorter_23x11x2	253	2	43
Sorter_23x11x6	253	6	43
Sorter_23x11x10	253	10	43

Die Untersuchung des Steuerungsalgorithmus geschieht durch Simulationsstudien, Simulationsexperimente und Simulationsläufe. Eine Simulationsstudie hat zum Ziel, den Einfluss eines einstellbaren Parameters auf die logistischen Zielgrößen zu ermitteln. Um diesen Einfluss zu ermitteln, werden mehrere Simulationsexperimente durchgeführt. Die Simulationsexperimente unterscheiden sich voneinander dadurch, dass einer der einstellbaren Parameter verändert wird. Jedes Simulationsexperiment setzt sich aus mehreren Simulationsläufen zusammen. Bei einem Simulationslauf werden hierbei die einstellbaren Parameter nicht verändert, sondern nur der Startwert (Engl.

seed) für den Zufallszahlengenerator (siehe Kapitel 5.4 für eine Beschreibung des Zufallszahlengenerators). Durch die Durchführung von mehreren Simulationsläufen können die Ergebnisse der einzelnen Simulationsexperimente statistisch abgesichert werden. Insgesamt werden für jedes Simulationsexperiment 20 Simulationsläufe durchgeführt. Da die gemessenen Größen alle normalverteilt um den Mittelwert streuen, kann die Standardabweichung mit der folgenden Formel berechnet werden:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (7.1)$$

Hierbei ist n die Anzahl der Simulationsläufe, x_i das Ergebnis eines einzelnen Simulationslaufs und \bar{x} der Mittelwert. Anschließend wird mit Hilfe der Standardabweichung der Standardfehler folgendermaßen berechnet:

$$SE = \frac{s}{\sqrt{n}} \quad (7.2)$$

7.2 Systemverhalten von quadratischen Sortern

In diesem Kapitel wird das Systemverhalten von Sortern mit Hilfe des in Abbildung 7.1 dargestellten Layouts vorgestellt. Bei diesem Layout handelt es sich um das kleinste untersuchte Sorterlayout, welches hier verwendet wird, da es sich am kompaktesten darstellen lässt. Die Untersuchungen der größeren Layouts sind im Anhang A6 dokumentiert.

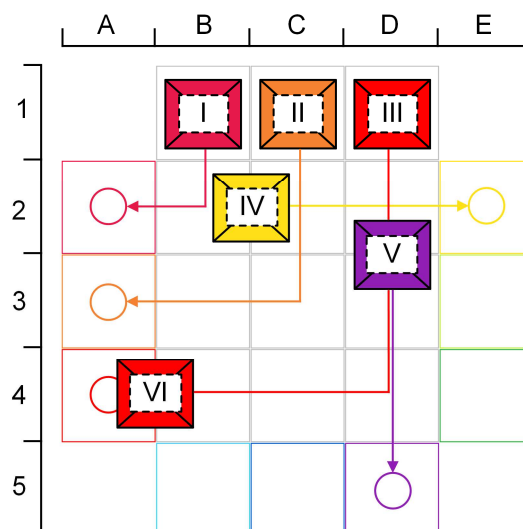


Abbildung 7.1: Layout „Sorter_3x3“

Das Layout „Sorter_3x3“ besteht hierbei aus neun Fördermatrizen, um welche die Quellen und Senken angeordnet sind. Die drei Quellen sind hierbei in der Zeile 1 angeordnet und die Senken sind in den Spalten A und E und in der Zeile 5 angeordnet.

Jede Senke wird mit einer eigenen Farbe markiert. Transporteinheiten, die zu dieser Senke transportiert werden müssen, erhalten die gleiche Farbe wie die entsprechende Senke. Jede Farbe wird hierbei nur einmal vergeben. Die Quellen werden so eingestellt, dass so viele Transporteinheiten eingeschleust werden, wie es physisch möglich ist. Dies bedeutet, dass sobald auf einer Quelle keine Transporteinheit mehr ist, sofort eine neue Transporteinheit eingeschleust wird. Die Quellen werden so eingestellt, weil dies zu der höchsten Auslastung der Fördermatrizen führt und somit zum wirtschaftlich sinnvollsten Betrieb.

Zunächst wird der Einfluss des Sicherheitsfaktors n_s (siehe Formel 6.1) auf den Durchsatz untersucht. Zur Erinnerung: Durch eine Vergrößerung des Sicherheitsfaktors wird die Dauer der Reservierungsanfragen verlängert, sodass es wahrscheinlicher wird, dass auch bei einem temporär erhöhten Aufkommen an Transporteinheiten die Reservierung erfolgreich abgeschlossen wird. n_s hat einen Einfluss auf die Reservierungszeit, die wiederum einen Einfluss auf den Durchsatz hat. Nachdem der optimale Wert für n_s ermittelt wurden, werden die modulbezogenen Kenngrößen (relativer Durchsatz und relative Belegung) und die transporteinheitenbezogenen Kenngrößen genauer dargestellt. Die transporteinheitenbezogenen Kenngrößen sind der Mittelwert der Durchlaufzeit und die dazugehörige Standardabweichung.

Für jeden Wert von n_s wurden 20 Simulationsläufe durchgeführt, um die Standardabweichung zu berechnen. Die Ergebnisse sind in der Tabelle 7.2 dargestellt. Der Startwert für n_s wird hierbei auf 1,0 festgelegt, da dies bedeutet, dass eine neue Routenreservierung genauso viel Zeit benötigt, um abgeschlossen zu werden, wie vorherige Routenreservierungen im Mittel benötigt haben. Der maximale Wert für n_s wird zunächst auf 3,0 festgelegt. Es wird erwartet, dass der Durchsatz des Fördersystems bei einem Wert um 1,5 am größten ist, da Routenreservierungen etwas mehr Zeit zur Verfügung haben als vorherigen Routenreservierungen im Mittel benötigt hatten und somit auch bei einer temporär höheren Auslastung Routenreservierungen noch erfolgreich abgeschlossen werden können.

Tabelle 7.2: Darstellung des Durchsatzes auf dem Layout „Sorter_3x3“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	8612	8624	8615	8611	8621
Standardabweichung in TE/h	56	86	62	110	68

Aus der Tabelle ist zu erkennen, dass der Durchsatz, wie vorher erwartet, tatsächlich am höchsten bei einem Wert von 1,5 ist. Jedoch ist auch zu erkennen, dass die Ergebnisse insgesamt sehr nahe beieinanderliegen und somit der Unterschied als nicht signifikant betrachtet wird. n_s hat also, wenn überhaupt, nur einen sehr kleinen Einfluss auf den Durchsatz. Im Anhang A6 sind die Tabellen für die anderen untersuchten Sorterlayouts dargestellt. Auch bei den anderen Layouts ist zu erkennen, dass die Ergebnisse sehr nahe beieinanderliegen und somit n_s insgesamt wenig oder gar keinen Einfluss auf den Durchsatz besitzt. Hierfür wurden die folgenden zwei Gründe ermittelt:

Erstens hat n_s nur einen Einfluss auf die Dauer des Reservierungsprozesses und nicht auf die Dauer des darauffolgenden Transportprozesses. Dies hat zu Folge, dass selbst wenn der Reservierungsprozess deutlich länger dauert als bei vergangenen Reservierungsprozessen, der prozentuale Anstieg der Durchlaufzeit nur gering ist. Der prozentuale Zeitanteil des Reservierungsprozesses an der Durchlaufzeit nimmt hierbei ab, umso länger die Route wird.

Zur Veranschaulichung wird die Durchlaufzeit für den Transport einer Transporteinheit von der Quelle B1 zu der Senke B5 in der Tabelle 7.3 dargestellt. Hierbei werden zwei Fälle unterschieden: Im ersten Fall erfolgt die Routenreservierung schnellstmöglich (also ohne Auftreten einer einzigen ABLEHNUNGS-Nachricht) und im zweiten Fall dauert der Reservierungsvorgang 10 Mal länger. Die Zeiten werden folgendermaßen berechnet: Die Reservierungsnachricht muss zunächst von der Quelle zu der Senke gesendet werden. Die direkte Route ist hierbei: B1→B2→B3→B4→B5. Die Bestätigung der Routenanfrage erfolgt in umgekehrter Richtung. Da das Senden und Empfangen jeder Nachricht 1 ms dauert, beträgt die Zeit für den erfolgreichen Abschluss der Reservierung im günstigsten Fall 8 ms. In beiden Fällen (also im bestmöglichen Fall und im ungünstigeren Fall) erfolgt der darauffolgende Transportprozess schnellstmöglich (also ohne dass die Transporteinheit stoppt). Der Transport einer Transporteinheit von einem Modul zu einem benachbarten Modul dauert 250 ms. Somit dauert der Transportprozess von der Quelle B1 zu der Senke B5 1000 ms.

Tabelle 7.3: Durchlaufzeit im günstigsten Fall und in einem ungünstigeren Fall von der Quelle B1 zu der Senke B5 im Layout „Sorter_3x3“

	Bestmöglicher Fall	Ungünstiger Fall
Reservierungszeit in ms	8	80
Transportzeit in ms	1000	1000
Durchlaufzeit in ms	1008	1080
Relativer Anteil der Reservierungszeit	0,8 %	7,4 %

Anhand der Tabelle ist zu erkennen, dass der prozentuale Anstieg der Durchlaufzeit zwischen dem bestmöglichen Fall und dem ungünstigen Fall nur 6,6 % beträgt. Da der Anstieg der Reservierungszeit nur einen geringen Anteil auf den Durchlaufzeit hat, hat auch n_s nur einen geringen Einfluss auf die Durchlaufzeit.

Der zweite Grund, warum n_s einen geringen Einfluss auf die Durchlaufzeit besitzt, liegt darin begründet, dass der Algorithmus alle Entscheidungen basierend auf lokalen Informationen trifft und es somit möglich ist, dass die global beste Lösung gar nicht oder erst verspätet gefunden wird. Dies wird an dem in Abbildung 7.2 dargestellten Beispiel verdeutlicht. In dem Beispiel ist die Transporteinheit II kurz nach der Transporteinheit I eingeschleust worden. Beide Transporteinheiten haben als Ziel die Senke A2. Zu dem dargestellten Zeitpunkt ist gerade erfolgreich die Route für die Transporteinheit I reserviert worden, während die Routenreservierung für die Transporteinheit II im aktuellen Systemzustand nicht abgeschlossen ist.

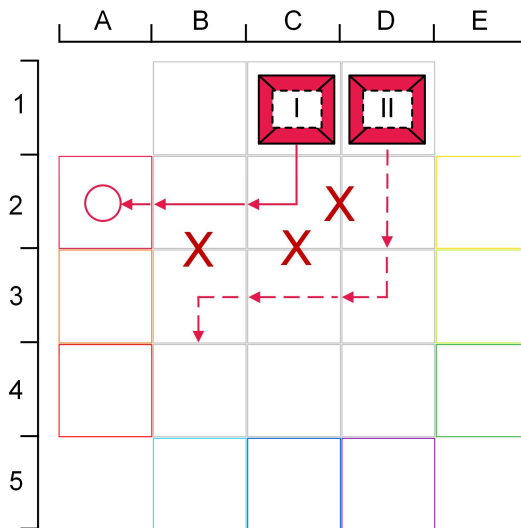


Abbildung 7.2: Die Routenreservierung entfernt sich von der Senke

Die Routinganfrage für die Transporteinheit II wurde zunächst von dem Modul D1 zu dem Modul D2 gesendet. Das Modul D2 hat nun eine Anfrage an das Modul C2 gesendet, da dieses die geringste Distanz zu der Senke A2 aufweist. Das angefragte Zeitfenster auf dem Modul C2 wurde jedoch bereits von der Transporteinheit I reserviert, sodass eine ABLEHNUNGS-Nachricht an das Modul D2 gesendet wird. Abgelehnte ANFRAGE-Nachrichten für die Transporteinheit I werden hierbei mit einem roten X dargestellt. Die Routenreservierung wurde weiter fortgesetzt, jedoch sendete das Modul C2 zum zweiten Mal eine ABHLEHNUNGS-Nachricht und das Modul B2 ebenfalls eine ABLEHNUNGS-Nachricht auf Grund einer Überschneidung der Zeitfenster mit der Transporteinheit I. Das Zeitfenster, welches das Modul B2 dem Modul B3 durch

die ABLEHNUNGS-Nachricht mitgeteilt hat, liegt so weit in der Zukunft, dass das Modul B3 auf Grund seiner lokalen Informationen davon ausgeht, dass die Senke A1 schneller über das Modul B4 zu erreichen ist. Als Folge hiervon wird eine ANFRAGE-Nachricht an das Modul B4 gesendet. Mit dem Senden dieser ANFRAGE-Nachricht ist der erfolgreiche Abschluss der Routenreservierung in diesem Iterationsschritt unwahrscheinlich geworden. Dies hängt mit zwei Eigenschaften des Algorithmus zusammen: Die erste Eigenschaft ist, dass Module bereits vorhandene Routingeinträge erst dann löschen oder modifizieren, wenn sie eine ABLEHNUNGS-Nachricht von dem nachfolgenden Modul erhalten haben. Die zweite Eigenschaft ist, dass Module Routinganfragen für Transporteinheiten, für die sie bereits einen Routingeintrag besitzen, immer ablehnen, um Livelocks zu verhindern.

In dem dargestellten Beispiel führt dies dazu, dass als nächstes Routingeinträge auf dem Modulen B4, C4 und D4 erstellt werden. Da das Modul D4 an kein benachbartes Modul eine ANFRAGE-Nachricht senden kann, sendet es eine ABLEHNUNGS-Nachricht an das Modul C4, welches wiederum eine ABLEHNUNGS-Nachricht an B4 sendet. Wenn die ABLEHNUNGS-Nachricht das Modul B3 erreicht, ist es möglich, dass das Modul B3 eine ANFRAGE-Nachricht an das Modul B2 sendet und die Routenreservierung doch noch erfolgreich abgeschlossen werden kann. Die in diesem Beispiel beschriebene Routenreservierung für die Transporteinheit II dauert auf Grund des Vorhandenseins von den Routingeinträgen der Transporteinheit I wesentlich länger. Der Wert von n_s hat in diesem Beispiel keinen großen Einfluss auf die Länge der Routenreservierung. Für die nachfolgenden Untersuchungen wird der Wert für n_s immer auf 1,5 festgelegt.

In der Abbildung 7.3 ist der Belegungsgrad und der relative Durchsatz (siehe Kapitel 2.1.2) für das Layout „Sorter_3x3“ für einen typischen Simulationslauf dargestellt. Der Belegungsgrad jeder einzelnen Quelle ist 100%, da sofort eine neue Transporteinheit eingeschleust wird, wenn eine Quelle frei ist. Weiterhin beträgt der relative Durchsatz aller drei Quellen zusammen 100%, da alle Transporteinheiten über eine der drei Quellen eingeschleust werden müssen.

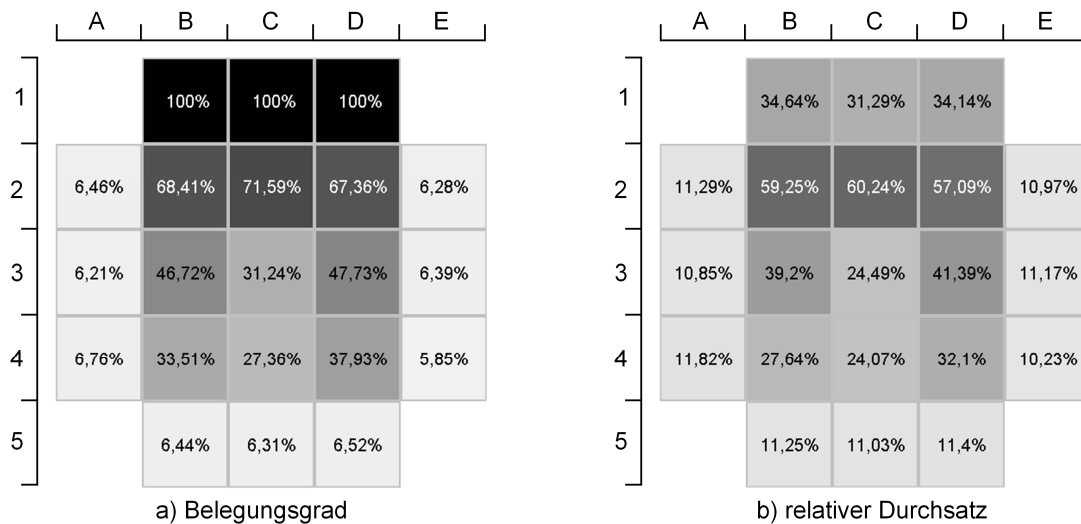


Abbildung 7.3: Belegungsgrad und relativer Durchsatz für das Layout „Sorter_3x3“

Die Module B2, C2 und D2 besitzen ebenfalls einen hohen Belegungsgrad und einen hohen relativen Durchsatz auf Grund mehrerer Faktoren: Der erste Faktor ist, dass alle drei Module an Quellen angrenzen und somit z.B. jede Transporteinheit, die auf der Quelle B1 eingeschleust wird, immer über das Modul B2 transportiert werden muss. Der zweite Faktor ist, dass die Module B2 und D2 zusätzlich noch an eine Senke angrenzen. Dies hat zur Folge, dass jede Transporteinheit, die als Ziel die Senke E2 besitzt, über das Modul D2 transportiert werden muss. Der dritte Faktor ist, dass die Module B2, C2 und D2 für viele Kombinationen von Quelle und Senke die kürzeste Route darstellen. Zum Beispiel ist die kürzeste Route von der Quelle D1 zu der Senke A2 folgendermaßen: $D1 \rightarrow D2 \rightarrow C2 \rightarrow B2 \rightarrow A2$.

Der Belegungsgrad und der relative Durchsatz der Module C3 und C4 ist geringer als der Belegungsgrad und der relative Durchsatz der umliegenden Module, da diese nur für Transporteinheiten, die auf der Quelle C1 eingeschleust worden sind und zu dem Ziel C5 transportiert werden müssen, die kürzeste Route darstellen. Für alle anderen Kombinationen von Quelle und Senke liegen die Module C3 und C4 entweder nicht auf dem kürzesten Weg oder es gibt alternative Routen, die genauso kurz sind.

Mit Hilfe der Daten in der Tabelle 7.4 erfolgt eine genauere Analyse der Durchlaufzeiten. Hierbei sind für jede Kombination von Quelle und Senke die bestmögliche Durchlaufzeit (Min.) und der Durchschnitt der Durchlaufzeiten ($\bar{\theta}$) inklusive der jeweiligen Standardabweichung (SE) eingetragen.

Tabelle 7.4: Durchlaufzeit von jeder Quelle zu jeder Senke für das Layout „Sorter_3x3“

Senken	Quellen								
	B1			C1			D1		
	Min.*	Ø*	SE*	Min.*	Ø*	SE*	Min.*	Ø*	SE*
A2	504	1251	53	756	1916	50	1008	2645	49
A3	756	1819	50	1008	2204	30	1260	2647	33
A4	1008	2348	51	1260	2500	26	1512	2781	54
E2	1008	2630	40	756	2053	43	504	1271	45
E3	1260	2662	56	1008	2292	45	756	1852	50
E4	1512	2778	50	1260	2531	37	1008	2353	62
B5	1008	2376	52	1260	2512	56	1512	2746	40
C5	1260	2435	56	1008	2313	49	1260	2474	42
D5	1512	2780	45	1260	2533	42	1008	2367	72

* in ms

Die größte prozentuale Abweichung von der durchschnittlichen Durchlaufzeit zu der bestmöglichen Durchlaufzeit ergibt sich hierbei für die Senken A2 und E2 (zur besseren Visualisierung gelb hinterlegt). Hierbei ist es unerheblich, von welcher Quelle die Transporteinheiten gestartet sind.

Der Grund für diese große prozentuale Abweichung wird am Beispiel der Senke E2 erläutert. Alle Transporteinheiten, die als Ziel die Senke E2 besitzen, müssen über das Modul D2 transportiert werden. Wie auch schon bei der Analyse des Belegungsgrades und des relativen Durchsatzes festgestellt wurde, ist das Modul D2 jedoch nicht nur für die Senke E2 ein Engpass, sondern auch für die Quelle D1. Somit ist das Modul D2 für zwei benachbarte Module ein Engpass, wodurch es, im Vergleich zu den direkten Nachbarmodulen C2 und D3 überbelastet ist, und somit die Durchlaufzeit zu der Senke E2 einbricht.

Die folgende **Regel für die Praxis** wird abgeleitet: Um die Überlastung einzelner Module zu verhindern, sollte jedes Modul höchstens eine Senke oder Quelle als Nachbarmodul besitzen. Im Anhang A6 ist der relative Durchsatz und der Belegungsgrad für die anderen quadratischen Sorter dargestellt. Die für das Layout „Sorter_3x3“ getroffenen Erkenntnisse sind hierbei auch auf die größeren Layouts übertragbar.

Weiterhin wurde untersucht, wie groß der Einfluss der Anzahl der Module, der Quellen und der Senken auf den Durchsatz ist. Da der Einfluss der Anzahl der Quellen und der Senken nur sehr gering mit dem Durchsatz korreliert, werden die entsprechenden Diagramme im Folgenden nicht dargestellt. Wie jedoch in Abbildung 7.4 zu erkennen ist, hat die Anzahl der Module einen Einfluss auf den Durchsatz. Hierbei werden nur die Werte für $n_S = 1,5$ dargestellt. Die Standardabweichung ist so gering, dass sie in diesem Maßstab nicht angezeigt werden kann.

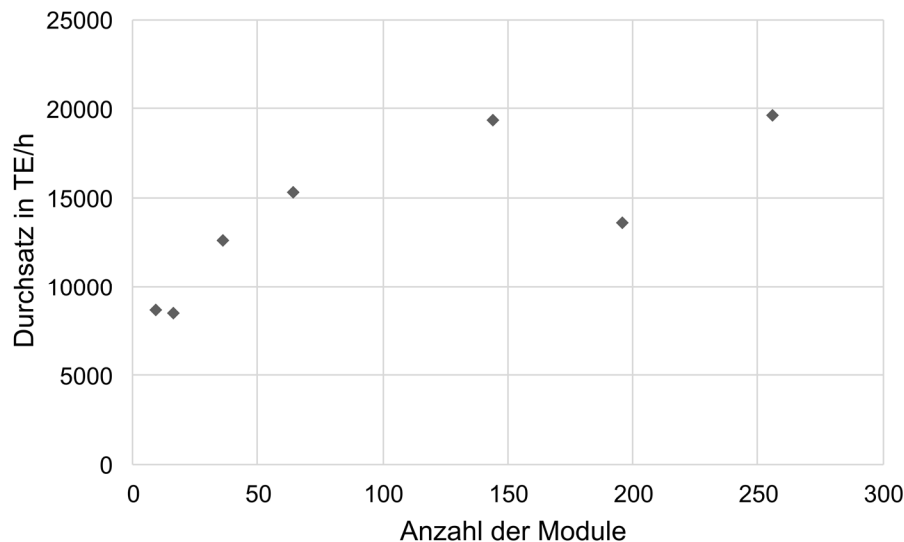


Abbildung 7.4: Durchsatz in Abhängigkeit von der Anzahl der Module für quadratische Sorterlayouts

Dass der Durchsatz nicht linear mit der Anzahl der Module ansteigt ist dadurch begründbar, dass die Anzahl der Quellen und Senken nur linear ansteigt, während die Anzahl der Module quadratisch ansteigt. Somit wird das Hinzufügen von Modulen in größeren Layouts immer unwirtschaftlicher.

7.3 Systemverhalten von rechteckigen Sortern

Nach der Vorstellung des Systemverhaltens der quadratischen Layouts, werden nun die Ergebnisse für rechteckige Sorterlayouts dargestellt. SEIBOLD hat in [Sei16b] insgesamt 84 Sorterlayouts untersucht. Von diesen 84 Layouts hat SEIBOLD drei Layouts ausgewählt, um die wesentlichsten Erkenntnisse über das Systemverhalten beispielhaft darzustellen.

Diese drei von SEIBOLD ausgewählten Sorterlayouts wurden auch im Rahmen dieser Arbeit näher untersucht. Bei diesen drei Layouts wird die Anzahl der Fördermodule und der Senken konstant gehalten und nur die Anzahl der Quellen variiert. Das erste Layout „Sorter_23x11x2“ besitzt zwei Quellen und ist in der Abbildung 7.5 dargestellt.

Im Layout „Sorter_23x11x6“ sind an den folgenden Positionen Quellen: C1, G1, K1, O1, S1 und W1. Im Layout „Sorter_23x11x8“ sind an den folgenden Positionen Quellen: D1, E1, G1, I1, K1, M1, O1, Q1, S1 und U1.

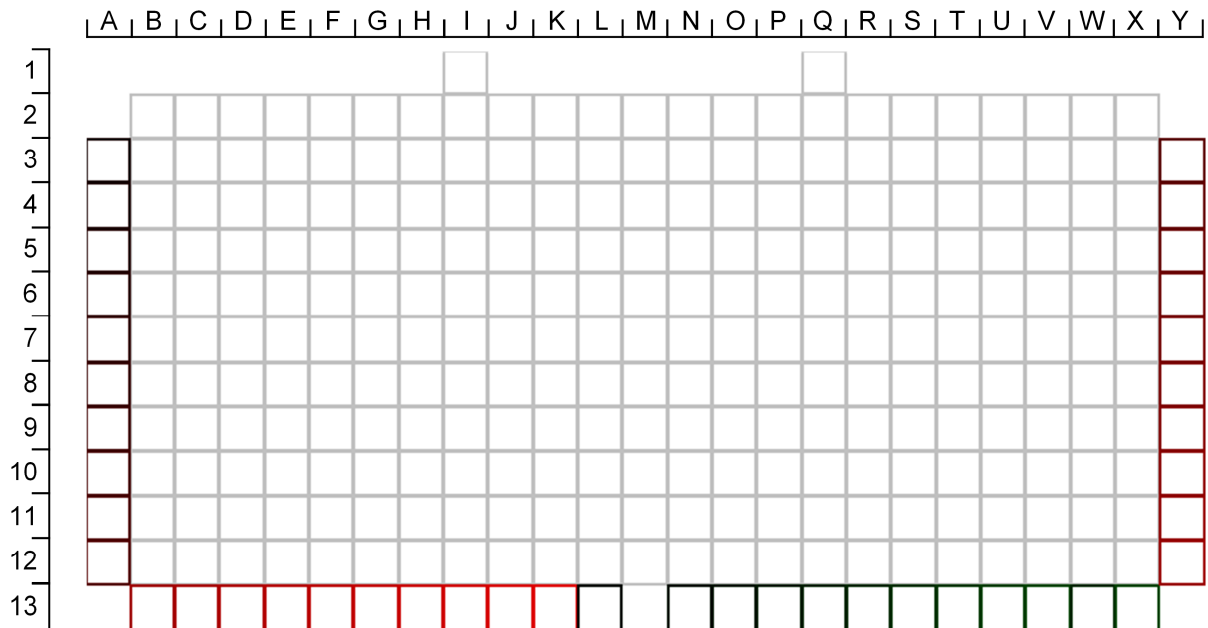


Abbildung 7.5: Layout „Sorter_23_11_2“

Obwohl die gleichen Layouts untersucht wurden, sind die Ergebnisse dennoch nicht direkt vergleichbar, da es Unterschiede in den restlichen Simulationsparametern gibt. Die wesentlichsten Unterschiede sind in der Tabelle 7.5 zusammengefasst.

