

Entwicklung einer generischen Simulationsmethode für das zeitfensterbasierte Routing Fahrerloser Transportfahrzeuge

Development of a generic simulation method for the time window routing of automated guided vehicles

Thomas Lienert
Johannes Fottner

Lehrstuhl für Fördertechnik Materialfluss Logistik
Fakultät für Maschinenwesen
Technische Universität München

Fahrerlose Transportfahrzeuge werden unter anderem zur Produktionsversorgung und in automatisierten Lagersystemen eingesetzt. Die Anzahl an Fahrzeugen, welche notwendig ist, um einen geforderten Durchsatz zu erzielen, kann auf analytischem Wege nur näherungsweise bestimmt werden. Sind exakte Aussagen über den Durchsatz in Abhängigkeit der Fahrzeuganzahl gefordert, ist die Durchführung einer Simulationsstudie notwendig. Die Erstellung eines Simulationsmodells ist jedoch aufwändig und erfordert Kenntnisse über Steuerungsalgorithmen, um ein ablauffähiges Simulationsmodell zu erhalten.

Der Beitrag beschreibt vor diesem Hintergrund eine Methode zur Simulation von Fahrerlosen Transportsystemen, um eine aufwandsarme Leistungsbestimmung zu ermöglichen. Ein Fahrerloses Transportsystem wird allgemeingültig modelliert, damit es in eine Simulationsumgebung überführt werden kann. Es werden Strategien zum Routing und Deadlock-Handling beschrieben, welche in das Modell hinterlegt und auf beliebigen Layouts angewendet werden können. Somit entsteht ein ablauffähiges Modell, welches zur individuellen Systemanalyse herangezogen werden kann.

[Schlüsselwörter: FTS, Kommissionierroboter, Durchsatzanalyse, Simulation, Zeitfensterbasiertes Routing]

Automated guided vehicles are used for the supply of production areas, in automated storage and retrieval systems and for the supply of picking areas in goods to person fulfillment systems. The number of vehicles, that is required to reach a