Tabelle 7.5: Wesentlichste Unterschiede in den Simulationsparametern im Vergleich zu [Sei16b]

Simulationsparameter in ms	[Sei]	[Soh]
Transport zwischen zwei Modulen	1000	250
Änderung der Transportrichtung	200	0
Zeitlicher Abstand bei Tandemtransport	50	1
Dauer der Übertragung einer Nachricht	0,0000001	1

Die von SEIBOLD gewählten Simulationsparameter wirken sich negativer auf den Gesamtdurchsatz aus, als die Simulationsparameter, die in dieser Arbeit gewählt wurden. Um dennoch einen Vergleich der Durchsätze zu ermöglichen, wird ein Proportionalitätsfaktor k_{ges} eingeführt (siehe Formel 7.3). Indem die Durchsätze dieser Arbeit durch den Proportionalitätsfaktor dividiert werden, können sie direkt mit den Durchsätzen von SEIBOLD verglichen werden.

$$k_{ges} = k_{trans} \cdot k_{pro} + k_{com} \cdot (1 - k_{pro}) \quad (7.3)$$

Hierbei ist k_{trans} das Verhältnis aller transportbezogenen Zeiten von [Sei] zu allen transportbezogenen Zeiten von [Soh], k_{com} ist das Verhältnis aller kommunikationsbezogenen Zeiten von [Sei] zu allen kommunikationsbezogenen Zeiten von [Soh] und k_{pro} ist die prozentuale Gewichtung zwischen den transportbezogenen und den kommunikationsbezogenen Zeiten.

Zunächst wird k_{trans} bestimmt, indem die Zeiten für den Transport zwischen zwei Modulen, die Änderung der Transportrichtung und der zeitliche Abstand beim Tandemtransport aufaddiert wird. Für [Sei] ergibt sich hier 1250 ms und für [Soh] 250 ms. Somit ergibt sich für $k_{trans} = 5,0$. Für die kommunikationsbezogenen Zeiten ergibt sich folglich $k_{trans} = 0,00000001$. Die Durchlaufzeit setzt sich aus der transportbezogenen und der kommunikationsbezogenen Zeit zusammen. Hierbei wird angenommen, dass die transportbezogene Zeit 90% der Durchlaufzeit ausmacht und die kommunikationsbezogene Zeit 10%. Somit ergibt sich der folgende Proportionalitätsfaktor $k_{ges} \approx 4,5$.

Bevor die Durchsätze verglichen werden, wird zunächst der Einfluss von n_s auf den Durchsatz bestimmt. Genau wie bei den quadratischen Layouts hat der Wert für n_s nur einen geringen Einfluss auf den Durchsatz. Für das Layout „Sorter_23x11x2“ ist der Durchsatz am höchsten für einen Wert von 1,5, jedoch steigt bei den anderen untersuchten rechteckigen Sorterlayouts der Durchsatz mit größeren Werten für n_s . Der Anstieg ist allerdings so minimal, dass keine tiefere Analyse durchgeführt wird. Die Ergebnisse für die anderen beiden Layouts sind im Anhang A7 dargestellt.

Tabelle 7.6: Darstellung des Durchsatzes auf dem Layout „Sorter_23x11x2“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	12529	12640	12605	12633	12557
Standardabweichung in TE/h	83	67	73	66	78

Die von SEIBOLD ermittelten Durchsätze und die im Rahmen dieser Arbeit ermittelten Durchsätze für einen Wert von 1,5 für n_s sind in der nachfolgenden Tabelle für alle drei Layouts dargestellt. Die Division durch den Proportionalitätsfaktor ist hierbei schon geschehen. Die Werte von SEIBOLD sind mit [Sei] gekennzeichnet, die Werte dieser Arbeit sind mit [Soh] gekennzeichnet.

Tabelle 7.7: Vergleich der Durchsätze für rechteckige Sorterlayouts

	Sorter_23_11_2		Sorter_23_11_6		Sorter_23_11_10	
	[Sei]	[Soh]	[Sei]	[Soh]	[Sei]	[Soh]
Durchsatz in TE/h	4570	2809	~7700	4031	8650	4268

Aus der Tabelle ist ersichtlich, dass die Durchsätze sich für alle drei Layouts in der gleichen Größenordnung bewegen und dass der Durchsatz bei der Verwendung von SEIBOLDS Algorithmus immer höher ist, als bei der Verwendung des Routingalgorithmus, der in dieser Arbeit entwickelt wurde. Weiterhin ist erkennbar, dass der Algorithmus von SEIBOLD besser skaliert, da der Anstieg des Durchsatzes annähernd proportional mit dem Anstieg der Anzahl der Quellen geschieht, während bei Benutzung des Routingalgorithmus dieser Arbeit der Durchsatz nicht proportional mit der Anzahl der Quellen ansteigt.

Ein Grund für die größeren Durchsätze ist, dass der von SEIBOLD entwickelte Algorithmus eine wesentlich größere Anzahl von Steuerungsparametern aufweist, als der Algorithmus, der in dieser Arbeit entwickelt wurde. Ein Beispiel: In dieser Arbeit wird zwar über die Lebenszeit festgelegt, wie lange eine Routenreservierung dauern darf, aber es gibt kein Abbruchkriterium auf Grund der Länge einer Route. SEIBOLD hat nicht nur untersucht, wie sich zeitliche Abbruchkriterien auf den Durchsatz auswirken, sondern auch, welchen Einfluss räumliche Abbruchkriterien besitzen. Auch wurde von SEIBOLD untersucht, wie stark sich die Vermeidung von Engpässen auf den Durchsatz auswirkt. Die von SEIBOLD genutzten zusätzlichen Steuerungsparameter lassen sich auch in den Algorithmus dieser Arbeit integrieren, jedoch ist dies nicht geschehen, weil das vorrangige Ziele dieser Arbeit darin besteht festzustellen, ob überhaupt ein dezentraler Routingalgorithmus mit physikalischen Zeitfenster möglich ist. In zukünftigen wissenschaftlichen Arbeiten ist es daher sinnvoll, die zusätzlichen Parameter von SEIBOLD in diesen Algorithmus zu integrieren, um die Auswirkungen auf den Durchsatz zu ermitteln.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1									100%								100%								
2									85,43%	23,88%	12,46%	9,25%	8,72%	11,19%	12,74%	46,95%	60,27%	33,68%	31,85%	29,99%	28,22%	26,17%	24,49%	23,03%	
3	2,07%	6,64%	4,52%	4,45%	4,38%	4,45%	4,43%	41,79%	62,16%	51,07%	50,14%	48,43%	45,48%	43,91%	48,44%	52,56%	6,03%	6,51%	8,27%	4,95%	5,63%	7,44%	7,29%	24,31%	2,26%
4	1,98%	8,35%	6,88%	6,95%	6,8%	7,56%	8,47%	39,7%	6,4%	7,57%	7,56%	5,41%	3,04%	8%	62,11%	53,36%	31,35%	30,24%	28,91%	25,82%	24,33%	24,37%	11,12%	22,01%	2,07%
5	2,26%	8,23%	5,56%	27,32%	29,47%	31,87%	34,34%	36,61%	8,22%	10,89%	13,13%	13,31%	11,06%	9,8%	32,19%	6,89%	4,27%	4,65%	4,72%	4,72%	5,04%	22,7%	23,48%	32,13%	1,97%
6	2,03%	22,45%	22,92%	24,92%	4,41%	4,42%	4,8%	4,54%	4,14%	4,82%	5,21%	5,71%	11,35%	14,6%	32,6%	6,82%	5,06%	5,8%	5,94%	5,46%	4,41%	4,41%	4,99%	28,5%	2,1%
7	1,92%	19,42%	4,5%	4,29%	3,93%	4,2%	4,62%	4,15%	3,61%	4,17%	4,65%	26,95%	29,05%	32,98%	34,38%	13,15%	11,02%	10,4%	10,27%	5,06%	4,47%	4,55%	5,55%	25,06%	2,09%
8	2,08%	16,67%	5,49%	4,59%	4,34%	4,54%	4,82%	4,5%	3,93%	4,8%	6,91%	26,84%	0,88%	4,49%	4,82%	4,09%	4,01%	3,94%	9,74%	9,23%	8,86%	8,54%	8,72%	24,53%	1,97%
9	2,29%	14,09%	6,29%	4,64%	4,64%	4,51%	4,72%	4,57%	4,1%	21,94%	24,13%	26,31%	1,82%	5,9%	5,72%	4,92%	4,87%	4,07%	4,59%	4,11%	4,3%	4,39%	4,4%	20,51%	2,2%
10	2,11%	12,06%	6,23%	4,84%	4,82%	4,69%	4,83%	4,68%	4,65%	20,71%	4,52%	4,42%	0,04%	4,06%	4,07%	4,09%	5,07%	5,24%	5,28%	4,73%	4,87%	4,85%	4,77%	16,42%	2,12%
11	1,87%	9,15%	4,75%	4,36%	4,55%	4,75%	5,33%	5,3%	4,61%	19,13%	4,53%	4,51%	0,12%	4,15%	4,11%	4,05%	4,19%	4,12%	4,03%	4,11%	4,35%	4,34%	4,39%	12,13%	2,13%
12	2,06%	8,49%	7,59%	8,72%	10,05%	11,85%	13,61%	15,11%	16,15%	17,71%	4,46%	4,43%	0,04%	4,08%	4,08%	4,1%	4,26%	4,19%	4,07%	4,15%	4,39%	4,35%	4,44%	8,02%	2%
13	2,19%	2,09%	2,01%	1,96%	2,25%	2,18%	2,22%	1,96%	2,22%	2,24%	2,22%			2,03%	2%	2,01%	2,1%	2,06%	2%	2,06%	2,17%	2,16%	2,12%	1,97%	

Abbildung 7.6: Belegungsgrad für das Layout „Sorter_23x11x2“

In der Abbildung 7.6 ist der Belegungsgrad für das Layout „Sorter_23x11x2“ dargestellt und im Anhang A7 sind der Belegungsgrad und der relative Durchsatz für alle rechteckigen Sorterlayouts dargestellt. In der Abbildung sind „wurzelartige“ Pfade zu erkennen, die entstanden sind, weil Module entlang dieser Pfade eine höhere Belegung haben als ihre direkten Nachbarn. Diese „wurzelartigen“ Pfade sind auch erkennbar, wenn die relativen Durchsätze betrachtet werden. Die Pfadbildung ist hierbei nicht symmetrisch. Dies ist dadurch zu erklären, dass der Algorithmus, wenn zwei benachbarte Module gleich große Kosten aufweisen, eine Priorisierung in der folgenden Reihenfolge durchführt: Osten, Süden, Westen und Norden. Somit erklärt sich, warum im rechten Teil des Layouts die Pfade stärker nach Osten gehen als nach Süden. Dies ist genau umgekehrt im linken Teil des Layouts: Hier gehen die Pfade zuerst nach Süden, bevor sie nach Westen gehen.

Wenn der Belegungsgrad und der relative Durchsatz mit den Ergebnissen von SEIBOLD verglichen werden ist auffällig, dass sich bei SEIBOLD keine „wurzelartigen“ Pfade ergeben. Stattdessen werden die meisten Transporteinheiten auf rechtwinkligen Routen transportiert, also zuerst in die entsprechende Zeile und anschließend in die entsprechende Reihe. Dies ist darauf zurückzuführen, dass Richtungsänderungen in SEIBOLDS Simulation zusätzliche Zeit benötigen und somit diagonale Routen mehr Zeit benötigen als rechtwinklige.

7.4 Systemverhalten von quadratischen Kreuzungen

In diesem Kapitel wird das Systemverhalten von quadratischen Kreuzungslayouts am Beispiel des Layouts „Inter_3x3“ dargestellt. In der Abbildung 7.7 ist der Belegungsgrad und der relative Durchsatz für das Layout dargestellt. Die Quellen sind hierbei daran zu erkennen, dass sie einen Belegungsgrad von 100% aufweisen. Im Gegensatz zu dem Belegungsgrad und dem relativen Durchsatz der Sorterlayouts ist zu erkennen, dass die Auslastung der Module wesentlich homogener ist, da auch die Quellen und Senken homogener verteilt sind.

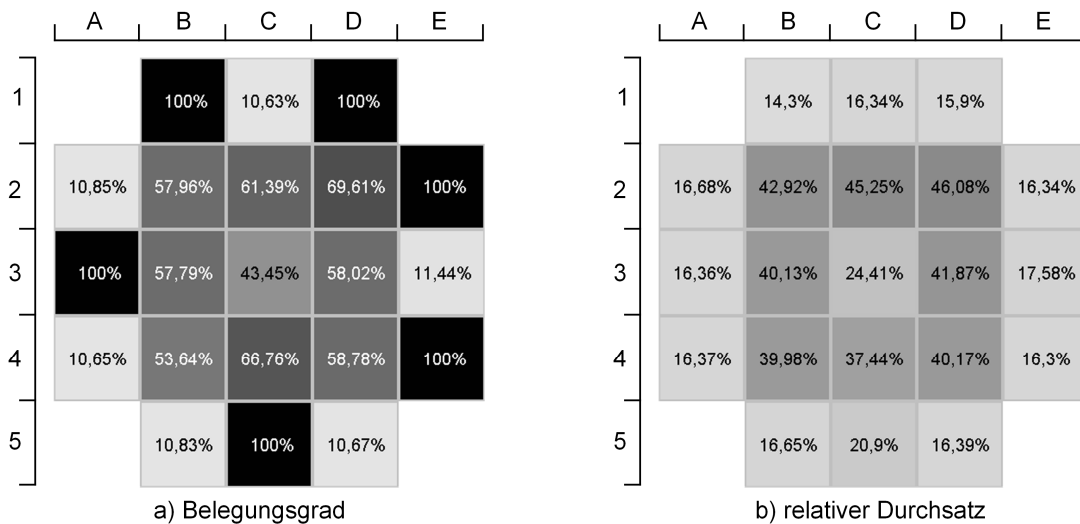


Abbildung 7.7: Belegungsgrad und relativer Durchsatz für das Layout „Inter_3x3“

In der Tabelle 7.8 ist der Einfluss von Werten für n_s auf den Durchsatz abgebildet. Der höchste Durchsatz ergibt sich hierbei für einen Wert von 1,0, jedoch ist die Auswirkung von verschiedenen Werten von n_s auf den Durchsatz genau wie bei den vorher untersuchten Sortern gering. Im Anhang A8 sind die Auswirkungen von unterschiedlichen Werten von n_s auf den Durchsatz von weiteren Layouts dargestellt. Hierbei sind bei allen die Auswirkungen auf den Durchsatz immer sehr gering und es ergeben sich die höchsten Durchsätze für Werte für n_s , die zwischen 1,0 und 1,5 liegen.

Tabelle 7.8: Darstellung des Durchsatzes auf dem Layout „Inter_3x3“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	9456	9349	9244	9261	9219
Standardabweichung in TE/h	72	75	57	74	75

Im direkten Vergleich zu dem Sorterlayout „Sorter_3x3“ (siehe Tabelle 7.2) sind die Durchsätze für das Kreuzungslayout „Inter_3x3“ höher. Für den höheren Durchsatz gibt es mehrere Gründe: Der erste Grund besteht darin, dass das gleichmäßigere Verhältnis von Quellen zu Senken dazu führt, dass die Quellen nicht mehr die Engpässe darstellen. Weiterhin sinkt durch die Gleichverteilung der Quellen und Senken die durchschnittliche Distanz zwischen Senken und Quellen. Jedoch gibt es auch Effekte, die sich nachteilig auf den Durchsatz auswirken in den Kreuzungslayouts: So gibt es mehr Möglichkeiten für Konflikte, da Routen häufiger entgegengesetzt reserviert werden müssen. Da sich jedoch dieser negative Effekt nicht so stark auswirkt, wie der positive Effekt, steigt der Durchsatz an.

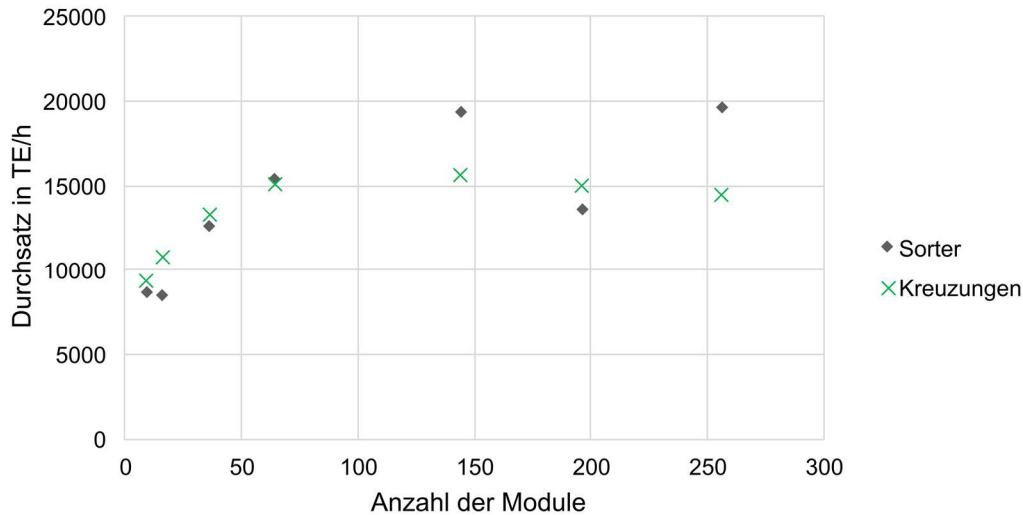


Abbildung 7.8: Durchsatz in Abhängigkeit von der Anzahl der Module für quadratische Kreuzungslayouts und Sorterlayouts

In der Abbildung 7.8 ist die Entwicklung des Durchsatzes sowohl für Sorter- als auch für Kreuzungslayouts in Abhängigkeit von der Anzahl der Module dargestellt. Aus der Abbildung ist zu erkennen, dass sich die Durchsätze für Sorter- und Kreuzungslayouts bis zu einer Anzahl von 100 Modulen nicht stark unterscheiden. Während danach jedoch bei den Sorterlayouts der Durchsatz weiterhin ansteigt, sinkt dieser für die Kreuzungslayouts. Dies ist dadurch zu erklären, dass durch die steigende Anzahl von Quellen auch die Anzahl an Transporteinheiten ansteigt. Durch die größere Anzahl an Transporteinheiten müssen mehr Konflikte verhindert werden und die Routenreservierung dauert entsprechend länger.

7.5 Fazit

Über alle Simulationsläufe haben über 29 Millionen Transporteinheiten ihr Ziel erreicht. Zur Einordnung: Im Advent 2018 wurden durch die deutsche Post pro Tag elf Millionen Transporteinheiten zugestellt [Süd18]. Durch die große Anzahl der erfolgreich durchgeführten Transporte in den Simulationsläufen ist nachgewiesen, dass der Algorithmus auch korrekt implementiert wurde, während der theoretische Korrektheitsbeweis bereits in dem Kapitel 6.3.1 erbracht wurde.

Die Auswirkungen des einstellbaren Parameters n_s auf den Durchsatz wurden untersucht. Hierbei wurde festgestellt, dass dieser nur einen geringen Einfluss auf den Durchsatz aufweist. Ein Wert zwischen 1,0 und 1,5 hat jedoch leicht positive Auswirkungen. Der Durchsatz wird jedoch von anderen Faktoren wesentlich stärker beeinflusst. Bei den Sorterlayouts stellen z.B. die Quellen und ihre direkt benachbarten Module Engpässe dar. Besonders das Vorhandensein von Modulen, die mit zwei oder

mehr Quellen/Senken benachbart sind, wirkt sich negativ auf den Durchsatz aus. Da bei den untersuchten Kreuzungslayouts das Verhältnis von Quellen zu Senken 1:1 ist, stellen hier die Quellen nicht die Engpässe dar.

Bei den quadratischen Sorterlayouts steigen die Durchsätze mit steigender Anzahl der Module an. Jedoch ist dieser Anstieg nicht proportional, sondern der Zuwachs des Durchsatzes wird immer geringer. Dies liegt daran, dass die Anzahl der Quellen linear ansteigt, während die Anzahl der Module quadratisch ansteigt. Es wird somit immer unwirtschaftlicher Module hinzuzufügen. Dieser Trend verstärkt sich bei den quadratischen Kreuzungslayouts sogar noch, da hier das Hinzufügen von Modulen ab einer Anzahl von 150 Modulen sogar zu einer Verringerung des Durchsatzes führt. Dies ist dadurch begründet, dass in Kreuzungslayouts Konflikte wesentlich häufiger verhindert werden müssen und somit der Durchsatz sinkt.

Der in dieser Arbeit entwickelte Algorithmus wurde mit dem von SEIBOLD in [Sei16b] entwickelte Algorithmus verglichen. Hierbei wurde ermittelt, dass durch den Einsatz des Algorithmus von SEIBOLD in jedem untersuchten Layout größere Durchsätze erzielt werden. Der Grund hierfür ist, dass der Algorithmus von SEIBOLD eine wesentlich größere Anzahl von einstellbaren Parametern aufweist, mit denen der Algorithmus besser eingestellt werden kann. Die einstellbaren Parameter von SEIBOLD können auch in den Algorithmus dieser Arbeit integriert werden. In zukünftigen Arbeiten sollte daher untersucht werden, wie sich das Hinzufügen dieser zusätzlichen Parameter auf den Durchsatz auswirken wird.

In dem Anhang A9 wurde die Laufzeit des Algorithmus näher untersucht. Hierdurch konnte ermittelt werden, dass die benötigte Rechenkapazität eines Moduls unabhängig von der Gesamtanzahl der Module im System ist. Somit kann das System beliebig skaliert werden.

Kapitel 8

Empfehlungen für den industriellen Einsatz

8.1 Weiterentwicklung der netkoPs-Fördermatrix

Das Referenzfördersystem (siehe Kapitel 4) besteht aus kleinskaligen netkoPs-Fördermodulen. Diese netkoPs-Fördermodule können sowohl auf der Hardware- als auch auf der Softwareebene weiterentwickelt werden. Auf der Hardwareebene bietet sich die Entwicklung von spezialisierten Modulen an. MELZIAN hat in der Studienarbeit von [Mel16] den Einsatz der netkoPs-Fördermodule in einer Reifenproduktion untersucht. Für den Produktionsschritt des „Spottings“ ist es wichtig, dass der Reifenrohling zentriert ausgerichtet wird. Hierzu kann eines der netkoPs-Fördermodule durch ein Zentrierungsmodul ausgetauscht werden. Die Zentrierung des Reifenrohlings erfolgt, indem ein konischer Zentrierstift ausgefahren wird.

Eine weitere Möglichkeit zur Weiterentwicklung besteht darin, die Anzahl der benötigten Steuerungen zu reduzieren, um die Kosten zu senken. In dem Referenzfördersystem steuert eine übergeordnete Steuerung 36 einzelne Fördermodule an. Die Software ist objektorientiert aufgebaut und somit ist jedem Hardware-Fördermodul ein Software-Fördermodul zugeordnet. Wenn eine größere Anzahl von Hardware-Fördermodulen durch eine einzige Steuerung angesteuert wird, hat dies nicht nur Kostenvorteile, sondern auch das Senden und Empfangen von Nachrichten zwischen den einzelnen Software-Fördermodulen wird stark beschleunigt. Nachrichten werden nicht mehr über ein physikalisches Medium übertragen, sondern nur noch im internen Speicher der Steuerung zwischen den Software-Fördermodulen ausgetauscht.

Da in dem Referenzfördersystem nur eine Fördermatrix mit 36 Modulen verwendet wurde, konnte auf eine Regelung verzichtet werden. Weil sich immer nur eine Transporteinheit auf der Fördermatrix befinden kann und zudem die Wege auf der Matrix sehr kurz sind, tritt ein Verkanten der Transporteinheiten nicht auf. Wenn jedoch auf einer größeren Fördermatrix mehrere Transporteinheiten gleichzeitig bewegt werden, kann es z.B. auf Grund von Abnutzungserscheinungen einzelner Förderrollen zu einem Verkanten kommen. Um die durch das Verkanten entstehenden Bahnabweichungen zu korrigieren, kann eine Regelung eingesetzt werden. Die Reservierung von Routen mit physikalischen Zeitfenstern erleichtert hierbei den Einsatz einer Regelung, da die einzelnen Module auf Grund der Zeitfenster die Information besitzen, wann die Transporteinheiten auf einem Modul detektiert werden sollten. Falls eine Transporteinheit später detektiert wird als vorherberechnet, kann das Fördermodul die Fördergeschwindigkeit proportional zu der Verspätung erhöhen. Somit wird der Rückstand wieder ausgeglichen. Falls die Transporteinheit zu früh detektiert wird, kann das Fördermodul die Geschwindigkeit entsprechend reduzieren.

Eine weitere Möglichkeit zur Verbesserung besteht darin, dass die einzelnen Module abspeichern, wie häufig sie befahren werden. Bei der Routenreservierung können dann Module bevorzugt werden, die weniger häufig befahren werden. Hierdurch kann das Stauen von Transporteinheiten vermindert werden, da diese gleichmäßiger auf die vorhandenen Module verteilt werden.

Weiterhin können die von SEIBOLD in [SEI16a] vorgestellten zusätzlichen Einstellungsparameter in den im Rahmen dieser Arbeit entwickelten Algorithmus integriert werden. Hierzu gehören z.B. partielle Routenreservierungen, bei denen der Transport auch begonnen werden kann, wenn die Routenreservierung noch nicht die Senke erreicht hat oder auch das Hinzufügen von zusätzlichen Abbruchkriterien bei der Routenfindung, um die Wahrscheinlichkeit zu erhöhen, die global optimale Route schneller zu finden.

In dem Referenzfördersystem wurden zwei Typen von Förderern eingesetzt: Stetigförderer und Fördermatrizen, die mit kleinskaligen Fördermodulen bestückt sind. Der im Rahmen dieser Arbeit entwickelte Algorithmus eignet sich sowohl für die Ansteuerung von großskaligen als auch kleinskaligen Fördermatrizen. In dem Referenzfördersystem wurde eine erste Version einer dezentralen Steuerung für Stetigförderer umgesetzt. Diese Steuerung nutzt die gleichen Verfahren zu Vermeidung von Konflikten wie bei den Fördermatrizen. Ein Nachteil besteht darin, dass die Steuerung nur eine Transporteinheit pro Bandförderer zulässt. Somit werden Stetigförderer ineffizient genutzt, da die Hardware in der Lage ist, mehrere Transporteinheiten gleichzeitig aufzunehmen und zu transportieren.

Stetigförderer sind ein wesentlicher Bestandteil von konventionellen Fördersystemen. Wenn ein bestehendes konventionelles Fördersystem mit dezentralen Steuerungen nachgerüstet wird, dann müssen die verwendeten Steuerungen in der Lage sein, eine effiziente Ansteuerung der Stetigförderer zu gewährleisten. Aus der Steuerungsperspektive besteht die Besonderheit der Ansteuerung der Stetigförderer darin, dass Transporteinheiten nicht einzeln bewegt werden können, sondern alle Transporteinheiten auf einem Stetigförderer gleichzeitig bewegt werden. RUMBERG hat in der Studienarbeit [Rum17] schon die Schwierigkeiten und Randbedingungen bei der Erstellung einer dezentralen Steuerung für Stetigförderer beschrieben.

8.2 Anwendungsszenario

Bevor ein modulares Fördersystem in eine Produktion integriert wird, ist zunächst eine Wirtschaftlichkeitsbetrachtung durchzuführen. Eine veröffentlichte Methodik zur Erkennung der „Marktnische“ für modulare Fördersysteme ist dem Autor dieser Dissertation nicht bekannt. Im Bereich des Materialflusses gibt es drei vorherrschende Systeme: Konventionelle Fördertechnik, FTF-Flotten und manuelle Arbeit. Die modularen Fördersysteme sind hierbei unflexibler als FTFs und manuelle Arbeit, aber flexibler als konventionelle Fördertechnik. Der Durchsatz von modularen Fördersystemen ist größer als der von FTF und manueller Arbeit und vergleichbar mit dem Durchsatz von konventionellen Fördersystemen. Die Anschaffungskosten von modularen Fördersystemen sind höher als die initialen Kosten für das Anlernen von Mitarbeitern, welche die manuelle Arbeit ausführen. Weiterhin sind die Anschaffungskosten von modularen Fördersystemen auch höher als die Anschaffungskosten von konventionellen Fördersystemen. Die Anschaffungskosten von modularen Fördersystemen werden als gleichhoch geschätzt, wie die von einer FTF-Flotte. Die Betriebskosten von konventionellen Fördersystemen, modularen Fördersystemen und FTF-Flotten sind geringer als die Betriebskosten von manueller Arbeit. In der Tabelle 8.1 sind die Eigenschaften der Fördersysteme zusammenfassend gegenübergestellt. Aus dieser Tabelle kann ein Anwendungsfall für modulare Fördersysteme abgeleitet werden: Der Einsatz ist sinnvoll, wenn ein hoher Durchsatz bei einer hohen Flexibilität und geringen Betriebskosten gefordert wird. Da die FTF ähnliche Eigenschaften aufweisen wie die modularen Fördersysteme und darüber hinaus in zunehmendem Maße in der Industrie eingesetzt werden, muss bei der Wirtschaftlichkeitsuntersuchung besonders auf die Abgrenzung zu den FTF geachtet werden.

Falls eine Firma durch eine Wirtschaftlichkeitsuntersuchung zu dem Schluss kommt ein modulares Fördersystem einzusetzen, muss das Layout von diesem Fördersystem

noch erstellt werden. SHCHEKUTIN hat in [Shc17a, Shc17b, Shc19] Methoden zur Erstellung und Optimierung von Layouts bestehend aus modularen Fördersystemen entwickelt.

Tabelle 8.1: Gegenüberstellung der Eigenschaften von Fördersystemen

Eigenschaft	Konventionelle Fördersysteme	FTF	Manuelle Arbeit	Modulare Fördersysteme
Anschaffungskosten / initiale Kosten	Hoch	Hoch	Niedrig	Hoch
Betriebskosten	Niedrig	Niedrig	Hoch	Niedrig
Durchsatz	Hoch	Mittel	Niedrig	Hoch
Flexibilität	Niedrig	Hoch	Sehr hoch	Hoch

Ein Hinderungsgrund für den Einsatz von modularer Fördertechnik in der Produktion kann die dezentrale Steuerung sein, da es sich um ein vollständig anderes Steuerungsparadigma handelt als bei den konventionellen Fördersystemen. Die einsetzenden Personen sind es gewohnt, dass das Fördersystem nach ihren Vorgaben programmiert wird und dann anschließend genau die programmierten Abläufe einhält. Da ein dezentral gesteuertes System die Wahl der Route selbstständig durchführt, haben die einsetzenden Personen keine direkte Eingriffsmöglichkeit mehr. Dieser im Vergleich zu konventionellen Fördersystemen ungewöhnliche Wegfall der Eingriffsmöglichkeit kann zu Akzeptanzproblemen führen, da die einsetzenden Personen das Gefühl eines Kontrollverlustes haben können. Um die Akzeptanz von modularen Fördersystemen zu erhöhen, sollte daher auch die Möglichkeit der klassischen Programmierung vorgesehen werden. Dies kann so aussehen: Die einsetzenden Personen positionieren die modularen Fördersysteme in ihrer Produktion und verbinden sie mit den vor- und nachgelagerten Prozessen. Das modulare Fördersystem wird angeschaltet und erkennt das aktuelle Layout, welches auf einem Tablet den einsetzenden Personen angezeigt wird. Die einsetzenden Personen können nun Routen durch das Fördersystem definieren und Vorfahrtsregeln an Kreuzungen festlegen. Die Firma flexlog GmbH, die ein modulares Fördersystem anbietet, hat auf diversen Messeständen eine solche Möglichkeit der Programmierung vorgestellt [Fle20]. Auch der Celluveyor verfügt über die Möglichkeit der externen Routenerstellung [Cel20].

Kapitel 9

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein dezentraler Steuerungsalgorithmus für Fördermodule entwickelt, der Routen mit physikalischen Zeitfenstern reserviert. Zuerst wurden nach dem Stand der Wissenschaft und Technik die Anforderungen an ein flexibles Fördersystem dargestellt (Kapitel 2.1). Anschließend erfolgte die nähere Untersuchung der zwei Arten von flexiblen Fördersystemen: Modulare Fördersysteme und fahrerlose Transportfahrzeuge (FTF).

Die modularen Fördersysteme wurden anhand ihrer Größe in zwei Klassen unterteilt: Die großskaligen Fördersysteme und die kleinskaligen Fördersysteme. Bei den großskaligen Fördersystemen nutzt nur der Algorithmus von SEIBOLD Zeitfenster bei der Reservierung von Routen (Kapitel 2.2.1). Hierbei werden jedoch keine physikalischen Zeitfenster reserviert, sondern logische. Bei den kleinskaligen Fördersystemen nutzt kein Algorithmus einen zeitfensterbasierten Routingansatz. Jedoch hat KRÜHN gezeigt, dass durch das Konzept der Modulnachbarschaften Algorithmen, die für großskalige Fördersysteme entwickelt worden sind, auch für kleinskalige genutzt werden können. Anschließend wurde der Stand der Wissenschaft und Technik zu den Routingalgorithmen von FTF untersucht. Für die Steuerung von FTF gibt es eine große Anzahl von zeitfensterbasierten Routingalgorithmen. Im Rahmen dieser Arbeit wurde der Algorithmus von KIM & TANCHOCO näher untersucht.

Aufbauend auf den Erkenntnissen aus dem Stand der Wissenschaft und Technik wurde die Forschungsfrage abgeleitet. Dieses lautete: Ist es möglich, einen zeitfensterbasierten Routingalgorithmus für ein dezentral gesteuertes modulares Fördersystem zu entwickeln?

Im Rahmen dieser Arbeit wurde als Referenzfördersystem das netkoPs-Fördersystem ausgewählt, welches am ITA entwickelt wurde (Kapitel 4). Teil dieses Referenzfördersystems sind modulare Fördermatrizen. Von diesen Fördermatrizen wurde ein Software-Modell gebildet (Kapitel 5). Dieses Software-Modell der Fördermatrix wurde zusammen mit einem Software-Modell eines Transportgutes in einer Simulationsumgebung eingesetzt. Mit Hilfe der Simulationsumgebung ist es möglich, eine große Anzahl von Fördermatrizen und Transporteinheiten zu simulieren.

Im Kapitel 6 wurde der im Rahmen dieser Arbeit entwickelte zeitenfensterbasierte Routingalgorithmus beschrieben. Die Funktionsweise wurde zunächst für großskalige Fördermodule beschrieben und im Anschluss wurde eine Modifikation für kleinskalige Fördermodule dargestellt. Nach der Beschreibung der Funktionsweise wurde der Beweis erbracht, dass Konflikte, zu denen Kollisionen, Verklemmungen, Livelocks und das Verhungern zählen, ausgeschlossen sind. Weiterhin wurde die Laufzeit des Algorithmus bestimmt.

In dem Kapitel 7 wurde anschließend der entwickelte Steuerungsalgorithmus mit Hilfe der Simulationsumgebung aus dem Kapitel 5 untersucht. Hierbei wurde der Einfluss der Steuerungsparameter auf wichtige Messgrößen, wie z.B. den Durchsatz an Transporteinheiten, bestimmt. Hierbei wurde auch untersucht, ab welchen Größen von Layouts das Hinzufügen von Modulen unwirtschaftlich wird.

Abschließend wurden in dem Kapitel 8 Empfehlungen für den industriellen Einsatz gegeben. Da in konventionellen Fördersystemen eine große Anzahl von Stetigförderern eingesetzt werden, ist die Entwicklung und Untersuchung von dezentralen Steuerungsalgorithmen für Stetigförderer sowohl wirtschaftlich als auch wissenschaftlich interessant. Besonders das Zusammenspiel von dezentral gesteuerten, modularen Fördermatrizen mit dezentral gesteuerten Stetigförderern muss erforscht werden.

In vorangegangenen Arbeiten ist die Entwicklung und Untersuchung von Steuerungsalgorithmen für großskalige und kleinskalige Fördermodule getrennt erfolgt. Im Rahmen dieser Arbeit wurde gezeigt, dass Algorithmen, die für großskalige Module entwickelt worden sind, durch die Verwendung des Konzeptes der Modulnachbarschaft mit geringfügigen Modifikationen auch für kleinskalige Module eingesetzt werden können. Weiterhin wurde gezeigt, dass mit minimalen Anpassungen Algorithmen, die für FTF entwickelt worden sind, auch für modulare Fördersysteme eingesetzt werden können. Eine strikte Trennung der Steuerungsalgorithmen für großskaligen und kleinskaligen Fördermodulen sowie FTF ist somit nicht mehr sinnvoll. Bei der zukünftigen Weiterent-

wicklung von Steuerungsalgorithmen für modulare Fördersysteme sollte in Zukunft zuerst geprüft werden, ob ein bereits veröffentlichter Steuerungsalgorithmus, der für FTF entwickelt wurden, nicht schon die gewünschten Eigenschaften aufweist.

Ein wissenschaftlich interessantes Forschungsfeld ist die Ansteuerung eines modularen Fördersystems mit einer Steuerung, die mit Hilfe der Methoden der künstlichen Intelligenz (KI) entstanden ist. Besonders vielversprechend erscheint hierbei der Einsatz des „Deep Learning“. Bei diesem Ansatz wird ein künstliches neuronales Netz mit zahlreichen Zwischenschichten trainiert. Die Bedeutung dieses Ansatzes wird dadurch unterstrichen, dass die Veröffentlichung von BAHDANAU ET AL. [Bah14], die das „Deep Learning“ erstmalig vollständig beschreibt, bereits über 10 000 Mal zitiert wurde und zu einer deutlichen Verbesserung z.B. im Bereich der maschinellen Übersetzung von Texten geführt hat [Pin20]. Das neuronale Netz kann hierbei mit der in dieser Arbeit entstandenen Simulationsumgebung trainiert werden: Zuerst werden verschiedene Kriterien, wie z.B. die Vermeidung von Verklemmungen und die Erreichung eines hohen Durchsatzes, als Zielparameter definiert. Das Netz wird anschließend in einem Layout so lange trainiert, bis die Zielparameter alle ausreichend erfüllt werden. Anschließend wird das Training in einem anderen Layout fortgesetzt, bis auch dort eine ausreichende Erfüllung der Zielparameter erfolgt ist. Das Training in verschiedenen Layouts muss solange fortgesetzt werden, bis das Netz in der Lage ist, auch in unbekanntem Layouts zuverlässig die Zielparameter zu erfüllen.

Abbildungsverzeichnis

Abbildung 2.1: Zwei mögliche Arten von Verklemmungen in Fördersystemen, nach [Sei16b].....	4
Abbildung 2.2: Ring- und Drehsorter [Jod12]	5
Abbildung 2.3: FlexConveyor-Modul, nach [May09].....	11
Abbildung 2.4: Verklemmungssituation in einem Kreisverkehr, nach [May09].....	13
Abbildung 2.5: Szenario, in dem die Benutzung des Algorithmus von MAYER nachteilig ist, angelehnt an [May09].....	14
Abbildung 2.6: Vergleich der möglichen Routen des GridStore-Systems und des FlexConveyor-Systems.....	17
Abbildung 2.7: FlexTechnology, nach [Fle20]	18
Abbildung 2.8: Festo Motion Cube, nach [Wek17]	19
Abbildung 2.9: Karis Pro [Sto12]	20
Abbildung 2.10: AutoStore, nach [Aut18]	21
Abbildung 2.11: cogniLog-Module, nach [Krü15].....	22
Abbildung 2.12: MCS-Modul, [Ito17].....	23
Abbildung 2.13: Celluveyor-Module, nach [Uri13]	24
Abbildung 2.14: Modulnachbarschaft, nach [Krü15].....	25
Abbildung 2.15: Verschiedene FTF-Bauformen, nach [Ama20, Fra17, Sig17, SSI17]	26
Abbildung 3.1: Bearbeitungsreihenfolge der Teilziele	41
Abbildung 4.1: Beispiellayout bestehend aus Fördermatrizen und Bandförderern.....	44
Abbildung 4.2: netkoPs-Fördermodul.....	45
Abbildung 4.3: netkoPs-Fördermatrix.....	46
Abbildung 4.4: Aufbau der Beagle Bone basierten Steuerung der Fördermatrix	47
Abbildung 4.5: Aufbau der Beckhoff basierten Steuerung der Fördermatrix	49

Abbildung 5.1:	Screenshot der Simulationsumgebung	58
Abbildung 6.1:	Algorithmus zur Verarbeitung von empfangenen Nachrichten.....	62
Abbildung 6.2:	Sub-Algorithmus zum Starten einer neuen Routinganfrage	65
Abbildung 6.3:	Einfügen einer neuen Routinganfrage in den Ablaufplan.....	67
Abbildung 6.4:	Sub-Algorithmus zur Verarbeitung von BESTÄTIGUNGS- Nachrichten.....	68
Abbildung 6.5:	Systemzustände zum Zeitpunkt $t=10$ ms und $t=11$ ms	70
Abbildung 6.6:	Systemzustände zum Zeitpunkt $t=12$ ms und $t=13$ ms	71
Abbildung 6.7:	Systemzustand für den Zeitpunkt $t=14$ ms.....	73
Abbildung 6.8:	Verarbeitung von Nachrichten durch kleinskalige Fördermodule.....	77
Abbildung 6.9:	Verarbeitung von ABLAUFPLAN, ABLAUFPLAN- BESTÄTIGUNGS und ABLAUFPLAN-ANFRAGE Nachrichten.....	78
Abbildung 6.10:	Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN Nachrichten durch Hauptmodule.....	80
Abbildung 6.11:	Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN Nachrichten durch Nachbarschaftsmodule	81
Abbildung 6.12:	Sub-Algorithmus zur Verarbeitung von ABLAUFPLAN- BESTÄTIGUNGS Nachrichten.....	82
Abbildung 6.13:	Sequenzdiagramm für die Verarbeitung einer ANFRAGE- Nachricht.....	83
Abbildung 7.1:	Layout „Sorter_3x3“	95
Abbildung 7.2:	Die Routenreservierung entfernt sich von der Senke.....	98
Abbildung 7.3:	Belegungsgrad und relativer Durchsatz für das Layout „Sorter_3x3“	100
Abbildung 7.4:	Durchsatz in Abhängigkeit von der Anzahl der Module für quadratische Sorterlayouts	102
Abbildung 7.5:	Layout „Sorter_23_11_2“	103
Abbildung 7.6:	Belegungsgrad für das Layout „Sorter_23x11x2“	105
Abbildung 7.7:	Belegungsgrad und relativer Durchsatz für das Layout „Inter_3x3“	107
Abbildung 7.8:	Durchsatz in Abhängigkeit von der Anzahl der Module für quadratische Kreuzungslayouts und Sorterlayouts.....	108

Tabellenverzeichnis

Tabelle 2.1:	Kleinladungsträger nach VDA 4500 [VDA17]	9
Tabelle 2.2:	Sicherstellung des Betriebes durch die einzelnen Algorithmen	33
Tabelle 2.3:	Beurteilung der Leistungsfähigkeit durch die Algorithmen.....	34
Tabelle 2.4:	Ermöglichung der Wandlungsfähigkeit durch die einzelnen Algorithmen.....	35
Tabelle 7.1:	Übersicht über alle simulierten Layouts.....	94
Tabelle 7.2:	Darstellung des Durchsatzes auf dem Layout „Sorter_3x3“ für unterschiedliche Werte von n_S	96
Tabelle 7.3:	Durchlaufzeit im günstigsten Fall und in einem ungünstigeren Fall von der Quelle B1 zu der Senke B5 im Layout „Sorter_3x3“.....	97
Tabelle 7.4:	Durchlaufzeit von jeder Quelle zu jeder Senke für das Layout „Sorter_3x3“.....	101
Tabelle 7.5:	Wesentlichste Unterschiede in den Simulationsparametern im Vergleich zu [Sei16b].....	103
Tabelle 7.6:	Darstellung des Durchsatzes auf dem Layout „Sorter_23x11x2“ für unterschiedliche Werte von n_S	104
Tabelle 7.7:	Vergleich der Durchsätze für rechteckige Sorterlayouts.....	104
Tabelle 7.8:	Darstellung des Durchsatzes auf dem Layout „Inter_3x3“ für unterschiedliche Werte von n_S	107
Tabelle 8.1:	Gegenüberstellung der Eigenschaften von Fördersystemen.....	114

Literaturverzeichnis

- [All83] ALLEN, J. F.: *Maintaining knowledge about temporal intervals*, Communications of the ACM, Vol. 26, Nr. 11, S. 832–843, <https://doi.org/10.1145%2F200836.200848>, 1983.
- [Ama20] AMAZON ROBOTICS LLC (HRSG.): *Amazon Robotics*, <https://www.amazonrobotics.com/>, Stand: 04.10.2020.
- [Aut18] JAKOB HATTELAND GROUP MANAGEMENT AS (HRSG.): *AutoStore Systems*, <http://www.autostoresystem.com/>, Stand: 05.01.2018.
- [Bah14] BAHDANAU, D.; CHO, K. UND BENGIO, Y.: *Neural Machine Translation by Jointly Learning to Align and Translate*, archivx.org, <http://arxiv.org/pdf/1409.0473v7>, 2014.
- [Bec20] BECKHOFF AUTOMATION GMBH & Co. KG (HRSG.): *CX 2030*, <https://www.beckhoff.de/CX2030/>, Stand: 14.02.2020.
- [BMB13] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG (HRSG.): *Zukunftsbild Industrie 4.0*, https://www.bmbf.de/pub/Zukunftsbild_Industrie_4.0.pdf, Stand: 10.01.2018.
- [CAN20] CAN IN AUTOMATION (HRSG.): *CiA@ 402 series: CANopen device profile for drives and motion control*, <https://www.can-cia.org/can-knowledge/canopen/cia402/#>, Stand: 03.10.2020.
- [Cel20] CELLUMATION (HRSG.): *celluveyor - Omnidirectional Cellular Conveyor*, <https://www.youtube.com/watch?v=mBCcct8bZs>, Stand: 25.03.2020.
- [Cof71] COFFMAN, E. G.; ELPHICK, M. UND SHOSHANI, A.: *System Deadlocks*, ACM Computing Surveys (CSUR), Vol. 3, Nr. 2, S. 67–78, <https://www.doi.org/10.1145/356586.356588>, 1971.

- [Cro20] CROWN EQUIPMENT CORPORATION (HRSG.): *Automated Warehouse Solutions*, <http://www.crown.com/en-us/automation.html>, Stand: 24.03.2020.
- [Dan59] DANTZIG, G. B. UND RAMSER, J. H.: *The Truck Dispatching Problem*, MANAGEMENT SCIENCE, Vol. 6, Nr. 1, S. 80–91, <https://doi.org/10.1287/mnsc.6.1.80>, 1959.
- [Dij59] DIJKSTRA, E. W.: *A note on two problems in connexion with graphs*, Numerische Mathematik, Vol. 1, Nr. 1, S. 269–271, <https://www.doi.org/10.1007/BF01386390>, 1959.
- [Eid06] EIDSON, J. C.: *Measurement, Control, and Communication Using IEEE 1588*, Berlin/Heidelberg: Springer-Verlag, ISBN: 1-84628-250-0, 2006.
- [Fle20] FLEXLOG GMBH (HRSG.): *Dezentral steuerbar – Modulbaukasten mit FlexTechnology*, www.flexlog.de/, Stand: 03.10.2020.
- [FLT20] FL TECHNOLOGY GMBH (HRSG.): *Fluidlogistics*, <https://fluidlogistics.com/>, Stand: 12.09.2020.
- [For18] PROMOTORENGRUPPE KOMMUNIKATION DER FORSCHUNGSUNION WIRTSCHAFT – WISSENSCHAFT (HRSG.): *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*, https://www.bmbf.de/files/Umsetzungsempfehlungen_Industrie4_0.pdf, Stand: 10.02.2018.
- [Fra17] FRAUNHOFER-INSTITUT FÜR MATERIALFLUSS UND LOGISTIK (HRSG.): *Zellulare Transportsysteme - Schwarmintelligenz für die Logistik*, <https://www.industrie40.iml.fraunhofer.de/de/ergebnisse/schwarmintelligenz.html>, Stand: 21.11.2017.
- [Fur10] FURMANS, K.; SCHÖNUNG, F. UND GUE, K. R.: *Plug-and-work material handling systems*, Progress in Material Handling Research, S. 132–142, https://digitalcommons.georgiasouthern.edu/pmhr_2010/1, 2010.
- [Gue12] GUE, K. R.; ULUDAG, O. UND FURMANS, K.: *A High-Density System for Carton Sequencing*, Proceedings of the 6th International Scientific Symposium on Logistics (ISSL), https://www.researchgate.net/publication/264933685_A_High_Density_System_for_Carton_Sequencing, 2012.
- [Gue14] GUE, K. R.; FURMANS, K.; SEIBOLD, Z. UND ULUDAĞ, O.: *GridStore: A Puzzle-Based Storage System With Decentralized Control*, IEEE Trans. Automat.

- Sci. Eng, Vol. 11, Nr. 2, S. 429–438,
<https://www.doi.org/10.1109/TASE.2013.2278252>, 2014.
- [Har68] HART, P. E.; NILSSON, N. J. UND RAPHAEL, B.: *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*, IEEE Transactions on Systems Science and Cybernetics, Vol. 4, Nr. 2, S. 100–107,
<https://www.doi.org/10.1109/TSSC.1968.300136>, 1968.
- [Hen17] HENRICH PUBLIKATIONEN GMBH (HRSG.): *Weltmarkt Flurförderzeuge 2016*,
<http://www.logistik-journal.de/weltrangliste/>, Stand: 21.11.2017.
- [IEC14] IEC-61158:2014: *Industrial communication networks*, <https://webstore.iec.ch/publication/4625>, 2014.
- [IEE08] IEEE-1588:2008: *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, <https://standards.ieee.org/standard/1588-2008.html>, 2008.
- [IFL16] Institut für Fördertechnik und Logistiksysteme: *KARIS PRO – Autonomer Materialtransport für flexible Intralogistik*, Karlsruhe Institute of Technology, Karlsruhe, <http://karispro.de/Abschlussbericht%20KARIS%20PRO.pdf>, 2016.
- [ISO05] ISO-19501:2005: *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*,
<https://www.iso.org/standard/32620.html>, 2005.
- [Ito17] ITOH DENKI LTD. (HRSG.): *Nouveau: MCS Unit, Sorter Module Multidirectional and Multi-Format*, <http://www.itoh-denki.com/en/les-modules-2/mcs>, Stand: 17.11.2017.
- [Jod12] JODIN, D.; HOMPEL, M. ten: *Sortier- und Verteilsysteme: Grundlagen, Aufbau, Berechnung und Realisierung*, Berlin: Springer Vieweg, ISBN: 9783642312908, 2012.
- [Jun20] JUNGHEINRICH AG (HRSG.): *Fahrerlose Transportfahrzeuge | Jungheinrich*,
<https://www.jungheinrich.de/systeme/fahrerlose-transportfahrzeuge>, Stand: 03.10.2020.
- [Kim91] KIM, C. W. UND TANCHOCO, J. M. A.: *Conflict-free shortest-time bidirectional AGV routeing*, The International Journal of Production Research, Vol. 29,

- Nr. 12, S. 2377–2391, <https://www.doi.org/10.1080/00207549108948090>, 1991.
- [Kle13] KLEIN, N.: *The impact of decentral dispatching strategies on the performance of intralogistics transport systems*, Dissertation, Institut für Technische Logistik und Arbeitssysteme, TU Dresden, <http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-147739>, 2013.
- [Krü15] KRÜHN, T.: *Dezentrale, verteilte Steuerung flächiger Fördersysteme für den innerbetrieblichen Materialfluss*, Dissertation, Institut für Transport- und Automatisierungstechnik, Leibniz Universität Hannover, <http://www.tewiss-verlag.de/katalog/details/?isbn=978-3-95900-014-7>, 2015.
- [Krü16] KRÜHN, T.; SOHRT, S. UND OVERMEYER, L.: *Mechanical feasibility and decentralized control algorithms of small-scale, multi-directional transport modules*, Logistics Research, Vol. 9, Nr. 1, <https://www.doi.org/10.1007/s12159-016-0143-x>, 2016.
- [Lam78] LAMPORT, L.: *Time, clocks, and the ordering of events in a distributed system*, Communications of the ACM, Vol. 21, Nr. 7, S. 558–565, <https://amturing.acm.org/p558-lamport.pdf>, 1978.
- [Lan09] LANDAU, E.: *Handbuch der Lehre von der Verteilung der Primzahlen*, Leipzig und Berlin: Druck und Verlag von B.G.Teubner, ISBN: 978-0-8218-2650-8, 1909.
- [Lea06] LE-ANH, T. UND KOSTER, M.B.M. de: *A review of design and control of automated guided vehicle systems*, European Journal of Operational Research, Vol. 171, Nr. 1, S. 1–23, <https://www.doi.org/10.1016/j.ejor.2005.01.036>, 2006.
- [Lin20] LINDE MATERIAL HANDLING GMBH (HRSG.): *Automatisierter Niederhubwagen T-Matic*, <https://www.linde-mh.de/>, Stand: 25.03.2020.
- [Lit61] LITTLE, J. D. C.: *A Proof for the Queuing Formula: $L = \lambda W$* , Operations Research, Vol. 9, Nr. 3, S. 383–387, <https://www.doi.org/10.1287/opre.9.3.383>, 1961.
- [Luk05] LUKE, S.; CIOFFI, C.; PANAIT, L.; SULLIVAN, K. UND BALAN, G.: *MASON: A Multiagent Simulation Environment*, Simulation, Vol. 81, Nr. 7, S. 517–527, <https://www.doi.org/10.1177/0037549705058073>, 2005.

- [Mad96] MADRAS, N.; SLADE, G.: *The Self-Avoiding Walk*, Boston, MA: Birkhäuser Boston, ISBN: 9780817638917, 1996.
- [Mas19] LUKE S. (HRSG.): *MASON*, <https://cs.gmu.edu/~eclab/projects/mason/>, Stand: 28.11.2019.
- [Mat98] MATSUMOTO, M. UND NISHIMURA, T.: *Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator*, ACM Transactions on Modeling and Computer Simulation, Vol. 8, Nr. 1, S. 3–30, <https://www.doi.org/10.1145/272991.272995>, 1998.
- [May09] MAYER, S. H.: *Development of a completely decentralized control system for modular continuous conveyor systems*, Dissertation, Institut für Fördertechnik und Logistiksysteme (IFL), Karlsruhe Institute of Technology, <https://publikationen.bibliothek.kit.edu/1000011463>, 2009.
- [Maz05] MAZA, S. UND CASTAGNA, P.: *A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles*, Computers in Industry, Vol. 56, Nr. 7, S. 719–733, <https://www.doi.org/10.1016/j.compind.2005.03.003>, 2005.
- [Möh05] MÖHRING, R. H.; KÖHLER, E.; GAWRILOW, E. UND STENZEL, B.: *Conflict-free Real-time AGV Routing*, Operations Research Proceedings 2004, Vol. 2004, https://link.springer.com/chapter/10.1007/3-540-27679-3_3, 2005.
- [MöI08] MÖLLER, N.: *Bestimmung der Wirtschaftlichkeit wandlungsfähiger Produktionssysteme*, München: Utzverlag, ISBN: 9783831607785, 2008.
- [Nyh08] NYHUIS, P.; HEINEN, T.; REINHART, G.; RIMPAU, C.; ABELE, E. UND WÖRN, A.: *Wandlungsfähige Produktionssysteme: Theoretischer Hintergrund zur Wandlungsfähigkeit von Produktionssystemen*, wt-online, Nr. 1/2, S. 85–91, [https://www.werkstattstechnik.de/wt/article.php?data\[article_id\]=39785](https://www.werkstattstechnik.de/wt/article.php?data[article_id]=39785), 2008.
- [Ove17a] OVERMEYER, L.; KRÜHN, T.; SOHRT, S.; ROLOFF, F.; VAHLSING, D.-A.: *Transportmodul und Transportmodulgruppe*, DE 10 2016 107 665 A1, Deutsches Patent- und Markenamt, 26.10.2017.
- [Ove17b] OVERMEYER, L, STICHWEH, H. (HRSG.) *Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, 12.2017.

- [Pin20] PINA MERKERT (HRSG.): *Maschinelle Übersetzer: DeepL macht Google Translate Konkurrenz*, <https://www.heise.de/newsticker/meldung/Maschinelle-Uebersetzer-DeepL-macht-Google-Translate-Konkurrenz-3813882.html>, Stand: 26.03.2020.
- [Roc20] ROCLA OY (HRSG.): *Rocla automated guided vehicles brochures*, <https://www.rocla-agv.com/en/automated-guided-vehicles>, Stand: 25.03.2020.
- [Sch09] SCHMIDT, T. UND SCHULZE, F.: *Future Approaches to Meet Complexity Requirements in Material Handling Systems*, FME Transactions, Vol. 37, Nr. 4, S. 159–166, http://www.mas.bg.ac.rs/_media/istrazivanje/fme/vol37/4/01_tschmidt.pdf, 2009.
- [Sch15] SCHWAB, M.: *A decentralized control strategy for high density material flow systems with automated guided vehicles*, Dissertation, Institut für Förder-technik und Logistiksysteme (IFL), Karlsruhe Institute of Technology, <http://dx.doi.org/10.5445/KSP/1000047227>, 2015.
- [Sei16a] SEIBOLD, Z. UND FURMANS, K.: *GridSorter – Logische Zeit in dezentral gesteuerten Materialflusssystemen*, Logistics Journal: Proceedings, Vol. 2014, https://www.doi.org/10.2195/LJ_PROC_SEIBOLD_DE_201411_01, 2016.
- [Sei16b] SEIBOLD, Z.: *Logical time for decentralized control of material handling systems*, Dissertation, Karlsruher Institut für Technologie (KIT), Institut für Förder-technik und Logistiksysteme (IFL), <http://dx.doi.org/10.5445/KSP/1000057838>, 2016.
- [Set90] SETHI, A. K. UND SETHI, S. P.: *Flexibility in manufacturing: A survey*, International Journal of Flexible Manufacturing Systems, Vol. 2, Nr. 4, S. 289–328, <https://www.doi.org/10.1007/BF00186471>, 1990.
- [Shc17a] SHCHEKUTIN, N.; SOHRT, S.; OVERMEYER, L.: *Multi-objective layout optimization for material flow system with decentralized and scalable control* in 13. *Fachkolloquium Logistik der Wissenschaftlichen Gesellschaft für Technische Logistik (WGTL)*, Wissenschaftliche Gesellschaft für Technische Logistik (Hrsg.) S. 141–153, 2017.
- [Shc17b] SHCHEKUTIN, N.; SOHRT, S. UND OVERMEYER, L.: *Multi-objective layout optimization for material flow system with decentralized and scalable control*,

- Logistics Journal : Proceedings, Vol. 2017, Nr. 10, https://www.doi.org/10.2195/LJ_PROC_SHCHEKUTIN_EN_201710_01, 2017.
- [Shc19] SHCHEKUTIN, N.: *Layout Optimization for Cognitive Material Flow Systems*, Dissertation, Institut für Transport- und Automatisierungstechnik, Leibniz Universität Hannover, <https://www.tewiss-verlag.de/katalog/details/?isbn=978-3-95900-330-8>, 2019.
- [Sig17] SIGMATEK GMBH & Co KG (HRSG.): *LogiMover: Doppelkufensystem von EISENMANN - SIGMATEK*, <http://www.sigmatek-automation.com/de/unternehmen/news-presse/aktuelle-news/logimover-das-doppelkufensystem-von-eisenmann/>, Stand: 21.11.2017.
- [Smo10] SMOLIC-ROCAK, N.; BOGDAN, S.; KOVACIC, Z. UND PETROVIC, T.: *Time Windows Based Dynamic Routing in Multi-AGV Systems*, IEEE Transactions on Automation Science and Engineering, Vol. 7, Nr. 1, S. 151–155, <https://www.doi.org/10.1109/TASE.2009.2016350>, 2010.
- [Soh14] S.SOHRT, Z. SEIBOLD, T. KRÜHN, L.PRÖSSDORF, L. OVERMEYER, K. FURMANS: *Buffering Algorithms for Modular, Decentralized Controlled Material Handling Systems*, 1st Symposium on Automated Systems and Technologies (AST), Berichte aus dem ITA Band 4/2014, S. 29–36, https://www.ast.uni-hannover.de/fileadmin/ast/AST2014/Proceedings2014/16_10_2014_Buffering_Algorithms_for_Modular__Decentralized_Controlled_Material_Handling_Systems_Sohrt.pdf, 2014.
- [Soh16a] SOHRT, S.; SHCHEKUTIN, N. UND OVERMEYER, L.: *Steuerung von kleinskaligen Fördermodulen*, Logistics Journal Proceedings, Vol. 2016, Nr. 10, https://www.doi.org/10.2195/lj_Proc_sohrt_de_201610_01, 2016.
- [Soh16b] SOHRT, S.; SHCHEKUTIN, N.; OVERMEYER, L.: *Dezentrales Routing in wandlungsfähigen Förderanlagen* in *Gastvortragsreihe Logistik 2016*, Schenk, M. (Hrsg.) <https://www.iff.fraunhofer.de/content/dam/iff/de/dokumente/publikationen/gastvortragsreihe-logistik-2016-tagungsband-fraunhofer-iff.pdf>, S. 33–39, 2016.
- [Soh17a] SOHRT, S.; HEINKE, A.; SHCHEKUTIN, N.; EILERT, B.; OVERMEYER, L.; KRÜHN, T.: *Kleinskalige, cyber-physische Fördertechnik* in *Handbuch Industrie 4.0 Bd.3: Logistik*, Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M. ten (Hrsg.)

2. Auflage, Berlin Heidelberg: Springer, ISBN: 9783662532508, S. 21–44, 2017.
- [Soh17b] SOHRT, S.: *Routingalgorithmen für dynamische Materialflüsse in vernetzten, kognitiven Produktionssystemen* in *Berichte des TEWISS Verlags, Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Overmeyer, L.; Stichweh, H. (Hrsg.) Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, S. 59–65, 2017.
- [Soh17c] SOHRT, S.; SHCHEKUTIN, N.; HEINKE, A.: *Validierung des vernetzten, kognitiven Produktionssystems im netkoPs.Lab* in *Berichte des TEWISS Verlags, Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Overmeyer, L.; Stichweh, H. (Hrsg.) Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, S. 95–97, 2017.
- [Soh17d] SOHRT, S.; DEPPE, F.-D.; SHCHEKUTIN, N.; HEINKE, A.: *Aufbau von vernetzten, kognitiven Produktionssystemen* in *Berichte des TEWISS Verlags, Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Overmeyer, L.; Stichweh, H. (Hrsg.) Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, S. 9–24, 2017.
- [Soh19] SOHRT, S.; OVERMEYER, L.: *Decentralized Routing Algorithm with Physical Time Windows for Modular Conveyors*, <https://engrxiv.org/f4eq6/>, <https://www.doi.org/10.31224/osf.io/f4eq6>, 2019.
- [Soh20a] SOHRT, S. UND OVERMEYER, L.: *Decentralized Routing Algorithm with Physical Time Windows for Modular Conveyors*, *Logistics Research*, Vol. 13, Nr. 1, https://www.doi.org/10.23773/2020_8, 2020.
- [Soh20b] SOHRT, S.; HEINKE, A.; SHCHEKUTIN, N.; EILERT, B.; OVERMEYER, L.: *Kleinskalige, cyber-physische Fördertechnik* in *Springer Reference Technik, Handbuch Industrie 4.0 Bd.3: Logistik*, Hoppel, M. ten; Bauernhansl, T.; Vogel-Heuser, B. (Hrsg.) 3. Auflage, Berlin: Springer Berlin; Springer Vieweg, ISBN: 9783662585290, 2020.
- [SSI17] SSI SCHÄFER FRITZ SCHÄFER GMBH (HRSG.): *Fahrerlose Transportsysteme (FTS)*, <https://www.ssi-schaefer.com/de-de/produkte/foerdern-transportieren/fahrerlose-transportsysteme>, Stand: 21.11.2017.
- [Sti16] STICHWEH, H.; THEßELING, M.; SOHRT, S.; HEINKE, A.; OVERMEYER, L.: *Intelligent routen, fördern und verteilen: Die Conveyor Matrix für die kognitive*

- Produktion der Zukunft*. in *VDI-Berichte*, vol. 2275, 25. *Deutscher Materialfluss-Kongress*, VDI - Gemeinschaft Produktion und Logistik (Hrsg.) Düsseldorf: VDI Verlag GmbH, ISBN: 978-3-18-092275-1, S. 127–142, 2016.
- [Sto12] STOLL, T.: *Dezentral gesteuerter Aufbau von Stetigförderern mittels autonomer Materialflusselemente*, Dissertation, Institut für Fördertechnik und Logistiksysteme (IFL), Institut für Fördertechnik und Logistiksysteme (IFL), <http://dx.doi.org/10.5445/KSP/1000028697>, 2012.
- [Süd18] MÜLLER, B.: *Elf Millionen Pakete täglich*, Süddeutsche Zeitung, <https://www.sueddeutsche.de/wirtschaft/deutsche-post-elf-millionen-pakete-taeglich-1.4199018>, 2018.
- [Tad12a] TADAKUMA, K.; TADAKUMA, R.; IOKA, K.; KUDO, T.; TAKAGI, M.; TSUMAKI, Y.; HIGASHIMORI, M. UND KANEKO, M.: *Omnidirectional driving gears and their input mechanism with passive rollers*, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, S. 2881–2888, <https://www.doi.org/10.1109/IROS.2012.6385464>, 2012.
- [Tad12b] TADAKUMA, K.; TADAKUMA, R.; IOKA, K.; KUDO, T.; TAKAGI, M.; TSUMAKI, Y.; HIGASHIMORI, M. UND KANEKO, M.: *Additional manipulating function for limited narrow space with omnidirectional driving gear*, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, S. 5438–5439, <https://www.doi.org/10.1109/IROS.2012.6386290>, 2012.
- [Tan11] TANENBAUM, A. S.; WETHERALL, D.: *Computer networks*, Boston, Mass.: Pearson, ISBN: 9780132126953, 2011.
- [Tan15] TANENBAUM, A. S.; BOS, H.: *Modern operating systems*, Boston: Pearson, ISBN: 1-292-06142-1, 2015.
- [Tan16] TANENBAUM, A. S.; VAN STEEN, M.: *Distributed systems: Principles and paradigms*, Leiden: Maarten van Steen, ISBN: 9789081540605, 2016.
- [The16] THESSELING, M.; LIU, T. UND WENDT, H. J.: *Combining an External Rotor Motor with Vernier Concept for Drives in Intralogistics*, PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management, S. 1–7, <http://ieeexplore.ieee.org/document/7499598/>, 2016.

- [TI19] TEXAS INSTRUMENTS (HRSG.): *AM3358*, <https://www.ti.com/product/AM3358>, Stand: 14.02.2020.
- [Tie20] TIETJEN, T.; DECKER, A.: *FMEA-Praxis: Einstieg in die Risikoabschätzung von Produkten, Prozessen und Systemen*, München: Hanser, ISBN: 9783446462113, 2020.
- [TLP20] THE LINUX PTP PROJECT (HRSG.): *PTP IEEE 1588 stack for Linux*, <http://linuxptp.sourceforge.net/>, Stand: 26.03.2020.
- [Toy20] TOYOTA MATERIAL HANDLING DEUTSCHLAND GMBH (HRSG.): *Automatisierung*, <https://toyota-forklifts.de/automatisierung/automatisierung/>, Stand: 25.03.2020.
- [Ull14] ULLRICH, G.: *Fahrerlose Transportsysteme: Eine Fibel - mit Praxisanwendungen - zur Technik - für die Planung*, Wiesbaden: Springer Fachmedien Wiesbaden, ISBN: 978-3-8348-2592-6, 2014.
- [Ulu14] ULUDAĞ, O.: *GridPick: A High Density Puzzle Based Order Picking System with Decentralized Control*, Dissertation, Auburn University, <https://etd.auburn.edu/handle/10415/3984>, 2014.
- [Uri13] URIARTE, C.; ROHDE, A.-K. UND KUNASCHK, S.: *Celluveyor - Ein hochflexibles und modulares Förder- und Positioniersystem auf Basis omnidirektionaler Antriebstechnik*, 18. Magdeburger Logistiktage - Sichere und nachhaltige Logistik, <https://www.iff.fraunhofer.de/content/dam/iff/de/dokumente/publikationen/iff-wissenschaftstage-2013-logistik-tagungsband-fraunhofer-iff.pdf>, 2013.
- [Uri15] URIARTE, C.; THAMER, H. UND FREITAG, M.: *Fördertechnik aus der Zelle, Hebezeuge und Fördermittel 10*, https://www.technische-logistik.net/sites/default/files/Fachartikel/HF1015_Thamer_0.pdf, 2015.
- [VDA17] VDA-4500:11.2017: *Kleinladungsträger (KLT)-System Teil 1*, 11.2017.
- [Ven12] VENTZ, K.; HACHICHA, M. B.; RADOSAVAC, M.; KRÜHN, T. UND OVERMEYER, L.: *Aufbau hochfunktionaler Intralogistik-Knoten mittels kleinskaliger Module als Cognitive Conveyor*, Logistics Journal Proceedings, https://www.doi.org/10.2195/lj_Proc_ventz_de_201210_01, 2012.

- [Ven16] VENTZ, K.-U.: *Beitrag zur innovativen Gestaltung von Intralogistik durch Kopplung kleinskaliger Systeme*, Dissertation, Institut für Transport- und Automatisierungstechnik, Leibniz Universität Hannover, <http://www.tewiss-verlag.de/katalog/details/?isbn=978-3-95900-083-3>, 2016.
- [Vis06] VIS, I. F.A.: *Survey of research in the design and control of automated guided vehicle systems*, European Journal of Operational Research, Vol. 170, Nr. 3, S. 677–709, <https://www.doi.org/10.1016/j.ejor.2004.09.020>, 2006.
- [Viv10] VIVALDINI, K. C.T.; GALDAMES, J. P.M.; PASQUAL, T. B.; SOBRAL, R. M.; ARAÚJO, R. C.; BECKER, M. UND CAURIN, G. A.P.: *Automatic routing system for intelligent warehouses*, IEEE Int. Conference on Robotics and Automation, Vol. 1, S. 93–98, 2010.
- [Viv15] VIVALDINI, K. C. T.; ROCHA, L. F.; BECKER, M. UND MOREIRA, A. P.: *Comprehensive Review of the Dispatching, Scheduling and Routing of AGVs*, CONTROL'2014 - Proc. of the 11th Port. Conf. on Autom. Control, https://www.doi.org/10.1007/978-3-319-10380-8_48, 2015.
- [Wek17] WEKA BUSINESS MEDIEN GMBH (HRSG.): *Festo - Palettiersystem Motion Cube*, <https://www.handling.de/video---festo---palettiersystem-motion-cube.htm>, Stand: 20.11.2017.
- [Wes09] WESTKÄMPER, E.; ZAHN, E.: *Wandlungsfähige Produktionsunternehmen: Das Stuttgarter Unternehmensmodell*, Berlin: Springer, ISBN: 3540688900, 2009.

Eigene wissenschaftliche Publikationen

KRÜHN, T.; SOHRT, S. UND OVERMEYER, L.: *Mechanical feasibility and decentralized control algorithms of small-scale, multi-directional transport modules*, Logistics Research, Vol. 9, Nr. 1, <https://www.doi.org/10.1007/s12159-016-0143-x>, 2016.

OVERMEYER, L.; KRÜHN, T.; SOHRT, S.; ROLOFF, F.; VAHLSING, D.-A.: *Transportmodul und Transportmodulgruppe*, DE 10 2016 107 665 A1, Deutsches Patent- und Markenamt, 26.10.2017.

SHCHEKUTIN, N.; SOHRT, S.; OVERMEYER, L.: *Multi-objective layout optimization for material flow system with decentralized and scalable control* in *13. Fachkolloquium Logistik der Wissenschaftlichen Gesellschaft für Technische Logistik (WGTL)*, Wissenschaftliche Gesellschaft für Technische Logistik (Hrsg.) S. 141–153, 2017.

SHCHEKUTIN, N.; SOHRT, S. UND OVERMEYER, L.: *Multi-objective layout optimization for material flow system with decentralized and scalable control*, Logistics Journal : Proceedings, Vol. 2017, Nr. 10, https://www.doi.org/10.2195/LJ_PROC_SHCHEKUTIN_EN_201710_01, 2017.

S.SOHRT, Z. SEIBOLD, T. KRÜHN, L. PRÖSSDORF, L. OVERMEYER, K. FURMANS: *Buffering Algorithms for Modular, Decentralized Controlled Material Handling Systems*, 1st Symposium on Automated Systems and Technologies (AST), Berichte aus dem ITA Band 4/2014, S. 29–36, https://www.ast.uni-hannover.de/fileadmin/ast/AST2014/Proceedings2014/16_10_2014_Buffering_Algorithms_for_Modular__Decentralized_Controlled_Material_Handling_Systems_Sohrt.pdf, 2014.

SOHRT, S.; SHCHEKUTIN, N.; OVERMEYER, L.: *Dezentrales Routing in wandlungsfähigen Förderanlagen* in *Gastvortragsreihe Logistik 2016*, Schenk, M. (Hrsg.) <https://www.iff.fraunhofer.de/content/dam/iff/de/dokumente/publikationen/gastvortragsreihe-logistik-2016-tagungsband-fraunhofer-iff.pdf>, S. 33–39, 2016.

SOHRT, S.; SHCHEKUTIN, N. UND OVERMEYER, L.: *Steuerung von kleinskaligen Fördermodulen*, Logistics Journal Proceedings, Vol. 2016, Nr. 10, https://www.doi.org/10.2195/lj_Proc_sohrt_de_201610_01, 2016.

SOHRT, S.; HEINKE, A.; SHCHEKUTIN, N.; EILERT, B.; OVERMEYER, L.; KRÜHN, T.: *Kleinskalige, cyber-physische Fördertechnik in Handbuch Industrie 4.0 Bd.3: Logistik*, Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M. ten (Hrsg.) 2. Auflage, Berlin Heidelberg: Springer, ISBN: 9783662532508, S. 21–44, 2017.

SOHRT, S.; SHCHEKUTIN, N.; HEINKE, A.: *Validierung des vernetzten, kognitiven Produktionssystems im netkoPs.Lab in Berichte des TEWISS Verlags, Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Overmeyer, L.; Stichweh, H. (Hrsg.) Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, S. 95–97, 2017.

SOHRT, S.: *Routingalgorithmen für dynamische Materialflüsse in vernetzten, kognitiven Produktionssystemen in Berichte des TEWISS Verlags, Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Overmeyer, L.; Stichweh, H. (Hrsg.) Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, S. 59–65, 2017.

SOHRT, S.; DEPPE, F.-D.; SHCHEKUTIN, N.; HEINKE, A.: *Aufbau von vernetzten, kognitiven Produktionssystemen in Berichte des TEWISS Verlags, Vernetzte, kognitive Produktionssysteme: Abschlussbericht*, Overmeyer, L.; Stichweh, H. (Hrsg.) Garbsen: TEWISS - Technik und Wissen GmbH, ISBN: 978-3-95900-165-6, S. 9–24, 2017.

SOHRT, S.; OVERMEYER, L.: *Decentralized Routing Algorithm with Physical Time Windows for Modular Conveyors*, <https://engrxiv.org/f4eq6/>, <https://www.doi.org/10.31224/osf.io/f4eq6>, 2019.

SOHRT, S.; HEINKE, A.; SHCHEKUTIN, N.; EILERT, B.; OVERMEYER, L.: *Kleinskalige, cyber-physische Fördertechnik in Springer Reference Technik, Handbuch Industrie 4.0 Bd.3: Logistik*, Hompel, M. ten; Bauernhansl, T.; Vogel-Heuser, B. (Hrsg.) 3. Auflage, Berlin: Springer Berlin; Springer Vieweg, ISBN: 9783662585290, 2020.

SOHRT, S. UND OVERMEYER, L.: *Decentralized Routing Algorithm with Physical Time Windows for Modular Conveyors*, *Logistics Research*, Vol. 13, Nr. 1, https://www.doi.org/10.23773/2020_8, 2020.

STICHWEH, H.; THEBELING, M.; SOHRT, S.; HEINKE, A.; OVERMEYER, L.: *Intelligent routen, fördern und verteilen: Die Conveyor Matrix für die kognitive Produktion der Zukunft*. in *VDI-Berichte*, vol. 2275, 25. *Deutscher Materialfluss-Kongress*, VDI - Gemeinschaft Produktion und Logistik (Hrsg.) Düsseldorf: VDI Verlag GmbH, ISBN: 978-3-18-092275-1, S. 127–142, 2016.

Betreute studentische Arbeiten

- [Boc15] BOCK, M.; SOHRT, S.: *Entwicklung und Implementation eines Modelles von Transporteinheiten in einer Simulationsumgebung*, Hannover, 01.07.2015 (Bachelorarbeit)
- [Ehl19] EHLERS, S.; HOROWITZ, R.; WRIGHT, M.; SOHRT, S.: *Traffic Queue Length and Pressure Estimation for Road Networks with Geometric Deep Learning Algorithms*, Hannover, 07.02.2019 (Studienarbeit)
- [Gla15] GLAAS, E.; SOHRT, S.: *Entwicklung eines Routings zur Berechnung von direkten Transportbahnen auf einer Fördermatrix*, Hannover, 27.08.2015 (Studienarbeit)
- [Kun19] KUNDE, C.; SOHRT, S.: *Erstellung eines Programms zur Ansteuerung einer Fördermatrix*, Hannover, 30.09.2019 (Bachelorarbeit)
- [Mel16] MELZIAN, L.; SOHRT, S.: *Optimierung des Materialflusssystems zur Lagerung von unvulkanisierten Reifen und deren Anlieferung an die Reifenpresse*, Hannover, 31.08.2016 (Masterarbeit)
- [Pap16] PAPE, D.; SOHRT, S.: *Berechnung und Adressierung von Steuerbefehlen für Fördersysteme mit dezentralen, räumlich verteilten Steuerungen*, Hannover, 01.06.2015 (Studienarbeit)
- [Rum17] RUMBERG, L.; SOHRT, S.: *Erweiterung von bestehenden Routingalgorithmen um die Funktion Routen zusammenzufassen*, Hannover, 30.09.2017 (Studienarbeit)
- [Tar15] TARANETC, I.; SOHRT, S.: *Development and implementation of a transport model for continuous conveyors in a simulation environment*, Hannover, 22.01.2015 (Masterarbeit)
- [Tka16] TKANY, C.; SOHRT, S.: *Anpassung der Schnittstellen von kleinskaligen, modularen Fördermodulen zur Integration in eine Förderanlage*, Hannover, 22.12.2016 (Bachelorarbeit)

Zur verwendeten Sprache

In dieser Arbeit wird eine geschlechtergerechte Sprache verwendet, um die Gleichstellung der Geschlechter zu unterstützen.

Weiterhin entschuldigt sich der Autor für die Verwendung des rassistischen Begriffspaares „Master/Slave“ in zurückliegenden Veröffentlichungen. In dieser Arbeit wird dieses Begriffspaar nicht mehr verwendet. Der Autor folgt hiermit den Entscheidungen von Firmen wie z.B. der Microsoft Corporation oder der Amazon Corporation.

Der Autor bedankt sich bei seiner Freundin für die Hinweise zur Verwendung einer geschlechtergerechten Sprache und für die Hinweise zur Vermeidung von rassistischen Begriffen.

Anhang

A1 - Technische Eigenschaften der verwendeten Fördermodule nach [The16]

Tabelle A.1: *Technische Eigenschaften der verwendeten Fördermodule nach [The16]*

Eigenschaft	Wert
Abmessungen	295 x 60 x 60 mm
Durchmesser der Förderrolle	50 mm
Maximale Flächenlast	125 kg/m ²
Motorbauart	Synchronmaschine
Nenn Drehzahl	750 min ⁻¹
Nennmoment	0,45 Nm
Effektiver Nennstrom	1,4 A

A2 - Auswirkung einer inhomogenen Massenverteilung oder inhomogener Reibungskoeffizienten auf den Transportprozess

Der Transportprozess wird erheblich gestört, wenn es zu einem Verkanten der Transporteinheiten kommt. Zu einem Verkanten der Transporteinheiten kommt es, wenn die Transporteinheiten durch die Fördermatrix unbeabsichtigt gedreht werden. Dieses Verdrehen geschieht, wenn während der translatorischen Bewegung zusätzlich noch ein Drehmoment auf die Transporteinheit einwirkt. Dieses Drehmoment tritt auf, wenn einzelnen Module mehr Kraft auf die Transporteinheit übertragen als andere. Die maximal übertragbare Kraft eines Moduls wird durch die Haftreibung begrenzt. Die Haftreibung hängt hierbei von der Massenverteilung der Transporteinheit und dem Reibungskoeffizienten ab.

Bei einer inhomogenen Massenverteilung ist die maximal übertragbare Kraft durch die einzelnen Module ebenfalls inhomogen verteilt. Falls nun durch einzelne Module die maximal übertragbare Kraft lokal überschritten wird, kommt es zu einer unbeabsichtigten Verdrehung der Transporteinheit.

Ein weiterer Grund für ein unzulässiges Verkanten liegt vor, wenn der Reibungskoeffizient der Unterseite der Transporteinheit nicht homogen ist. Die Größe des Reibungskoeffizienten begrenzt die maximale Kraft, die ein Fördermodul auf die Transporteinheit übertragen kann. Wenn der Reibungskoeffizient an einer Stelle der Transporteinheit wesentlich kleiner ist, als an den restlichen Stellen der Transporteinheit, kann nicht mehr überall die gleiche Kraft auf die Transporteinheit übertragen werden. Dies kann beispielsweise der Fall sein, wenn auf der Unterseite der Transporteinheit ein Stück Klebeband vorhanden ist. Falls die maximal übertragbare Kraft von einzelnen Modulen überschritten wird, kommt es zu einer unbeabsichtigten Verdrehung der Transporteinheit.

Eine unbeabsichtigte Verdrehung kann durch die Erweiterung der Modulsteuerung um eine Regelung kompensiert werden. Die Regelung muss hierbei während des laufenden Transportprozesses die Verdrehung erkennen und der Verdrehung entgegenwirken. Durch die vorher reservierten Zeitfenster verfügt jedes Modul über die Information, wann es zum ersten Mal eine Transporteinheit detektieren soll. Falls ein Modul eine Transporteinheit früher oder später detektiert, kann es seine Fördergeschwindigkeit entsprechend verringern oder vergrößern. Hierdurch wird der unbeabsichtigten Verdrehung entgegengewirkt.

Diese Regelung kann noch weiter verbessert werden, indem sie um ein Beobachtermodell erweitert wird: Die Verdrehung einer Transporteinheit mit einer inhomogenen Massenverteilung ist annähernd konstant. Module, die bereits die Transporteinheit bewegt haben, speichern ab, wie stark sie regeln mussten. Diese Information wird an alle Module gesendet, die noch die Transporteinheit bewegen müssen. Somit erhalten die Module schon vor dem Eintreffen der Transporteinheit die Information, wie stark sie ihre Fördergeschwindigkeit verringern bzw. vergrößern müssen und können entsprechend schneller reagieren.

A3 - Inhalt der Routingeinträge und der Nachrichten

Hinweis: Die Datentypen sind spezifiziert in der Dokumentation für die Java SE 7. Weiterhin gibt es keine gesonderte Variable für die Priorität. Stattdessen wird die Paket-ID zur Priorisierung genutzt.

Tabelle A.2: *Inhalt eines Routingeintrages*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID
destinationID	int	ID der Senke
predeccesorConveyorID	int	ID des Vorgängers
successorConveyorID	int	ID des Nachfolgers
arrivalTime	double	Zeitpunkt zu dem der Transport beginnt
arrivesAtCenterTime	double	Zeitpunkt der Erreichung des Mittelpunktes des Moduls
leftCenterTime	double	Zeitpunkt des Verlassens des Mittelpunktes des Moduls
departedTime	double	Zeitpunkt zu dem der Transport endet
Status	enum	Status des Routingeintrages
conveyInDirection	enum	Richtung des Vorgängers
conveyOutDirection	enum	Richtung des Nachfolgers
TTL	double	Lebenszeit des Routingeintrages
iterationCounter	int	Iterationszähler
creationTimePredecessor	double	Erstellungszeitpunkt auf dem Vorgänger
creationTime	double	Erstellungszeitpunkt

Tabelle A.3: *Inhalt der ANFRAGE-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID
iterationCounter	int	Iterationszähler
creationTime	double	Erstellungszeitpunkt
destinationID	int	ID der Senke
senderID	int	ID des anfragenden Moduls
plannedArrivalTime	double	Geplante Ankunftszeit
TTL	double	Lebenszeit

Tabelle A.4: *Inhalt der HINFÄLLIG-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID
iterationCounter	int	Iterationszähler
creationTime	double	Erstellungszeitpunkt

Tabelle A.5: *Inhalt der ABHLEHNUNGS-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID
iterationCounter	int	Iterationszähler
creationTime	double	Erstellungszeitpunkt
senderID	int	ID des ablehnenden Moduls
earliestPossibleTime	double	Vorgeschlagene Zeit des ablehnenden Moduls

Tabelle A.6: *Inhalt der BESTÄTIGUNGS-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID
iterationCounter	int	Iterationszähler
creationTime	double	Erstellungszeitpunkt
grantedTime	double	Gewährte Ankunftszeit

Tabelle A.7: *Inhalt der RESERVIERUNGS-SPERRE-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID

Tabelle A.8: *Inhalt der RESERVIERUNGS-ENTSPERRE-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
packetID	int	Paket-ID

Tabelle A.9: *Inhalt der ABLAUFPLAN-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
schedule	schedule	Ablaufplan

Tabelle A.10: *Inhalt der ABLAUFPLAN-BESTÄTIGUNGS-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
ack	boolean	Bestätigung, ob der Ablaufplan angenommen wurde

Tabelle A.11: *Inhalt der ABLAUFPLAN-ANFRAGE-Nachricht*

Bezeichnung im Quellcode	Datentyp	Erläuterung
senderID	int	ID des Hauptmoduls

A4- Modifizierter Algorithmus zur Verhinderung vom Verhungern von Quellen

In diesem Anhang wird zunächst dargestellt, wie es zu dem Verhungern von Quellen kommen kann. Anschließend wird eine Modifikation des Algorithmus aus Kapitel 6 vorgestellt, um das Verhungern zu verhindern. Diese Modifikation kann sowohl für großskalige als auch kleinskalige Module umgesetzt werden. Zuletzt werden die Auswirkungen dieser Modifikationen auf die Laufzeit des Algorithmus analysiert.

Auf Grund des dezentralen Steuerungskonzeptes besitzt keine Quelle einen Überblick über den Zustand des Gesamtsystems. Auf Grund dieses fehlenden Überblickes kann es zu dem seltenen Fall kommen, dass eine Quelle niemals seine Routenreservierung erfolgreich abschließen kann. Dies wird an dem folgenden Beispiel verdeutlicht: Ein Modul m_n besitzt zwei Nachbarn: Die Module m_{n+1} und m_{n-1} . Das Modul m_{n+1} stellt hierbei den einzigen möglichen Weg zur Senke dar. Das Modul m_n sendet nun eine ANFRAGE-Nachricht an das Modul m_{n+1} . Da das Modul m_{n+1} bereits einen Routineeintrag im Zustand „bestätigt“ zu dem angefragten Zeitfenster besitzt, sendet es eine ABLEHNUNGS-Nachricht an das Modul m_n zurück. In dieser ABLEHNUNGS-Nachricht wird ein neuer möglicher Ankunftszeitpunkt mitgeschickt. Da nur über das Modul m_{n+1} die Senke erreicht werden kann, muss das Modul m_n das Zeitfenster entsprechend der vorgeschlagenen Zeit aus der ABLEHNUNGS-Nachricht anpassen. Durch die Anpassung überlappt aber nun der angepasste Routineeintrag mit einem anderen Routineeintrag, der den Zustand „bestätigt“ aufweist. Somit muss das Modul m_n den Routineeintrag löschen und selbst eine ABLEHNUNGS-Nachricht an sein Vorgängermodul m_{n-1} senden. Diese Abfolge von Ereignissen kann sich solange wiederholen, bis die Lebenszeit der Routenanfrage abgelaufen ist. Da es keine Garantie dafür gibt, dass diese gleiche Abfolge von Ereignissen bei der nächsten Routinganfrage nicht wieder auftritt, ist es möglich, dass eine Routinganfrage im System niemals erfolgreich abgeschlossen wird und somit „verhungert“ (Engl.: starvation).

Um dieses Verhungern zu verhindern, wird ein Synchronisierungsmechanismus eingeführt. Jede Quelle zählt die Anzahl der fehlgeschlagenen Routenreservierungen. Wenn eine vorher festgelegte Anzahl an Anfragen fehlgeschlagen ist, sendet die Quelle eine RESERVIERUNGS-SPERRE-Nachricht per Broadcast an alle anderen Quellen im Fördersystem. Teil der RESERVIERUNGS-SPERRE-Nachricht ist die eindeutige ID und die Priorität der Transporteinheit. Alle Quellen speichern alle empfangenen RESERVIERUNGS-SPERRE-Nachrichten. Bevor eine Quelle eine neue Reservierungsanfrage verschickt, überprüft dieser, ob die Priorität der eigenen Transporteinheit höher ist als die Priorität von allen empfangenen RESERVIERUNGS-SPERRE-

Nachrichten. Der modifizierte Sub-Algorithmus zum Starten von neuen Routinganfragen ist in Abbildung A.1 dargestellt. Das Senden, Empfangen und die Verarbeitung von RESERVIERUNGS-SPERRE und RESERVIERUNGS-ENTSPERRE Nachrichten ist identisch für kleinskalige und großskalige Module.

Wenn eine Quelle eine RESERVIERUNGS-SPERRE-Nachricht verschickt hat und nun eine BESTÄTIGUNGS-Nachricht für die entsprechende Transporteinheit erhält, wird eine RESERVIERUNGS-ENTSPERRE-Nachricht verschickt und der Zähler, der die Anzahl der fehlgeschlagenen Routenreservierungen zählt, wird wieder auf 0 zurückgesetzt. Daher ist es auch notwendig, den Sub-Algorithmus zur Verarbeitung von BESTÄTIGUNGS-Nachrichten zu modifizieren (siehe Abbildung A.2). Auch hier sind die Modifikationen sowohl für großskalige als auch kleinskalige Fördermodule identisch.

Die Verwendung von RESERVIERUNGS-SPERRE Nachrichten ist nur äußerst selten notwendig. In dem Kapitel 7 wurden über 29 Millionen Transporteinheiten bewegt, ohne dass auch nur eine einzige RESERVIERUNGS-SPERRE Nachricht versendet werden musste.

Da die Verarbeitung von RESERVIERUNGS-SPERRE und RESERVIERUNGS-ENTSPERRE Nachrichten konstant ist, haben beide Nachrichtentypen keinen Einfluss auf die Laufzeit des Algorithmus.

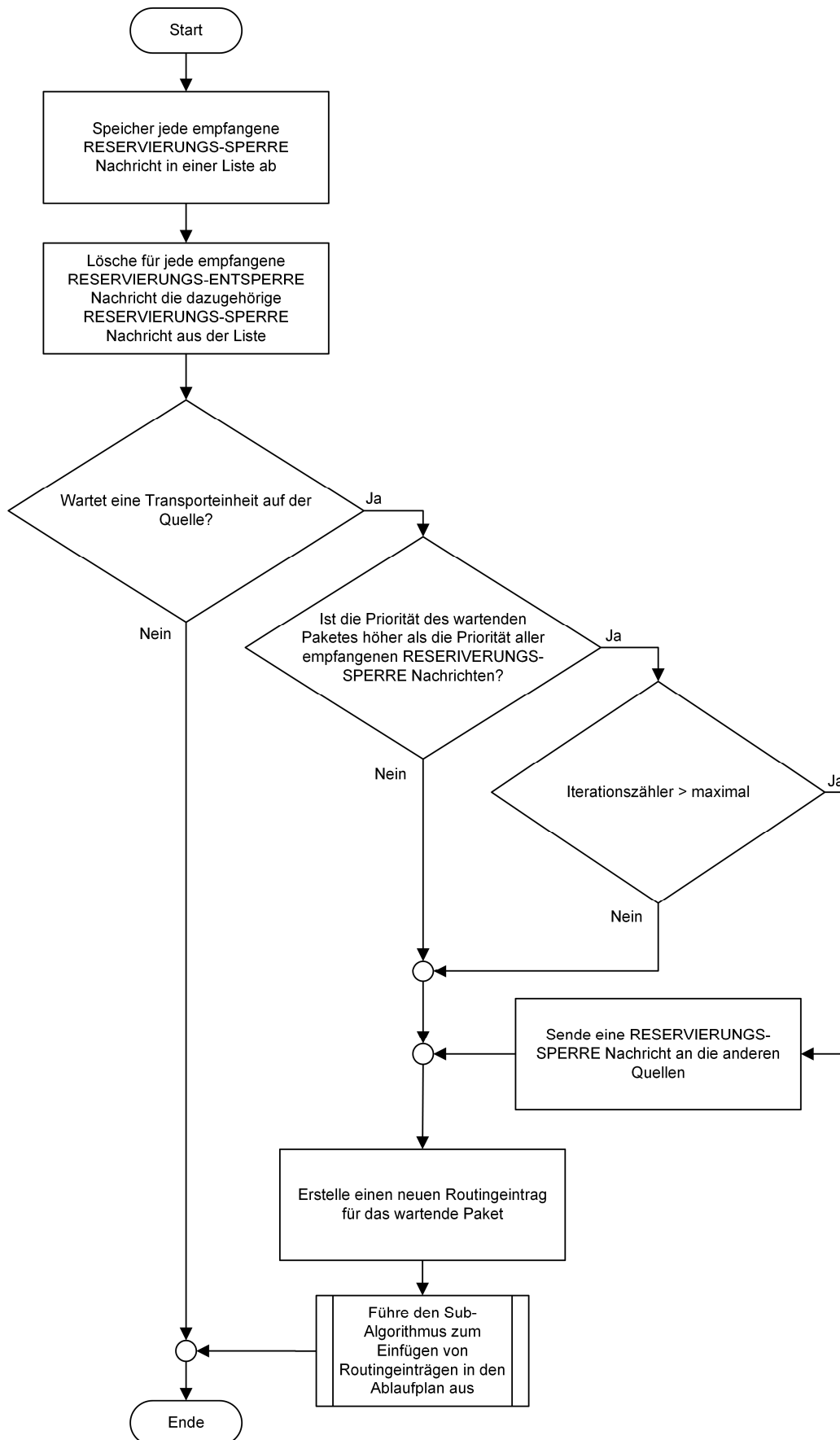


Abbildung A.1: Modifizierter Sub-Algorithmus zum Starten einer neuen Routinganfrage

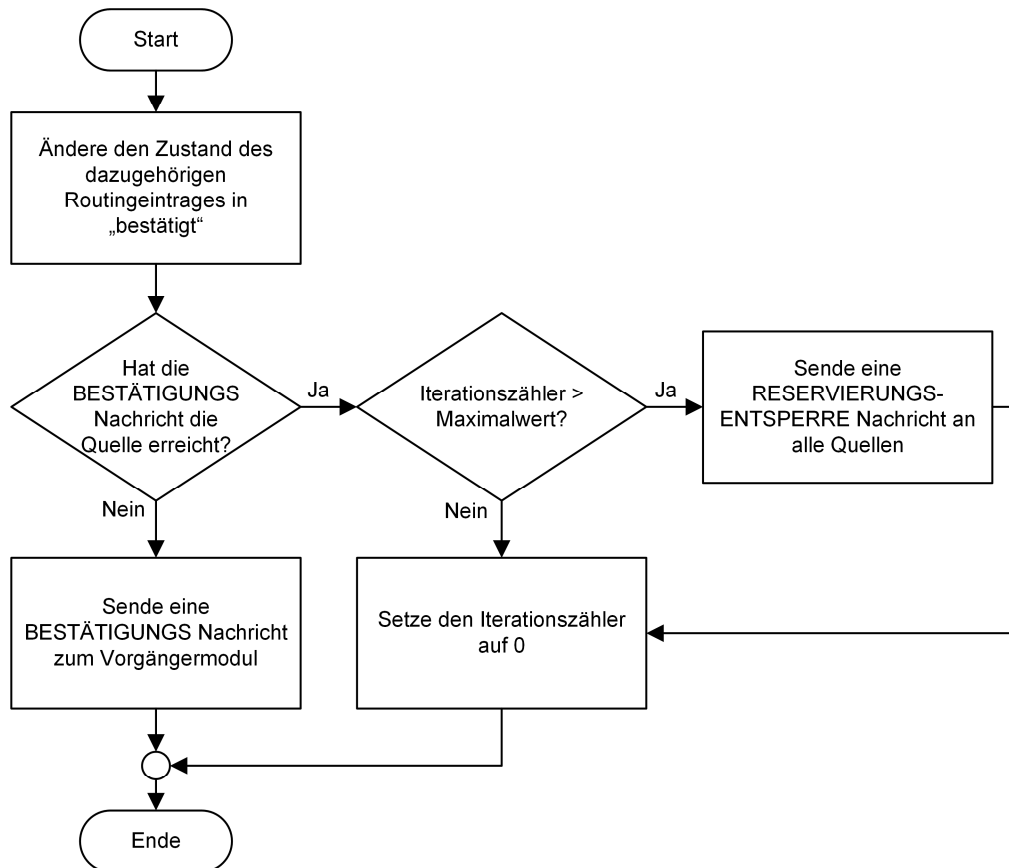


Abbildung A.2: Modifizierter Sub-Algorithmus zur Verarbeitung von **BESTÄTIGUNGS**-Nachrichten

A5 - Auswirkungen der Größe der Transporteinheiten auf die Leistungskenngrößen von Layouts mit kleinskaligen Fördermodulen

In diesem Kapitel werden die Auswirkungen der Größe der Transporteinheiten auf die Leistungskenngrößen von Layouts mit kleinskaligen Fördermodulen mit einer modulbasierten Ansteuerung (siehe Kapitel 5.3) qualitativ untersucht. Aus steuerungstechnischer Sicht sind die absoluten Größen der Module und der Transporteinheiten nicht relevant, sondern ihr Größenverhältnis. Durch die Definition von VENTZ in (siehe Kapitel 2.2) ist das maximale Größenverhältnis definiert: Wenn entlang der Kante einer Transporteinheit drei Module zum Transport benötigt werden, wird das Modul als kleinskalig bezeichnet. Wenn weniger als drei Module benötigt werden, handelt es sich um großskalige Module, deren Verhalten bereits in Simulationen (siehe Kapitel 7) quantitativ untersucht wurde. Nach der Definition von VENTZ ist das minimale Größenverhältnis jedoch nicht definiert: Module können beliebig klein werden in Bezug zu der Transporteinheit. Die in diesem Kapitel aufgestellten qualitativen Aussagen verstärken sich, umso extremer das Größenverhältnis zwischen Transporteinheit und Modul wird, also umso mehr Module für den Transport einer Transporteinheit benötigt werden.

Kleinskalige Fördermodule mit einer modulbasierten Ansteuerung bilden Modulnachbarschaft dynamisch. Wenn dynamische Nachbarschaften gebildet werden, können mehr Konflikte auftreten und somit wird die Verhinderung von Konflikte aufwendiger (siehe Kapitel 6.2). Dies führt dazu, dass der die Routenreservierung länger dauert als bei großskaligen Fördermodulen oder kleinskaligen Fördermodulen mit einer matrixbasierten Ansteuerung. Somit werden die Leistungskenngrößen negativ beeinflusst. Gleichzeitig werden durch die Verwendung von dynamischen Nachbarschaften die vorhandenen Module effektiver ausgenutzt, da immer nur genauso viele Module reserviert werden, wie benötigt werden. Die nicht benötigten Module können somit von anderen Transporteinheiten genutzt werden. Dieser Effekt tritt besonders dann auf, wenn ein breites Spektrum von unterschiedlich großen Transporteinheiten bewegt werden müssen, da in diesem Fall großskalige Fördermodule und kleinskalige Fördermodule mit einer matrixbasierten Ansteuerung besonders häufig zu große Modulnachbarschaften reservieren müssen. Die effektivere Nutzung der vorhandenen Module wirkt sich positiv auf die Leistungskenngrößen aus.

Da die Kommunikationsgeschwindigkeit um ein Vielfaches höher ist als die Transportgeschwindigkeit, wird die positive Beeinflussung der Leistungskenngrößen durch die effektivere Nutzung der Module die negative Beeinflussung der Leistungskenngrößen durch die längere Reservierungszeit übertreffen. Dieser Effekt wird mit abnehmender

Modulgröße weiter asymptotisch zunehmen, da die Modulnachbarschaft noch passender für die Größe der Transporteinheit gewählt werden kann.

Da es sich nur um eine asymptotische Zunahme handeln kann, wird mit dem folgenden Gedankenexperiment gezeigt: Bei einer Abnahme der Modulgröße müssen immer mehr Module innerhalb einer Nachbarschaft miteinander kommunizieren. Dies bedeutet, dass mit abnehmender Modulgröße die Kommunikationsgeschwindigkeit einen größeren Einfluss auf die Leistungskenngrößen hat und die Transportgeschwindigkeit somit einen kleineren Einfluss. Eine fortschreitende Verkleinerung der Module wird somit immer weniger vorteilhaft.

Weiterhin ist eine fortschreitende Verkleinerung der Module als ökonomisch fragwürdig anzusehen, da mehr Module und somit auch mehr Komponenten benötigt werden und somit ein Fördersystem, welches aus sehr kleinen Fördermodulen besteht, sehr teuer wird in der Anschaffung. In nachfolgenden Untersuchungen sollte geklärt werden, bis zu welchem Grad eine Verkleinerung der Module ökonomisch sinnvoll ist.

A6 - Systemverhalten von quadratischen Sortern

Tabelle A.12: Darstellung des Durchsatzes auf dem Layout „Sorter_4x4“ für unterschiedliche Werte von n_S

	$n_S = 1,0$	$n_S = 1,5$	$n_S = 2,0$	$n_S = 2,5$	$n_S = 3,0$
Durchsatz in TE/h	8470	8490	8482	8446	8483
Standardabweichung in TE/h	115	91	84	93	59

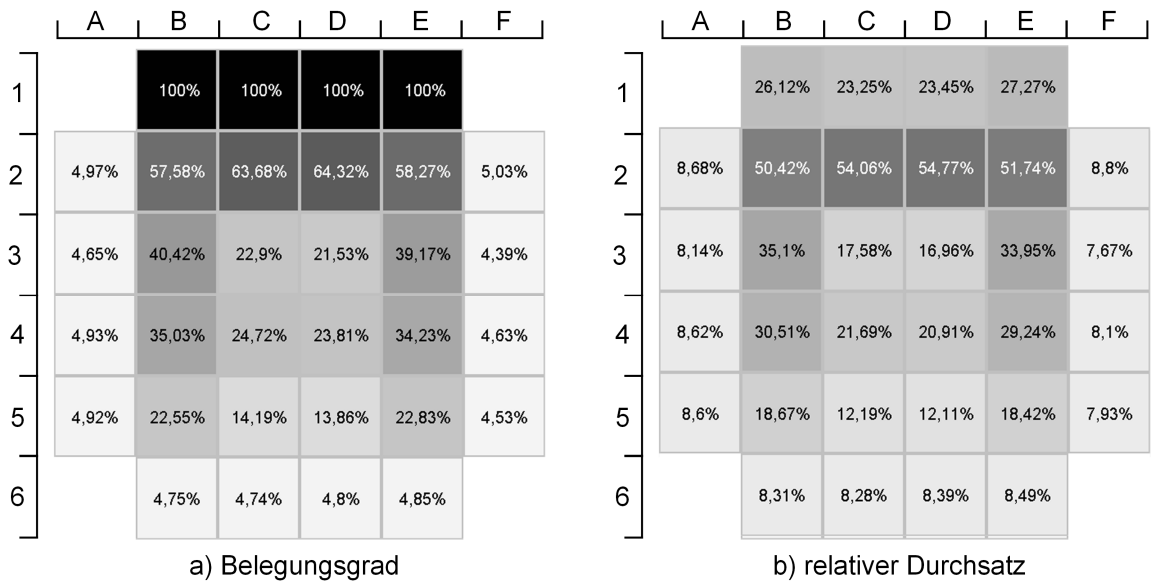


Abbildung A.3: Belegungsgrad und relativer Durchsatz für das Layout „Sorter_4x4“

Tabelle A.13: Darstellung des Durchsatzes auf dem Layout „Sorter_6x6“ für unterschiedliche Werte von n_S

	$n_S = 1,0$	$n_S = 1,5$	$n_S = 2,0$	$n_S = 2,5$	$n_S = 3,0$
Durchsatz in TE/h	12628	12574	12552	12501	12449
Standardabweichung in TE/h	83	144	115	128	109

	A	B	C	D	E	F	G	H
1		100%	100%	100%	100%	100%	100%	
2	4,65%	67,12%	67,35%	75,45%	77,12%	77,98%	66,81%	4,99%
3	4,63%	50,02%	63,1%	65,32%	65,9%	64,5%	59,48%	4,96%
4	4,83%	38,05%	47,92%	24,31%	22,74%	22,68%	43,63%	5,03%
5	4,69%	29,35%	46,54%	29,63%	30,01%	28,46%	42,18%	5,01%
6	4,33%	24,65%	33,47%	13,69%	13,41%	14,02%	29,17%	4,78%
7	5,17%	24,39%	29,44%	16,28%	16,03%	16,18%	22,96%	4,53%
8		4,84%	4,76%	4,84%	4,64%	4,92%	5,09%	

Abbildung A.4: Belegungsgrad für das Layout „Sorter_6x6“

	A	B	C	D	E	F	G	H
1		20,47%	15,52%	16,1%	16,28%	15,73%	16,01%	
2	5,37%	28,64%	30,25%	30,39%	31,2%	30,41%	26,47%	5,76%
3	5,34%	21,53%	36,19%	37,3%	37,48%	36,86%	34,56%	5,72%
4	5,57%	18,18%	27,23%	12,05%	11,95%	11,81%	25,23%	5,8%
5	5,41%	14,42%	26,75%	16,98%	17,13%	16,28%	24,47%	5,78%
6	5%	10,92%	19,26%	7,46%	7,39%	7,55%	16,83%	5,51%
7	5,97%	13,4%	17,04%	9,41%	9,24%	9,36%	12,72%	5,23%
8		5,58%	5,49%	5,58%	5,35%	5,67%	5,87%	

Abbildung A.5: Relativer Durchsatz für das Layout „Sorter_6x6“

Tabelle A.14: Darstellung des Durchsatzes auf dem Layout „Sorter_8x8“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	15364	15339	15288	15256	15203
Standardabweichung in TE/h	124	103	126	127	120

	A	B	C	D	E	F	G	H	I	J
1		100%	100%	100%	100%	100%	100%	100%	100%	
2	4,14%	64,65%	74,82%	74,79%	73,74%	70,72%	67,99%	67,19%	56,59%	4,57%
3	3,99%	53,17%	58,92%	60,85%	63,31%	59,14%	57,17%	50,64%	48,61%	4,53%
4	4,89%	41,64%	21,94%	27,51%	63,09%	61,99%	62,16%	59,1%	58,71%	4,7%
5	4,48%	36,8%	26,96%	27,84%	41,59%	18,75%	17,78%	17,91%	43,74%	4,49%
6	4,36%	42,43%	37,33%	38,09%	40,89%	23,62%	24,33%	19,59%	36,83%	4,31%
7	4,62%	33,21%	12,18%	12,02%	11,36%	15,4%	23,35%	22,13%	33,51%	4,17%
8	4,32%	26,57%	13,63%	13,73%	11,96%	15,01%	9,75%	9,48%	26,06%	4,69%
9	4,39%	17,97%	11,24%	12,88%	13,64%	14,84%	10,82%	10,7%	18,44%	4,23%
10		4,31%	4,59%	4,56%	4,38%	4,51%	4,19%	4,03%	4,44%	

Abbildung A.6: Belegungsgrad für das Layout „Sorter_8x8“

	A	B	C	D	E	F	G	H	I	J
1		14,18%	12,82%	13,01%	12,9%	12,67%	11,86%	11,73%	10,97%	
2	3,92%	20,7%	24,84%	25,65%	24,33%	24,31%	23,74%	21,37%	20,39%	4,32%
3	3,77%	25,09%	27,01%	27,32%	27,96%	22,23%	22,04%	18,38%	19,93%	4,29%
4	4,44%	19,69%	8,6%	10,19%	29,88%	29,3%	29,21%	27,79%	28,01%	4,45%
5	4,24%	16,81%	11,19%	10,58%	19,25%	8,26%	7,8%	7,3%	20,78%	4,25%
6	4,13%	20,27%	17,68%	18,02%	19,33%	10,92%	11,48%	8,78%	17,44%	4,07%
7	4,37%	15,79%	5,42%	5,47%	5,24%	7,28%	11,04%	10,48%	15,93%	3,94%
8	4,09%	12,52%	6,45%	6,45%	5,6%	7,03%	4,44%	4,42%	12,36%	4,44%
9	4,15%	8,37%	5,24%	6,05%	6,46%	7,03%	5,14%	5,05%	8,64%	4%
10		4,07%	4,34%	4,31%	4,14%	4,27%	3,97%	3,82%	4,21%	

Abbildung A.7: Relativer Durchsatz für das Layout „Sorter_8x8“

Tabelle A.15: Darstellung des Durchsatzes auf dem Layout „Sorter_12x12“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	19174	19318	19358	19380	19354
Standardabweichung in TE/h	185	196	190	248	197

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	
2	3,28%	55,09%	62,97%	64,51%	50,96%	50,85%	50,32%	64,17%	65,03%	55,22%	62,42%	62,56%	55,55%	3,24%
3	3,15%	44,44%	47,3%	44,47%	36,54%	21,92%	46,44%	51,98%	57,91%	56,26%	49,62%	46,12%	41,24%	3,43%
4	3,2%	39,88%	46,29%	48,17%	25,12%	22,53%	28,54%	21,23%	42,28%	49,84%	53,3%	54,65%	41,61%	3,4%
5	3,38%	30,31%	35,87%	36,33%	29,52%	30,67%	24,6%	18,76%	36,43%	15,11%	19,02%	45,39%	45,6%	3,28%
6	3,37%	24,98%	36%	34,02%	17,38%	18,29%	15,64%	23,52%	36,42%	15,59%	18,77%	11,39%	42,43%	3,45%
7	3,14%	20,25%	37,98%	34,62%	25,27%	19,58%	19,81%	33,32%	30,56%	10,83%	18,07%	15,63%	39,75%	3,1%
8	3,43%	18,75%	32,33%	14,05%	26,68%	30,62%	39,57%	41,02%	13,2%	12,44%	13,13%	10,08%	33,85%	3,23%
9	3,34%	16,55%	37,35%	21,97%	24,08%	26,12%	28,57%	27,15%	26,61%	26,13%	12,13%	9,51%	29,8%	3,15%
10	3,35%	14,71%	33,1%	8,37%	6,67%	8,14%	7,34%	7,89%	7,88%	19,65%	7,95%	9,36%	25,89%	3,39%
11	3,43%	13,15%	28,96%	8,65%	6,89%	7,99%	7,48%	8,73%	8,14%	18,22%	8,04%	8,86%	20,9%	3,26%
12	3,61%	20,27%	24,06%	8,39%	6,99%	8,55%	8,05%	8,75%	7,6%	16,19%	8,37%	9,64%	15,81%	3,37%
13	3,29%	14,03%	8,82%	9,28%	8,38%	9,56%	9,58%	9,72%	8,96%	15,34%	13,25%	12,04%	13,36%	3,39%
14		3,49%	3,42%	3,26%	3,11%	3,26%	3,33%	3,3%	3,31%	2,95%	3,22%	3,47%	3,15%	

Abbildung A.8: Belegungsgrad für das Layout „Sorter_12x12“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		11,4%	10,1%	8,87%	6,5%	6,43%	6,38%	8,78%	8,53%	7,35%	8,03%	8,62%	9,17%	
2	2,76%	16,04%	16,8%	15,36%	20,44%	20,26%	19,4%	16,45%	15,95%	17,52%	15,89%	15,89%	14,41%	2,73%
3	2,65%	14,31%	14,08%	17,94%	14,19%	6,89%	19,15%	21,31%	24,15%	23,21%	13,15%	14,37%	13,93%	2,88%
4	2,69%	12,51%	19,22%	19,89%	8,3%	7,54%	11,32%	7,6%	17,64%	20,9%	22,31%	23,1%	11,18%	2,86%
5	2,84%	10,66%	14,4%	14,8%	11,98%	12,79%	9,54%	5,99%	15,1%	5,25%	7,39%	18,97%	19,34%	2,76%
6	2,83%	9,66%	14,46%	13,86%	5,37%	7,2%	5,44%	6,61%	15,26%	6,41%	7,82%	4,27%	17,91%	2,9%
7	2,64%	7,01%	15,92%	14,14%	10,29%	7,3%	6,04%	13,81%	12,65%	4,08%	7,6%	6,5%	16,81%	2,61%
8	2,88%	7,3%	13,45%	4,36%	10,06%	10,94%	16,56%	17,21%	4,38%	4,6%	5,3%	4,02%	14,28%	2,72%
9	2,81%	6,24%	15,78%	9,24%	10,14%	10,99%	12,03%	11,44%	11,2%	11,02%	4,89%	3,79%	12,58%	2,64%
10	2,81%	5,6%	13,96%	3,42%	2,8%	3,33%	3,06%	3,3%	3,31%	8,26%	3,23%	3,84%	10,94%	2,85%
11	2,88%	4,79%	12,21%	3,55%	2,91%	3,36%	3,14%	3,67%	3,42%	7,65%	3,26%	3,72%	8,81%	2,74%
12	3,04%	8,55%	10,12%	3,51%	2,94%	3,54%	3,3%	3,61%	3,15%	6,81%	3,35%	3,84%	6,63%	2,83%
13	2,77%	5,79%	3,65%	3,91%	3,53%	4,02%	4,03%	4,09%	3,77%	6,46%	5,59%	5,07%	5,61%	2,85%
14		2,93%	2,88%	2,74%	2,62%	2,74%	2,8%	2,77%	2,79%	2,48%	2,71%	2,91%	2,65%	

Abbildung A.9: Relativer Durchsatz für das Layout „Sorter_12x12“

Tabelle A.16: Darstellung des Durchsatzes auf dem Layout „Sorter_14x14“ für unterschiedliche Werte von n_S

	$n_S = 1,0$	$n_S = 1,5$	$n_S = 2,0$	$n_S = 2,5$	$n_S = 3,0$
Durchsatz in TE/h	13816	13583	13538	13394	13277
Standardabweichung in TE/h	147	303	219	193	257

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	
2	3,09%	53,93%	49,69%	49,95%	58,98%	50,23%	56,51%	52,89%	58,66%	50,5%	56,4%	53,45%	58,68%	50,49%	52,95%	2,99%
3	3,15%	32,55%	35,89%	41,89%	45,7%	48,76%	35,32%	39,43%	45,23%	47,8%	38,48%	40%	42,71%	42,72%	34,93%	3,18%
4	3,1%	26,72%	41,67%	36,09%	49,6%	52,51%	34,31%	31,2%	45,41%	50,77%	37,61%	36,69%	45,81%	45,74%	28,44%	3,22%
5	3%	23,61%	35,19%	15,85%	14,38%	38,04%	42,56%	35,6%	40,73%	48,44%	20,45%	14,54%	14,45%	33,25%	35,74%	2,99%
6	3,13%	22,5%	35,43%	16,31%	14,75%	16,87%	45,81%	44,83%	35,84%	47,82%	23,84%	16,73%	13,82%	13,53%	33,01%	3,1%
7	2,93%	21,62%	32,58%	19,49%	18,61%	19,65%	27,54%	36,16%	43,76%	48,88%	19,05%	13,75%	15,73%	11,63%	29,85%	3,19%
8	3,12%	23,59%	43,61%	29,89%	35,26%	36,85%	39,55%	33,79%	52,67%	55,72%	19,23%	11,66%	15,46%	11,2%	26,99%	3,06%
9	2,88%	22,99%	41,76%	9,99%	8,1%	9,67%	8,43%	11,77%	11,16%	36,6%	36,14%	15,26%	16,23%	13,85%	24,45%	3,47%
10	3,15%	37,51%	39,34%	10,67%	8,01%	11,28%	9,98%	12,47%	10,84%	9,83%	34,64%	30,66%	15,21%	13,6%	19,49%	2,9%
11	2,95%	31,61%	7,26%	8,95%	7,56%	10,26%	6,66%	9,58%	11,75%	7,58%	10,31%	26,57%	27,84%	16,25%	18,13%	3,23%
12	2,86%	27,93%	9,34%	10,03%	9,6%	10,97%	6,95%	7,24%	11,82%	7,65%	11,79%	10,24%	26,73%	26,34%	17,11%	3,14%
13	3,28%	23%	7,18%	6,76%	6,44%	8,52%	8,34%	7,15%	11,51%	7,56%	10,74%	6,64%	7,16%	20,87%	22,4%	3,17%
14	3,08%	17,51%	7,22%	6,81%	6,59%	6,81%	8,24%	8,62%	11,05%	7,85%	10,63%	7,29%	7,27%	7,33%	16,82%	2,97%
15	3,01%	12,38%	7,75%	8,11%	8,85%	9,49%	9,85%	11,43%	15,46%	14,56%	12,72%	9,15%	9,28%	8,04%	12,35%	3,02%
16	2,84%	2,99%	3,03%	3,03%	3,08%	3,15%	3,12%	3,22%	3,06%	2,88%	3,03%	3,28%	2,94%	3,07%		

Abbildung A.10: Belegungsgrad für das Layout „Sorter_14x14“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		8,47%	6,37%	6,19%	7,58%	6,73%	6,91%	7,09%	7,83%	6,59%	6,6%	6,97%	7,81%	7,18%	7,9%	
2	2,39%	11,7%	15,01%	14,86%	12,88%	16,09%	14,82%	15,3%	13,19%	16,63%	14,79%	14,94%	13,17%	14,21%	11,59%	2,31%
3	2,44%	9,95%	11,79%	11,62%	15,6%	15,93%	8,9%	11,72%	15,61%	16,57%	8,69%	10,97%	14,59%	13,54%	9,61%	2,46%
4	2,4%	7,91%	16,12%	13,69%	19,06%	20,05%	8,41%	8,22%	14,19%	19,01%	13,35%	13,7%	17,29%	17,39%	8,18%	2,5%
5	2,32%	7,12%	13,57%	5%	4,59%	14,21%	15,82%	11,05%	13,49%	18,01%	6,11%	5,2%	4,66%	12,46%	13,97%	2,32%
6	2,43%	6,57%	13,46%	5,65%	5,25%	5,2%	17,52%	16,89%	9,68%	17,96%	7,02%	6,2%	5,08%	4,66%	12,92%	2,41%
7	2,27%	7,16%	12,4%	5%	6,34%	5,44%	9,43%	12,05%	15,49%	16,92%	5,62%	4,83%	5,9%	3,77%	11,69%	2,47%
8	2,42%	6,67%	17,01%	11,55%	13,68%	14,31%	15,33%	13,11%	20,44%	21,68%	4,99%	4,14%	5,78%	3,9%	10,5%	2,37%
9	2,23%	5,78%	16,14%	3,74%	3,06%	3,6%	3,23%	4,52%	4,13%	14,18%	13,98%	4,53%	5,98%	4,72%	9,49%	2,68%
10	2,44%	14,65%	15,25%	4,09%	3,09%	4,38%	3,86%	4,8%	4,09%	3,42%	13,4%	11,86%	5,01%	5,13%	7,42%	2,24%
11	2,29%	12,33%	2,7%	3,41%	2,89%	3,98%	2,57%	3,67%	4,56%	2,93%	3,83%	10,27%	10,81%	5,1%	6,94%	2,5%
12	2,22%	10,89%	3,59%	3,84%	3,72%	4,26%	2,66%	2,76%	4,58%	2,95%	4,57%	3,7%	10,36%	10,21%	5,92%	2,43%
13	2,54%	8,95%	2,72%	2,61%	2,47%	3,25%	3,21%	2,76%	4,41%	2,91%	4,15%	2,57%	2,73%	8,08%	8,71%	2,45%
14	2,38%	6,8%	2,79%	2,63%	2,53%	2,59%	3,19%	3,23%	4,2%	2,88%	4,11%	2,81%	2,78%	2,71%	6,54%	2,3%
15	2,33%	4,76%	3%	3,14%	3,43%	3,69%	3,82%	4,44%	6,01%	5,65%	4,94%	3,55%	3,6%	3,12%	4,79%	2,34%
16	2,2%	2,32%	2,35%	2,35%	2,38%	2,44%	2,44%	2,42%	2,5%	2,37%	2,23%	2,35%	2,54%	2,28%	2,38%	

Abbildung A.11: Relativer Durchsatz für das Layout „Sorter_14x14“

Tabelle A.17: Darstellung des Durchsatzes auf dem Layout „Sorter_16x16“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	19495	19617	19782	19880	19842
Standardabweichung in TE/h	166	211	235	222	193

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
2	2,86%	45,99%	50,15%	50,61%	49,02%	48,4%	52,26%	52,94%	50,69%	50,51%	50,77%	49,28%	48,88%	56%	52,86%	51,24%	52,48%	2,94%
3	3,03%	38,08%	31,47%	34,19%	42,6%	50,31%	33,01%	33,48%	44,81%	51,33%	28,45%	21,35%	35,09%	42,96%	45,02%	42,95%	41,96%	2,87%
4	2,88%	45,02%	32,95%	36,44%	20,07%	37,78%	39,9%	36,15%	20,18%	33,08%	28,06%	22,25%	39,97%	37,8%	32,63%	21%	38,45%	2,89%
5	3,04%	42,28%	13,91%	11,51%	11,55%	14,21%	44,83%	43,99%	19,52%	32,72%	33,97%	25,19%	29,22%	29,98%	34,74%	24,93%	35,8%	2,88%
6	2,66%	38,2%	13,69%	13,34%	15,79%	16,26%	26,42%	32,1%	21,21%	22,04%	35,68%	29,74%	32,72%	29,01%	37,82%	29,21%	28,15%	2,67%
7	3,01%	35,98%	15,64%	12,18%	18,54%	32,6%	35,35%	40,98%	40,68%	40,16%	47,79%	43,41%	45,36%	45,69%	17,9%	25,9%	25,99%	2,92%
8	2,85%	31,44%	18,68%	30,08%	32,02%	32,21%	11,4%	13%	11,22%	12,23%	10,01%	12,47%	25,59%	44,13%	24,96%	25,48%	23,86%	2,92%
9	2,74%	30,82%	21,88%	29,68%	8,4%	10,9%	10,33%	8,46%	7,15%	11,54%	15,29%	25,46%	25%	33,3%	31,25%	25,91%	21,61%	2,74%
10	2,85%	38,19%	29,49%	30,78%	7,88%	5,89%	9,06%	9%	7,56%	21,27%	26,3%	26,42%	7,19%	8,13%	28,7%	38,94%	42%	3,1%
11	2,94%	34,8%	9,88%	11,23%	8,53%	7,13%	10,59%	11,69%	13,14%	21,61%	6,93%	11,22%	7,86%	8,24%	6,67%	8,92%	36,61%	2,76%
12	2,71%	30,06%	9,09%	10,76%	8,02%	7,03%	7,12%	6,69%	21,83%	22,57%	7,24%	11,03%	8,88%	6,29%	5,33%	8,26%	32,61%	2,58%
13	2,83%	25,8%	8,72%	10,18%	7,64%	6,67%	7,77%	17,39%	18,12%	9,26%	6,03%	6,07%	8,52%	6,43%	5,89%	7,6%	28,2%	2,64%
14	2,6%	20,82%	8,24%	9,28%	7,67%	5,94%	15,7%	16,68%	5,76%	8,71%	5,98%	6,06%	8,3%	6,42%	5,63%	7,49%	24,24%	2,85%
15	2,92%	19,24%	9,3%	10,18%	11,1%	9,91%	14,73%	7,19%	6,26%	8,8%	6,21%	6,31%	7,97%	8,17%	5,71%	6,73%	19,51%	2,96%
16	2,84%	14,86%	6,52%	5,97%	7,53%	14,85%	15,66%	6,6%	6,14%	6,33%	7,97%	7,99%	5,75%	7,61%	7,32%	7,99%	14,64%	2,47%
17	2,76%	11,52%	8,85%	9,28%	11,42%	11,2%	6,01%	8,51%	8,26%	7,68%	7,17%	8,61%	8,06%	7,36%	6,75%	8,09%	11,73%	3,14%
18		2,83%	2,92%	2,42%	2,93%	2,71%	2,75%	2,83%	2,81%	2,63%	2,69%	2,79%	2,76%	2,55%	2,58%	2,95%	2,67%	

Abbildung A.12: Belegungsgrad für das Layout „Sorter_16x16“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		0,82%	5,04%	5,11%	5,17%	5,41%	5,58%	5,37%	5,71%	5,78%	5,9%	6,11%	6,59%	8,06%	7,48%	7,31%	8,17%	
2	2,12%	13,67%	12,39%	12,98%	13,31%	14,75%	14,02%	14,71%	14,41%	15,59%	15,2%	15,99%	15,68%	13,74%	13,73%	13,77%	12,15%	2,19%
3	2,25%	13,3%	6,73%	12,09%	15,4%	18,46%	6,53%	11,35%	16,03%	18,83%	6,93%	5,9%	11,7%	11,92%	14,88%	14,14%	15,39%	2,13%
4	2,13%	17,1%	11,85%	13,41%	6,82%	13,11%	14,19%	11,7%	5,28%	11,76%	7,15%	6,57%	14,1%	13%	9,9%	5,39%	14,08%	2,14%
5	2,26%	15,93%	4,14%	3,99%	4,02%	3,81%	16,6%	16,26%	5,27%	11,46%	12,36%	6,86%	9,03%	8,69%	12,36%	7,3%	12,82%	2,14%
6	1,99%	14,39%	4,34%	4,56%	5,55%	4,34%	9,38%	11,02%	5,85%	5,97%	12,34%	8,38%	9,76%	9,57%	12,76%	10,4%	9,93%	1,98%
7	2,23%	13,25%	5,27%	3,83%	5,29%	12,04%	12,97%	15,2%	15,1%	14,83%	17,72%	16,05%	16,61%	16,94%	4,28%	8,9%	9,05%	2,17%
8	2,12%	11,68%	5,39%	11,08%	11,89%	11,97%	4,14%	4,76%	4,14%	4,45%	3,59%	3,3%	9,47%	16,34%	4,1%	8,91%	8,03%	2,16%
9	2,03%	11,25%	6,39%	10,9%	3,1%	3,95%	3,83%	3,14%	2,62%	3,95%	5,12%	9,35%	9,22%	12,29%	11,59%	8,5%	7,12%	2,04%
10	2,11%	14,29%	10,91%	11,41%	2,77%	2,16%	3,29%	3,18%	2,71%	7,9%	9,76%	9,82%	2,59%	2,84%	10,64%	14,44%	15,71%	2,3%
11	2,19%	13,01%	3,41%	4,14%	3,11%	2,03%	3,78%	4,29%	4,22%	7,85%	2,48%	4,14%	2,84%	3,03%	2,48%	3,05%	13,67%	2,06%
12	2,02%	11,23%	3,15%	3,9%	2,97%	2,58%	2,63%	2,34%	8,1%	8,37%	2,67%	4,02%	3,3%	2,34%	1,95%	3,05%	12,17%	1,91%
13	2,1%	9,82%	3,15%	3,72%	2,8%	2,46%	2,74%	6,4%	6,73%	3,43%	2,22%	2,18%	3,16%	2,39%	2,1%	2,74%	10,52%	1,96%
14	1,93%	7,76%	2,99%	3,43%	2,72%	2,11%	5,78%	6,19%	2,13%	3,23%	2,2%	2,23%	3,08%	2,36%	2,09%	2,74%	9,04%	2,12%
15	2,16%	7,16%	3,45%	3,7%	4,07%	3,48%	5,4%	2,6%	2,32%	3,21%	2,21%	2,35%	2,96%	3,03%	2,12%	2,48%	7,27%	2,2%
16	2,11%	5,43%	2,41%	2,14%	2,75%	5,52%	5,01%	3,24%	2,15%	3,04%	2,95%	2,96%	2,12%	2,88%	2,72%	2,94%	5,45%	1,84%
17	2,05%	4,24%	3,29%	3,45%	4,21%	4,16%	2,23%	3,16%	3,07%	2,86%	2,66%	3,2%	2,99%	2,73%	2,51%	3,01%	4,32%	2,33%
18		2,1%	2,16%	1,79%	2,18%	2,01%	2,04%	2,1%	2,08%	1,95%	2%	2,07%	2,05%	1,89%	1,92%	2,19%	1,98%	

Abbildung A.13: Relativer Durchsatz für das Layout „Sorter_16x16“

Tabelle A.18: Darstellung des Durchsatzes auf dem Layout „Sorter_23x11x2“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	12529	12640	12605	12633	12557
Standardabweichung in TE/h	83	67	73	66	78

A7 - Systemverhalten von rechteckigen Sortern

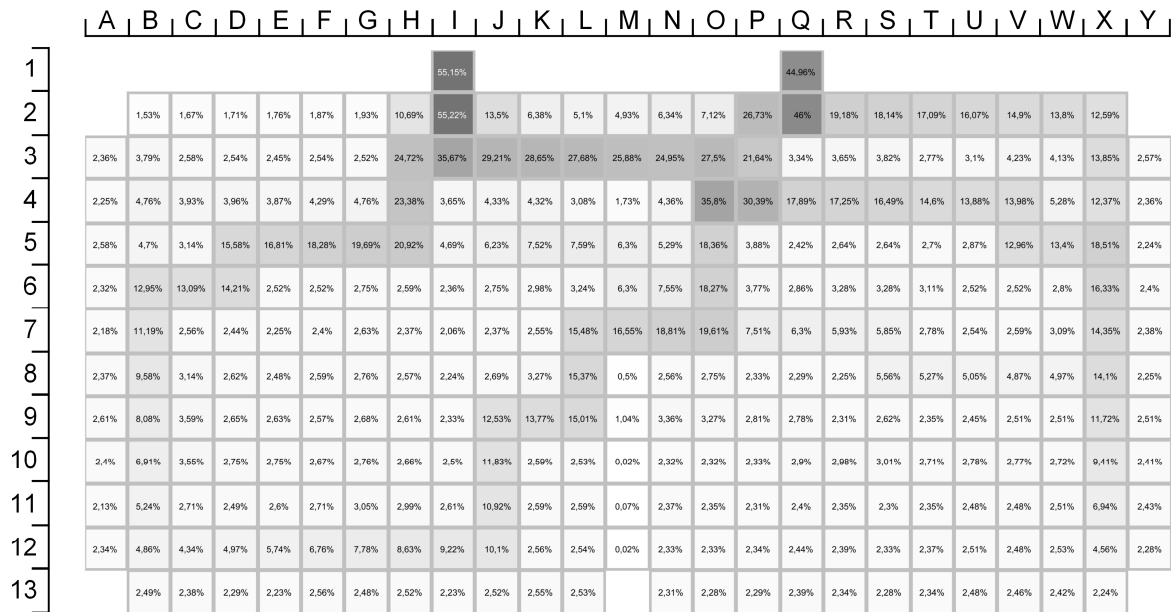


Abbildung A.14: Relativer Durchsatz für das Layout „Sorter_23x11x2“

Tabelle A.19: Darstellung des Durchsatzes auf dem Layout „Sorter_23x11x6“ für unterschiedliche Werte von n_S

	$n_S = 1,0$	$n_S = 1,5$	$n_S = 2,0$	$n_S = 2,5$	$n_S = 3,0$
Durchsatz in TE/h	17836	18138	18213	18262	18417
Standardabweichung in TE/h	131	148	191	171	121

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1			100%				100%				100%				100%				100%				100%			
2		24,49%	66,67%	43,12%	23,76%	22,63%	64,93%	34,78%	33,17%	36,23%	74,76%	42,03%	38,98%	37,97%	47,17%	35,53%	34,46%	34,77%	44,84%	29,32%	29,22%	30,81%	64,53%	30,1%		
3	2,97%	22,11%	21,26%	51,3%	48,88%	48,08%	57,69%	51,63%	51,67%	51,53%	58,96%	36,86%	14,55%	14,81%	14,63%	14,27%	14,08%	16,85%	19,35%	47,94%	46,33%	44,17%	45,74%	33,29%	2,94%	
4	3,13%	21,57%	23,74%	32,52%	13,81%	13,77%	13,83%	13,34%	17,2%	26,93%	59%	51,77%	38,44%	39,95%	38,69%	37,83%	36,94%	37,43%	33,33%	36,4%	15,99%	15,67%	17,66%	30,85%	2,86%	
5	2,73%	33,67%	31,78%	34,48%	22,26%	23,73%	24,66%	25,24%	27,94%	29,14%	37,32%	12,09%	7,96%	10,27%	11,79%	21,68%	21,37%	39,81%	41,98%	41,45%	28,74%	28,26%	27,29%	38,74%	3,15%	
6	3,05%	29,46%	9,77%	8,7%	9,14%	9,55%	10%	12,37%	27,84%	31,55%	39,39%	18,95%	21,54%	24,83%	25,44%	25,66%	8,24%	8,19%	10,65%	11,8%	9,73%	8,62%	8,92%	34,44%	3,23%	
7	3,25%	25,7%	10,8%	7,83%	8,54%	6,88%	9,32%	30,51%	39,7%	44,07%	47,69%	30,87%	25,06%	19,35%	18,52%	14,38%	9,18%	10,32%	12,71%	12,32%	8,12%	8,1%	10,44%	30,81%	3,35%	
8	2,78%	23,57%	11,11%	9,36%	9,19%	8,65%	12,72%	28,61%	9,38%	9,57%	8,33%	8,49%	2,92%	6,97%	21,92%	25,48%	24,46%	24,33%	24,54%	22,49%	18,24%	17,34%	10,87%	26,42%	3,19%	
9	2,83%	20,09%	7,37%	7,21%	6,57%	22,68%	24,8%	28,33%	11,88%	13,37%	13,51%	13,25%	8,72%	8,42%	11,38%	6,89%	7,08%	7,39%	7,7%	7,59%	7,19%	15,46%	15,64%	26,72%	2,8%	
10	3,28%	17,72%	8,49%	8,37%	8,05%	20,9%	6,49%	6,87%	7,23%	6,94%	11,1%	11,49%	11,77%	16,68%	16,11%	12,43%	11,94%	11,54%	11,02%	9,92%	6,5%	6,27%	7,05%	21,17%	2,88%	
11	3,01%	13,16%	6,74%	7,65%	7,9%	21,01%	9,17%	10,08%	10,75%	11,09%	11,3%	6,39%	0,56%	6,71%	6,18%	6,65%	6,3%	6,01%	6,26%	9%	6,82%	6,4%	7,12%	16,51%	3,01%	
12	2,94%	12,13%	11,61%	14,91%	16,63%	19,19%	6,78%	7,07%	6,81%	6,88%	7,62%	7,68%	1,83%	7,79%	6,73%	7,41%	6,9%	6,62%	6,97%	9,32%	8,89%	8,01%	7,76%	11,79%	2,76%	
13	3,03%	2,99%	3,17%	2,78%	3,02%	2,94%	3,15%	2,94%	2,99%	3,03%	3,03%			2,86%	2,6%	3,08%	2,9%	2,82%	2,92%	2,92%	3,08%	2,96%	3,1%	3,02%		

Abbildung A.15: Belegungsgrad für das Layout „Sorter_23x11x6“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1			22,14%				21,79%				15,39%				9,52%				11,02%				20,3%		
2		6,19%	22,88%	16,29%	8,19%	8,1%	24,17%	11,22%	9,75%	9,84%	19,82%	15,8%	15,11%	14,87%	18,53%	13,53%	13,32%	13,06%	16,78%	11,3%	6,5%	8,39%	20,63%	3,6%	
3	2,36%	8,45%	5,39%	19,86%	19,1%	18,82%	22,62%	20,3%	19,86%	18,8%	20,53%	12,77%	4,7%	5,34%	5,17%	5,37%	5,1%	5,95%	5,99%	18,93%	18,04%	17,23%	17,9%	13,39%	2,34%
4	2,5%	8%	6,59%	12,57%	5,1%	5,19%	5,1%	4,54%	5,55%	7,65%	23,63%	20,67%	15,25%	15,8%	15,34%	14,92%	14,52%	14,71%	7,94%	11,74%	4,02%	5,51%	6,04%	12,34%	2,28%
5	2,17%	13,59%	12,56%	13,66%	8,58%	9,19%	9,44%	9,51%	10,23%	7,78%	14,75%	4,26%	2,7%	3,76%	4,02%	8,39%	8,42%	15,91%	16,73%	16,54%	11,42%	11,21%	10,85%	15,67%	2,51%
6	2,43%	11,83%	3,74%	3,21%	3,53%	3,68%	3,84%	4,32%	9,12%	8,64%	15,21%	5,61%	8,35%	9,75%	9,88%	10,18%	3,13%	3,12%	4,14%	4,63%	3,62%	3,4%	3,52%	13,9%	2,57%
7	2,59%	10,26%	4,22%	3,01%	3,38%	2,62%	2,81%	12,21%	15,87%	17,63%	19,06%	12,31%	10,32%	7,63%	7,33%	5,29%	3,37%	3,84%	4,72%	4,75%	3,06%	3,14%	3,5%	12,38%	2,67%
8	2,22%	9,49%	4,4%	3,66%	3,6%	3,26%	3,15%	11,41%	3,53%	3,71%	3,21%	3,34%	1,13%	2,7%	8,76%	10,15%	9,77%	9,7%	9,74%	8,95%	7,29%	6,9%	3,84%	10,53%	2,55%
9	2,25%	8,07%	2,88%	2,84%	2,5%	9,08%	9,88%	11,29%	4,7%	5,34%	5,34%	5,24%	3,43%	3,19%	4,39%	2,63%	2,76%	2,91%	2,97%	2,92%	2,88%	6,19%	6,23%	10,75%	2,23%
10	2,62%	7,12%	3,38%	3,3%	2,03%	8,36%	2,61%	2,66%	2,83%	2,72%	4,43%	4,69%	4,66%	6,68%	6,38%	4,06%	4,77%	4,69%	4,4%	3,07%	2,61%	2,61%	2,73%	8,6%	2,20%
11	2,4%	5,28%	2,65%	3,01%	2,68%	8,4%	3,65%	4,03%	4,27%	4,4%	4,51%	2,54%	0,19%	2,68%	2,37%	2,66%	2,48%	2,36%	2,48%	3,58%	2,6%	2,55%	2,81%	6,6%	2,4%
12	2,34%	4,86%	4,65%	5,97%	6,66%	7,68%	2,71%	2,83%	2,72%	2,76%	3,04%	3,07%	0,73%	3,11%	2,69%	2,97%	2,76%	2,65%	2,76%	3,72%	3,56%	3,22%	3,11%	4,65%	2,2%
13	2,41%	2,39%	2,53%	2,21%	2,41%	2,34%	2,51%	2,35%	2,38%	2,41%	2,42%			2,28%	2,08%	2,45%	2,31%	2,25%	2,33%	2,33%	2,46%	2,36%	2,47%	2,41%	

Abbildung A.16: Relativer Durchsatz für das Layout „Sorter_23x11x6“

Tabelle A.20: Darstellung des Durchsatzes auf dem Layout „Sorter_23x11x10“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	18974	19207	19337	19450	19469
Standardabweichung in TE/h	130	180	177	179	224

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1				100%		100%		100%		100%		100%		100%		100%		100%		100%		100%		100%		100%
2		15,1%	12,97%	62,94%	42,45%	73,52%	49,62%	45,3%	38,17%	44,8%	38,33%	44,66%	36,69%	42,59%	35,36%	48,83%	57,21%	73%	56,18%	74,23%	48,29%	64,46%	25,51%	28,64%		
3	3,23%	27,49%	27,78%	46,9%	46,66%	57,51%	52,79%	36,9%	18,63%	16,26%	16,86%	14,94%	15,12%	17,74%	25,4%	63,46%	56,77%	59,71%	54,25%	56,53%	48,45%	49,79%	39,83%	26,61%	3,08%	
4	3,17%	25,95%	11,55%	13,97%	20,3%	31,5%	51,53%	52,03%	39,39%	38,71%	37,41%	36,31%	35,44%	35,04%	34,46%	44,99%	23,96%	21,62%	19,98%	17,42%	17,93%	14,18%	37,95%	37,38%	3,07%	
5	3,08%	25,89%	13,26%	16,43%	20,76%	29,19%	33,63%	13,04%	13,34%	15,6%	18,88%	33,09%	33,1%	46,9%	52,21%	55,41%	36,66%	35,11%	33%	29,27%	26,29%	12,75%	10,79%	33,08%	3,01%	
6	2,96%	22,23%	8,58%	9,76%	19,9%	24,8%	34,98%	22,66%	31,36%	34,02%	35,53%	37,02%	9,05%	9,25%	12,01%	16,99%	14,66%	10,91%	9,85%	9,62%	20,43%	11,9%	15,53%	30,04%	3,05%	
7	3,02%	19,77%	10,6%	28,51%	34,05%	39,33%	45,35%	32,83%	32,59%	16,18%	16,09%	16,06%	10,2%	12,68%	15,46%	17,86%	9,48%	7,8%	7,3%	10,75%	20,05%	19,91%	21,18%	34,65%	3,15%	
8	3,07%	17,59%	13,41%	27,23%	7,8%	7,91%	7,87%	8,95%	7,22%	7,61%	21,75%	27,22%	27,2%	28,31%	29,82%	27,32%	20,19%	18,54%	9,67%	10,12%	7,54%	8,93%	10,19%	28,6%	2,85%	
9	3,24%	25,89%	23,67%	25,26%	9,18%	10,6%	11,3%	12,77%	12,24%	9,57%	14,34%	6,97%	7,69%	7,73%	8,49%	9,43%	9,05%	18,77%	17,61%	16,46%	14,34%	13,97%	14,65%	28,29%	3,12%	
10	2,9%	20,7%	7,35%	6,54%	6,98%	7,62%	13,1%	14,08%	18,56%	19,98%	19,93%	13,94%	14,03%	13,61%	14,11%	12,61%	6,6%	6,86%	6,63%	6,68%	6,46%	6,61%	8,15%	22,51%	3,03%	
11	3,04%	17,75%	9,77%	9,97%	11,16%	11,92%	12,46%	6,5%	6,67%	7,98%	6,92%	7,18%	7,03%	6,73%	7,47%	10,66%	7,3%	7,45%	7,29%	7,3%	7,06%	7,49%	8,39%	17,86%	2,95%	
12	2,88%	12,01%	6,81%	6,38%	6,91%	7,16%	7,32%	7,58%	7,84%	8,88%	7,66%	7,71%	7,22%	7,48%	8,32%	10,89%	10,15%	9,9%	8,63%	8,31%	8,16%	8,11%	8,14%	13,16%	3,14%	
13	3,06%	3,11%	2,85%	3,08%	3,12%	2,93%	3,02%	2,88%	3,15%	2,93%	3,05%	2,87%	2,94%	3,09%	3,12%	3,03%	3,22%	2,97%	2,99%	2,98%	3,17%	3,42%	3,22%			

Abbildung A.17: Belegungsgrad für das Layout „Sorter_23x11x6“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1				18,17%		14,56%		6,46%		6,13%		5,84%		5,66%		5,74%		10,48%		12,47%		14,75%			
2		1,65%	3,31%	19,88%	8,99%	18,2%	7,76%	16,35%	13,56%	16,43%	13,54%	16,16%	13,24%	15,34%	11,94%	17,19%	8,02%	16,2%	10,38%	18,58%	9,77%	17,28%	4,38%	3,52%	
3	2,46%	10,54%	10,01%	15,83%	13,85%	16,25%	14,14%	13,07%	5,53%	5,3%	5,53%	4,62%	4,69%	4,94%	6,63%	23,95%	21,44%	22,49%	20,4%	21,19%	18,12%	18,81%	15,16%	5,95%	2,35%
4	2,42%	9,84%	3,9%	4,66%	5,68%	6,85%	19,54%	19,76%	14,65%	14,3%	13,88%	12,74%	12,46%	12,38%	8,92%	15,79%	5,87%	5,93%	6,13%	5,24%	5,68%	4,69%	14,19%	14,35%	2,34%
5	2,35%	9,98%	4,9%	5,35%	6,56%	5,97%	12,55%	4,11%	4,01%	4,52%	4,74%	12,45%	12,57%	17,83%	19,87%	21,14%	13,85%	13,28%	12,49%	11,13%	10,02%	3,81%	3,56%	12,56%	2,3%
6	2,25%	8,54%	2,92%	3,43%	5,79%	6,22%	12,68%	4,98%	11,94%	12,86%	13,48%	14,15%	3,07%	3,4%	4,43%	6,23%	5,33%	4,11%	3,46%	3,36%	7,67%	3,61%	4,4%	11,4%	2,32%
7	2,3%	7,48%	2,78%	10,9%	13,02%	15,03%	17,34%	12,51%	12,43%	6,01%	5,85%	5,84%	3,56%	4,32%	5,4%	6,34%	3,17%	2,72%	2,69%	3,96%	7,62%	7,51%	7,99%	13,33%	2,4%
8	2,34%	6,42%	3,7%	10,4%	2,83%	2,99%	2,84%	3,38%	2,66%	2,8%	8,28%	10,35%	10,37%	10,78%	11,37%	10,4%	7,67%	7,06%	3,02%	3,6%	2,84%	3,08%	3,49%	10,97%	2,17%
9	2,47%	9,94%	8,99%	9,65%	3,48%	3,88%	4,21%	4,6%	4,38%	3,15%	5,42%	2,61%	2,83%	2,84%	3,07%	3,51%	3,37%	7,12%	6,61%	6,25%	5,41%	5,33%	5,49%	10,87%	2,38%
10	2,21%	7,84%	2,62%	2,4%	2,66%	2,67%	4,98%	5,26%	7,07%	7,55%	7,6%	5,24%	5,27%	5,18%	6,26%	4,83%	2,61%	2,62%	2,46%	2,47%	2,39%	2,63%	3,02%	8,62%	2,31%
11	2,32%	6,82%	3,74%	3,61%	4,22%	4,53%	4,75%	2,41%	2,42%	2,97%	2,64%	2,7%	2,54%	2,53%	2,73%	3,63%	2,59%	2,6%	2,6%	2,6%	2,57%	2,84%	3,16%	6,84%	2,25%
12	2,19%	4,59%	2,6%	2,44%	2,63%	2,74%	2,8%	2,9%	2,99%	3,39%	2,92%	2,93%	2,76%	2,85%	3,18%	4,16%	3,87%	3,78%	3,29%	3,18%	3,05%	3,1%	3,11%	4,89%	2,39%
13	2,33%	2,37%	2,17%	2,34%	2,38%	2,24%	2,3%	2,19%	2,4%	2,23%	2,32%	2,19%	2,24%	2,35%	2,38%	2,31%	2,46%	2,26%	2,28%	2,27%	2,41%	2,6%	2,46%		

Abbildung A.18: Relativer Durchsatz für das Layout „Sorter_23x11x6“

A8 - Systemverhalten von Kreuzungen

Tabelle A.21: Darstellung des Durchsatzes auf dem Layout „Inter_4x4“ für unterschiedliche Werte von n_S

	$n_S = 1,0$	$n_S = 1,5$	$n_S = 2,0$	$n_S = 2,5$	$n_S = 3,0$
Durchsatz in TE/h	10935	10744	10682	10612	10540
Standardabweichung in TE/h	96	128	115	87	101

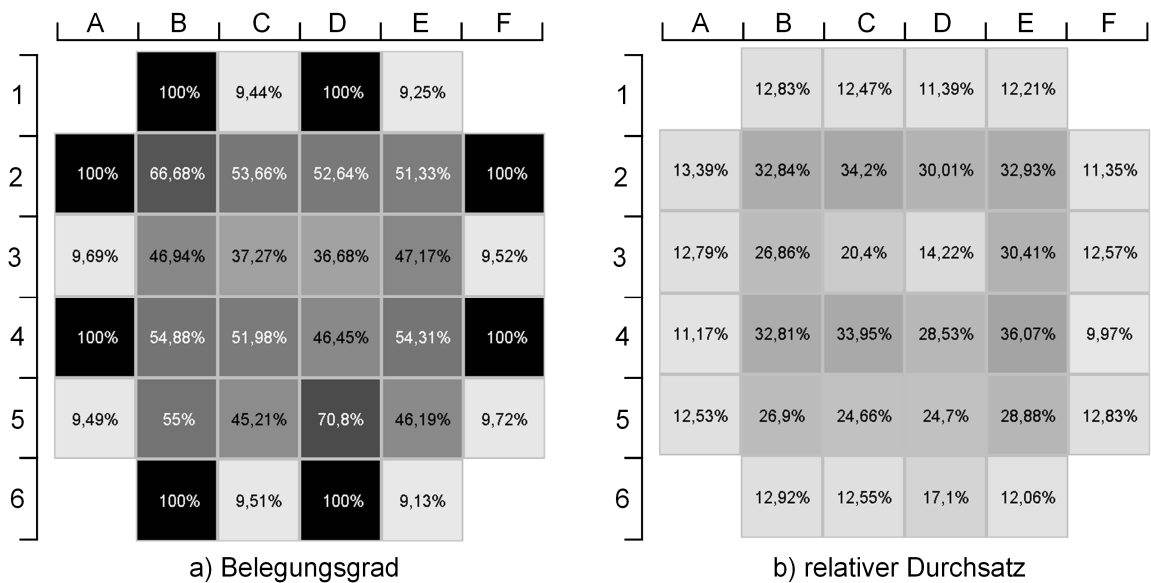


Abbildung A.19: Belegungsgrad und relativer Durchsatz für das Layout „Inter_4x4“

Tabelle A.22: Darstellung des Durchsatzes auf dem Layout „Inter_6x6“ für unterschiedliche Werte von n_S

	$n_S = 1,0$	$n_S = 1,5$	$n_S = 2,0$	$n_S = 2,5$	$n_S = 3,0$
Durchsatz in TE/h	13384	13271	13204	13163	13045
Standardabweichung in TE/h	150	119	135	163	169

	A	B	C	D	E	F	G	H
1		100%	7,81%	100%	7,4%	100%	7,85%	
2	100%	80,8%	46,91%	66,43%	38,5%	63,96%	43,98%	100%
3	7,33%	43,31%	51,13%	45,7%	47,09%	41,26%	48,39%	7,88%
4	100%	59,82%	40,69%	33,95%	25,48%	29,45%	40,19%	100%
5	7,31%	36,88%	46,36%	32,62%	38,52%	32,69%	47,67%	7,67%
6	100%	57,15%	37,93%	29,38%	23,61%	28,55%	37,09%	100%
7	7,77%	43,62%	42,3%	36,91%	38,82%	35,82%	39,49%	7,54%
8		100%	7,66%	100%	7,56%	100%	7,66%	

Abbildung A.20: Belegungsgrad für das Layout „Inter_6x6“

	A	B	C	D	E	F	G	H
1		10,31%	8,54%	9,79%	8,09%	9,96%	8,58%	
2	10,44%	22,27%	22%	17,55%	16,88%	15,97%	17,55%	7,26%
3	8,02%	18,53%	27,61%	23,39%	25,61%	21,91%	26,32%	8,61%
4	10,38%	18,29%	21,1%	7,82%	11,42%	7,6%	20,66%	5,86%
5	7,99%	16,67%	25,28%	16,6%	20,85%	16,72%	26,15%	8,39%
6	9,54%	16,57%	19,58%	6,06%	10,59%	6,84%	19,3%	5,84%
7	8,5%	17,31%	22,98%	18,08%	21,05%	17,45%	21,56%	8,25%
8		7,14%	8,38%	6,9%	8,27%	6,73%	8,38%	

Abbildung A.21: Relativer Durchsatz für das Layout „Inter_6x6“

Tabelle A.23: Darstellung des Durchsatzes auf dem Layout „Inter_8x8“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	15138	15101	14969	14950	14849
Standardabweichung in TE/h	155	157	174	232	216

	A	B	C	D	E	F	G	H	I	J
1		100%	6,54%	100%	6,86%	100%	6,1%	100%	6,42%	
2	100%	82,93%	40,68%	67%	35,41%	63,27%	34,1%	57,99%	39,47%	100%
3	6,14%	43,96%	42,73%	39,59%	41,47%	36,61%	31,88%	34,99%	35,39%	6,43%
4	100%	45,38%	26,34%	28,35%	38,4%	36,2%	39,2%	34,27%	41,29%	100%
5	6,34%	35,23%	27,34%	23,74%	28,92%	26,52%	21,71%	24,1%	36,67%	6,14%
6	100%	37,94%	36,38%	31,23%	35,15%	25,73%	29,73%	26,06%	35,04%	100%
7	6,69%	29,95%	23,73%	23,12%	21,83%	21,18%	34,37%	29,72%	37,08%	6,72%
8	100%	36,53%	33,27%	28,89%	24,62%	23,2%	19,64%	29,82%	32,97%	100%
9	6,58%	43,37%	29,63%	35,51%	34,6%	35,86%	32,55%	33,51%	32,38%	6,38%
10		100%	6,77%	100%	6,85%	100%	6,73%	100%	6,33%	

Abbildung A.22: Belegungsgrad für das Layout „Inter_8x8“

	A	B	C	D	E	F	G	H	I	J
1		8,94%	6,29%	8,7%	6,6%	8,73%	5,87%	7,84%	6,18%	
2	8,72%	18,53%	15,19%	14,21%	13,83%	13,61%	12,5%	12,37%	13,89%	5,63%
3	5,9%	19,43%	19,83%	17,2%	19,23%	12,6%	13,08%	10,35%	16,13%	6,18%
4	6,96%	15,39%	8,61%	5,21%	18,37%	16,36%	18,41%	15,91%	20,03%	4,07%
5	6,1%	15,78%	11,59%	7,96%	13,6%	5,86%	8,77%	5,84%	17,47%	5,9%
6	4,71%	17,95%	17,43%	14,05%	16,83%	10,57%	13,93%	10,37%	16,62%	4,31%
7	6,43%	14,06%	8,82%	5,17%	8,56%	9,09%	16,5%	13,73%	17,58%	6,46%
8	5,48%	15,25%	15,42%	12,58%	11,3%	8,67%	7,42%	4,32%	14,59%	4,29%
9	6,32%	12,54%	11,86%	13,83%	16,38%	15,05%	15,6%	12,64%	15,48%	6,14%
10		5,92%	6,51%	5,37%	6,59%	4,65%	6,47%	5,86%	6,08%	

Abbildung A.23: Relativer Durchsatz für das Layout „Inter_8x8“

Tabelle A.24: Darstellung des Durchsatzes auf dem Layout „Inter_12x12“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	15581	15581	15491	15601	15551
Standardabweichung in TE/h	223	297	214	229	175

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		100%	4,24%	100%	4,42%	100%	4,31%	100%	4,28%	100%	4,57%	100%	4,72%	
2	100%	80,95%	30,76%	50,51%	30,91%	29,79%	30,3%	42,79%	28,3%	36,94%	26,49%	58,71%	34,69%	100%
3	4,47%	40,36%	28,29%	26,53%	21,26%	16,36%	26,31%	26,65%	30,37%	27,84%	23,04%	30,22%	24,69%	4,65%
4	100%	44,74%	28,82%	25,4%	14,19%	16,38%	15,53%	14,4%	19,65%	22,46%	27,68%	26,32%	29,48%	100%
5	4,41%	31,18%	23,17%	20,4%	18,3%	15,97%	15,49%	9,6%	19,19%	9,56%	15,24%	20,47%	29,39%	4,11%
6	100%	35,04%	30,18%	18,33%	11,47%	9,49%	9,14%	12,71%	19,4%	10,25%	14,96%	19,87%	27,45%	100%
7	4,4%	25,41%	32,83%	28,22%	26,41%	13,92%	16,34%	18,22%	20,37%	7,13%	17,57%	15,43%	32,29%	4,33%
8	100%	33,51%	24,44%	22,92%	23,29%	23,95%	27,1%	23,12%	10,46%	8,87%	12,19%	14,16%	26,99%	100%
9	4,25%	24,46%	25,48%	14,02%	19,16%	15,08%	20,35%	20,53%	25,45%	21,17%	14,36%	12,61%	30,84%	4,49%
10	100%	32,66%	21,35%	14,05%	10,42%	6,19%	10,01%	8,97%	13,16%	14,99%	9,26%	14,59%	26,52%	100%
11	4,51%	23,5%	21,31%	12,81%	12,44%	9,43%	13,85%	13,7%	17,27%	19,73%	16,49%	18,7%	27,82%	4,35%
12	100%	26,85%	26,24%	15,41%	17,48%	16,75%	18,72%	14,54%	14,28%	18,17%	17,76%	18,63%	35,85%	100%
13	4,44%	35,47%	23,84%	25,22%	25,97%	24,79%	26,12%	26%	23,71%	25,29%	26,59%	27,95%	22,41%	4,5%
14		100%	4,32%	100%	4,67%	100%	4,45%	100%	4,24%	100%	4,67%	100%	4,33%	

Abbildung A.24: Belegungsgrad für das Layout „Inter_12x12“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		8,11%	4%	7,07%	4,17%	3,5%	4,06%	5%	4,04%	5,16%	4,3%	6,37%	4,44%	
2	7,98%	16,78%	10,62%	11,98%	14,56%	13,26%	13,92%	10,28%	9,72%	11,34%	9,89%	9,66%	9,45%	4,05%
3	4,21%	15,01%	10,92%	10,22%	9,72%	4,44%	12,24%	11,87%	14,31%	12,56%	7,32%	7,84%	10,95%	4,38%
4	4,56%	13,64%	11,81%	9,41%	5,01%	3,53%	6,45%	2,54%	8,9%	10,2%	12,97%	12,27%	10,57%	3,4%
5	4,15%	13,78%	9,39%	8,92%	8,42%	7,16%	7,14%	3,25%	9,02%	3,08%	6,92%	9,02%	13,89%	3,87%
6	3,64%	13,08%	12,74%	7,18%	3,94%	3%	3,79%	2,47%	9,09%	3,65%	6,63%	2,63%	12,78%	2,21%
7	4,14%	9,93%	15,5%	12,78%	12,35%	5,8%	6,4%	8,16%	9,44%	2,7%	8,3%	6,67%	15,28%	4,08%
8	4,72%	10,34%	10,51%	3,3%	10,35%	8,72%	12,81%	10,7%	3,56%	2,07%	5,59%	2,55%	12,54%	2,13%
9	4%	10,22%	12,01%	5,95%	8,86%	6,51%	9,43%	9,67%	12,01%	9,92%	6,73%	3,91%	14,5%	4,23%
10	3,9%	9,84%	9,6%	1,96%	4,56%	1,58%	4,37%	1,56%	4,57%	6,21%	3,35%	2,73%	12,48%	2,11%
11	4,25%	10,52%	9,25%	2,98%	5,78%	3,02%	5,85%	5,93%	7,84%	9,22%	7,37%	8,39%	13,16%	4,1%
12	3,16%	11,83%	12,42%	4,87%	7,36%	7,01%	8,75%	6,03%	5,31%	6,79%	8,15%	7,73%	8,43%	4,41%
13	4,19%	8,45%	10,6%	9,66%	12,25%	11,31%	11,32%	10,13%	11,19%	11,96%	12,56%	9,51%	10,23%	4,24%
14		3,95%	4,07%	3,21%	4,4%	2,94%	4,19%	3,19%	4%	2,3%	4,4%	3,18%	4,08%	

Abbildung A.25: Relativer Durchsatz für das Layout „Inter_12x12“

Tabelle A.25: Darstellung des Durchsatzes auf dem Layout „Inter_14x14“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	14959	14891	14746	14677	14646
Standardabweichung in TE/h	270	192	223	198	189

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		100%	4,08%	100%	4,15%	100%	3,9%	100%	4,15%	100%	3,71%	100%	4,17%	100%	3,95%	
2	100%	82,69%	34,33%	36,27%	25,28%	33,47%	27,82%	30,04%	22,64%	26,98%	25,36%	25,18%	19,47%	35,98%	39,05%	100%
3	3,98%	33,33%	28,27%	28,77%	26,88%	26,34%	14,72%	16,21%	18,78%	21,37%	14,06%	15,89%	18,28%	25,26%	22,81%	4,1%
4	100%	55,06%	29,59%	26,91%	31,45%	28,92%	15,09%	18,86%	17,3%	22,15%	16,85%	14,84%	19,56%	22,14%	35%	100%
5	3,99%	28,76%	22,99%	13,72%	13,88%	24,47%	27,22%	18,42%	15,99%	22,82%	11,15%	8,32%	10,25%	15,48%	26,89%	3,97%
6	100%	41,57%	25,12%	18,21%	12,8%	20,11%	25,81%	23,42%	18,68%	19,99%	10%	7,29%	7,76%	11,93%	25,96%	100%
7	4,43%	23%	22,74%	18,67%	18,41%	15,07%	16,9%	23,06%	26,84%	22,92%	11,23%	8,27%	11,87%	9,74%	27,75%	4,15%
8	100%	36,07%	26,54%	19,32%	26,32%	23,12%	27,02%	22,07%	35,46%	32,71%	14,37%	10,83%	10,25%	12,37%	24,84%	100%
9	3,86%	25,14%	22,79%	13,2%	11,15%	10,52%	10,64%	9,64%	13,78%	24,93%	26,64%	15,13%	14,48%	20,19%	27,57%	4,13%
10	100%	29,16%	23,97%	12,75%	9,12%	8,03%	8,91%	8,33%	9,75%	17,46%	24,53%	19,78%	17,9%	21,32%	34,16%	100%
11	3,99%	29,05%	12,54%	10,27%	13,82%	8,85%	8,74%	6,59%	11,69%	9,48%	15,05%	18,84%	22,85%	22,06%	24,98%	3,89%
12	100%	25,71%	12,11%	10,46%	14,35%	12,25%	6,89%	3,79%	8,79%	6,8%	19,24%	17,77%	26,63%	24,91%	35,65%	100%
13	4,37%	25,94%	12,51%	8,96%	10,42%	11,12%	15,22%	10,87%	14,93%	10,32%	20,42%	14,92%	12,33%	18,05%	25,87%	3,87%
14	100%	23,74%	13,56%	12,78%	10,79%	17,93%	17,67%	14,42%	12,09%	14,74%	19,61%	12,96%	13,26%	25,03%	21,77%	100%
15	4,07%	26,43%	19,42%	21,86%	23,67%	23,15%	25,24%	23,34%	27,91%	24,39%	26,03%	25,13%	23,41%	22,33%	21,06%	3,85%
16		100%	4,15%	100%	4,25%	100%	4,36%	100%	4,03%	100%	3,76%	100%	4%	100%	4%	

Abbildung A.26: Belegungsgrad für das Layout „Inter_14x14“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		6,26%	3,6%	3,84%	3,66%	4,23%	3,44%	3,88%	3,66%	3,4%	3,27%	2,77%	3,68%	4,66%	3,49%	
2	6,47%	13,08%	13,35%	12,26%	8,75%	11,37%	11,03%	10,59%	9,32%	11,71%	11,03%	9,43%	7,67%	9,2%	8,46%	4,31%
3	3,51%	10%	10,26%	8,81%	11,29%	8,97%	5,53%	5,66%	7,56%	8,83%	5,15%	6,09%	7,9%	8,06%	8,04%	3,62%
4	5,88%	10,03%	12,09%	9,05%	13,75%	12,53%	4,47%	2,75%	5,17%	8,91%	5,94%	5,35%	8,5%	8,18%	8,83%	4,67%
5	3,52%	8,94%	9,97%	4,5%	4,76%	10,26%	11,98%	5,31%	6,69%	9,84%	4,6%	3,28%	4,19%	5,98%	11,89%	3,5%
6	4,78%	9,27%	10,65%	5,6%	4,14%	2,18%	11,01%	9,52%	4,39%	8,22%	3,25%	1,9%	3,01%	2,12%	10,95%	2,56%
7	3,91%	9,06%	9,17%	5,37%	7,57%	3,95%	6,42%	9,35%	11,59%	9,87%	4,65%	3,29%	5,26%	3,51%	12,3%	3,66%
8	4,23%	9,03%	11,47%	8,24%	11,72%	10%	11,99%	9,55%	15,72%	14,47%	3,48%	1,99%	3,87%	2,35%	10,73%	2,36%
9	3,41%	8,76%	9,29%	4,33%	4,34%	3,13%	4,61%	3,22%	4,61%	10,9%	11,64%	4,74%	6,09%	8,29%	12,17%	3,65%
10	1,95%	12,89%	10,35%	5,12%	3,59%	2,95%	3,94%	2,46%	3,47%	2,07%	10,82%	8,62%	4,27%	8,29%	8,71%	4,24%
11	3,52%	12,81%	4,61%	4,19%	6%	3,63%	3,84%	2,68%	5,17%	2,83%	5,77%	7,74%	9,68%	9,51%	8,68%	3,44%
12	2,04%	11,12%	4,31%	2,81%	6,19%	4,32%	2,97%	0,93%	3,82%	1,56%	8,21%	6,14%	11,77%	11%	7,63%	3,47%
13	3,85%	11,48%	4,77%	3,08%	4,45%	4,6%	6,69%	4,3%	6,58%	4,25%	8,65%	3,44%	4,33%	7,18%	11,15%	3,41%
14	2,62%	9,13%	5,11%	3,36%	3,6%	1,75%	7,67%	5,67%	4,25%	2,43%	8,59%	5,1%	4,11%	2,32%	8,78%	2,44%
15	3,59%	8,94%	8,23%	8,59%	10,44%	9,18%	11,15%	9,9%	12,31%	10,72%	11,55%	9,08%	10,12%	7,75%	9,28%	3,4%
16		3,46%	3,66%	3,19%	3,75%	2,26%	3,85%	2,4%	3,55%	1,98%	3,32%	3,02%	3,54%	2,92%	3,53%	

Abbildung A.27: Relativer Durchsatz für das Layout „Inter_14x14“

Tabelle A.26: Darstellung des Durchsatzes auf dem Layout „Inter_16x16“ für unterschiedliche Werte von n_s

	$n_s = 1,0$	$n_s = 1,5$	$n_s = 2,0$	$n_s = 2,5$	$n_s = 3,0$
Durchsatz in TE/h	14536	14428	14533	14368	14403
Standardabweichung in TE/h	308	270	342	289	243

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		100%	3,44%	100%	3,19%	100%	3,51%	100%	3,52%	100%	3,22%	100%	3,23%	100%	3,58%	100%	3,31%	
2	100%	50,36%	22,69%	32,02%	23,76%	29,95%	22,69%	26,39%	23,44%	33,31%	24,28%	26,18%	22,01%	46,67%	22,75%	57,45%	33,55%	100%
3	3,33%	28,58%	15,2%	15,46%	17,82%	20,2%	12,57%	15,48%	18,91%	19,45%	11,81%	15,88%	17,26%	23,72%	21,01%	24,81%	22,66%	3,1%
4	100%	29,65%	15,8%	14,94%	10,14%	13,97%	16,04%	15,73%	9,98%	11,23%	10,5%	10,11%	15,68%	18,55%	15,92%	24,55%	21,81%	100%
5	3,81%	30,38%	10,86%	9,61%	8,44%	5,97%	19,63%	17,02%	10,11%	10,16%	12,46%	7,49%	12,3%	15,43%	17,8%	16,63%	24,95%	3,41%
6	100%	27,49%	10,59%	8,81%	9,27%	12,97%	11,97%	11,77%	8,74%	9,70%	11,47%	15,61%	15,12%	10,79%	20,30%	18,00%	20,02%	100%
7	3,54%	29,1%	15,08%	8,02%	13,39%	15,3%	18,78%	17,71%	20,78%	17,9%	22,44%	18,99%	23,58%	24,22%	12,73%	13,74%	21,29%	3,14%
8	100%	26,86%	16,68%	12,25%	14,16%	10,59%	7,37%	3,72%	5,13%	5,97%	6,48%	14,15%	18,39%	22,26%	17,86%	16,05%	23,97%	100%
9	3,46%	30,46%	20,66%	14,38%	7,45%	5,55%	8,38%	3,88%	6,47%	5,66%	10,4%	16,98%	19,39%	14,26%	16,39%	18,65%	20,15%	3,31%
10	100%	23,49%	19,7%	16,24%	6,31%	4,39%	6,55%	2,19%	5,62%	12,97%	17,94%	15,83%	7,85%	5,18%	16,24%	17,42%	25,63%	100%
11	9,01%	25,21%	10,19%	13,93%	9,51%	6,72%	12,07%	9,44%	12,57%	15,42%	7,47%	8,91%	9,09%	6,73%	10,13%	10,72%	25,09%	3,40%
12	100%	22,11%	11,1%	12,81%	10,17%	7,23%	6,33%	8,5%	19,68%	17,75%	8,1%	6,36%	7,07%	3,44%	5,96%	10,67%	25,99%	100%
13	3,41%	21,14%	12,42%	12,32%	9,59%	9,93%	13,3%	14,61%	18,17%	12,88%	7,97%	4,79%	8,34%	6,29%	8,87%	11,52%	25,84%	3,33%
14	100%	21,77%	14,58%	12,54%	9,2%	13,14%	18,97%	16,68%	10,85%	10,54%	9,33%	7,4%	10,13%	8,1%	14,49%	14,06%	25%	100%
15	3,42%	21,84%	18,38%	19,73%	21,03%	17,67%	16,91%	16,77%	17,59%	17,43%	15,93%	15,83%	18,78%	18,22%	15,35%	14,66%	21,03%	3,42%
16	100%	25,69%	15,5%	23,8%	13,62%	20,28%	22,91%	17,9%	16,21%	15,14%	14,27%	15,14%	13,59%	17,45%	21,52%	19,82%	36,35%	100%
17	3,36%	25,31%	19,49%	22,16%	23,09%	22,18%	18,76%	21,57%	23,91%	23,01%	21,24%	24,04%	23,36%	22,43%	20,79%	30,52%	17,42%	3,37%
18		100%	3,6%	100%	3,48%	100%	3,56%	100%	3,55%	100%	3,39%	100%	3,49%	100%	3,3%	100%	3,47%	

Abbildung A.28: Belegungsgrad für das Layout „Inter_16x16“

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		5,07%	3,17%	4,32%	2,93%	3,9%	3,23%	3,07%	3,24%	3,62%	2,96%	3,87%	2,97%	5,82%	3,3%	6,21%	3,05%	
2	4,89%	11,77%	8,22%	9,48%	9,96%	10,3%	10,03%	9,81%	10,51%	10,51%	10,53%	10,17%	9,81%	9,17%	7,81%	8,84%	7,58%	3,92%
3	3,07%	11,4%	5,35%	5,38%	8,17%	8,38%	4,73%	5,05%	8,72%	8,46%	4,46%	2,48%	6,71%	6,64%	9,52%	8,16%	10,17%	2,85%
4	2,47%	12,1%	6,29%	5,63%	4,16%	5,36%	7,21%	4,57%	3,35%	3,98%	3,92%	2,62%	6,98%	5,2%	5,71%	2,24%	9,67%	2,2%
5	3,51%	13,63%	4,6%	3,47%	3,88%	2,34%	9,06%	7,56%	4,16%	4,31%	5,75%	2,82%	5,5%	4,25%	7,98%	6,37%	11,41%	3,14%
6	1,85%	12%	3,36%	2,4%	4,06%	1,56%	4,69%	3,89%	3,04%	1,46%	4,7%	2,39%	4,8%	6,47%	9,38%	7,38%	9,17%	2,94%
7	3,26%	13,14%	5,39%	3,11%	5,51%	6,76%	8,67%	8,11%	9,58%	8,26%	10,36%	8,56%	10,36%	11,18%	4,18%	4,74%	9,42%	2,89%
8	2,09%	11,98%	4,75%	5,08%	6,38%	4,86%	2,62%	1,09%	2,34%	1,52%	2,69%	1,33%	8,15%	10,06%	4,47%	5,61%	9,63%	3,06%
9	3,18%	14,05%	9,49%	5,6%	3,43%	2,56%	3,86%	1,65%	2,98%	2,29%	4,36%	7,47%	8,89%	6,27%	7,53%	5,36%	9,06%	3,04%
10	1,42%	10,63%	8,59%	7,33%	2,33%	0,6%	2,86%	0,91%	2,36%	5,94%	8,25%	6,97%	2,39%	1,89%	7,21%	7,68%	11,75%	2,2%
11	2,77%	11,63%	3,83%	5,89%	4,38%	2,89%	5,56%	4,18%	5,6%	6,75%	3,23%	3,39%	4,2%	2,96%	3,89%	3,75%	11,86%	3,18%
12	1,63%	10,06%	4,18%	5,54%	4,31%	3,23%	2,71%	0,84%	9,05%	8,05%	3,58%	2,64%	3,14%	0,86%	2,74%	2,94%	11,73%	2,03%
13	3,14%	9,75%	5,49%	5,44%	3,83%	4,51%	6,11%	6,5%	8,2%	4,03%	3,39%	1,39%	3,85%	2,13%	3,85%	2,8%	11,96%	3,07%
14	2,81%	8,04%	5,06%	4,97%	3,42%	2,04%	8,75%	7,49%	2,91%	3,26%	4,24%	2,49%	4,56%	2,87%	6,71%	5,51%	11,4%	2,06%
15	3,14%	9,78%	8,46%	8,92%	9,7%	8,02%	6,54%	6,39%	8,01%	7,9%	7,33%	6,83%	8,66%	8,21%	6,85%	4,7%	9,47%	3,14%
16	3,65%	7,12%	3,63%	1,96%	3,3%	6,71%	10,36%	7,4%	2,86%	4,72%	5,66%	4,64%	2,92%	6,27%	7,61%	7,52%	8,32%	4,91%
17	3,09%	7,65%	8,76%	8,46%	10,49%	9,37%	8,04%	9,98%	10,95%	9,73%	9,53%	10,23%	10,53%	9%	7,75%	8,59%	7,27%	3,1%
18		2,82%	3,31%	2,02%	3,2%	2%	3,28%	2,17%	3,26%	1,93%	3,12%	2,39%	3,21%	2,21%	3,03%	4,81%	3,19%	

Abbildung A.29: Relativer Durchsatz für das Layout „Inter_16x16“

A9 - Dauer der Simulationsläufe

In diesem Kapitel wird untersucht, wie lange ein durchschnittlicher Simulationslauf in Abhängigkeit von der Anzahl der Module dauert. Die Dauer eines Simulationslaufes wird hierbei nicht in „Simulationszeit“ gemessen, sondern es wird die Zeit angegeben, die der verwendete Computer zur Durchführung des Simulationslaufes benötigt hat. Die Untersuchung der Dauer der Simulationsläufe ist sinnvoll, damit z.B. zukünftige simulationsgestützte Untersuchungen besser planbar werden.

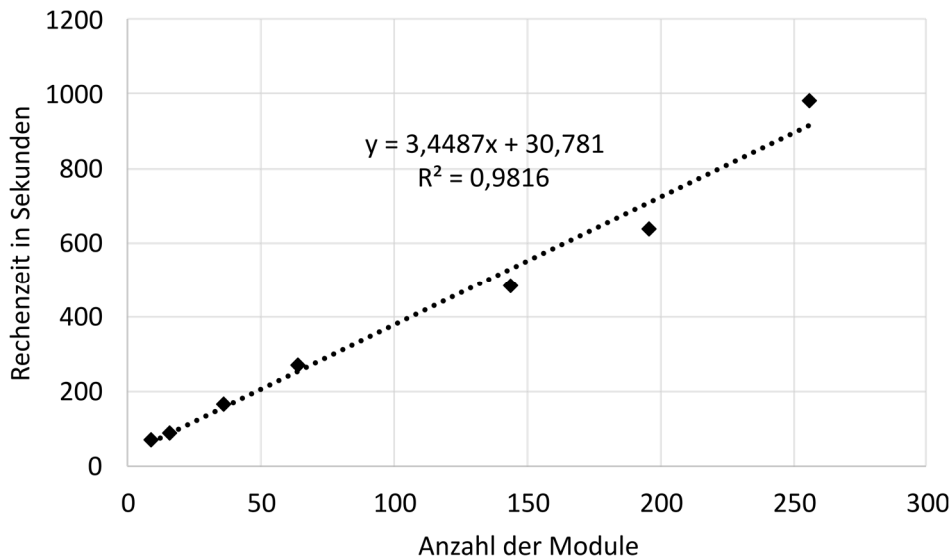


Abbildung A.30: Dauer der Simulationsläufe in Abhängigkeit von der Modulanzahl

In der Abbildung A.30 ist die Dauer der Simulationsläufe in Abhängigkeit von der Modulanzahl dargestellt für die quadratischen Sorterlayouts. Zur Durchführung wurde hierbei immer der gleiche Computer genutzt. Dieser Computer verfügt über 8 GB RAM und einen Intel Core i7-6600U CPU mit zwei Kernen, die mit 2,60 GHz getaktet sind. Der RAM Bedarf der Simulation ist hierbei annähernd konstant und beträgt ca. 200 MB. Bei der Durchführung der Simulation ist somit die Rechengeschwindigkeit der limitierende Faktor. Um das Debuggen zu erleichtern, wurde die Simulation als Single-Core Applikation geschrieben.

Anhand der Trendlinie ist zu erkennen, dass die Rechenzeit linear ansteigt mit der Anzahl der Module. Somit ist es möglich, dass auch größere Layouts effektiv simuliert werden können.

Weiterhin kann über die Dauer der Simulationsläufe auch die durchschnittliche Laufzeit des Algorithmus für ein Modul bestimmt werden. Im Kapitel 6.3.2 wurde die Laufzeit eines einzelnen Moduls bereits für den schlechtmöglichsten Fall bestimmt, jedoch noch nicht die durchschnittliche Laufzeit. Die Laufzeit eines Simulationslaufes setzt sich aus den Einzellaufzeiten der Module zusammen. Da mit steigender Anzahl der

Module die Simulationszeit proportional ansteigt, müssen die Einzelaufzeiten der Module konstant bleiben. Die Erklärung hierfür ist die Folgende: Da in den größeren Layouts mehr Quellen vorhanden sind, sind auch mehr Transporteinheiten im System vorhanden. Somit erhöht sich die Anzahl der Nachrichten. Da sich jedoch in gleichem Maße die Anzahl der Module erhöht hat, bleibt die Anzahl der Nachrichten für die einzelnen Module annähernd konstant und somit bleibt auch die Laufzeit konstant. Aus diesem Grund ist die benötigte Rechenkapazität eines einzelnen Moduls unabhängig von der Größe des Layouts, in dem es eingesetzt wird.

Lebenslauf

Persönliche Daten

Name	Simon Sohrt
Geburtsdatum	07.01.1987
Geburtsort	Neustadt a. Rbge.

Schulbildung

1993 - 1997	Grundschule Steinhude
1997 - 1999	Orientierungsstufe Steinhude
1999 - 2006	Hölty-Gymnasium Wunstorf (Abschluss: Abitur)

Studium

09/2007 - 01/2011	Maschinenbau (Konstruktionstechnik) als duales Studium an der Fachhochschule Hannover (Abschluss: Bachelor of Engineering)
04/2011 - 07/2013	Maschinenbau an der Leibniz Universität Hannover (Abschluss: Master of Science)

Berufliche Tätigkeiten

08/2006 - 07/2007	Zivildienstleistender in der Kindertagesstätte der Lebenshilfe Seelze e.V., Wunstorf, Deutschland
09/2007 - 07/2013	Dualer Student bei der Komatsu Hanomag GmbH, Hannover, Deutschland
09/2013 - 11/2013	Auslandspraktikum bei der Newcastle Innovation Ltd., Newcastle, Australien
05/2019 - 07/2019	Gastwissenschaftler an dem Logistics and Distribution Institute der University of Louisville, Louisville, USA
02/2014 - 05/2020	Wissenschaftlicher Mitarbeiter am Institut für Transport- und Automatisierungstechnik der Leibniz Universität Hannover, Hannover, Deutschland
Ab 09/2020	Softwareentwickler bei der HIS Hochschul-Informationen-System eG, Hannover, Deutschland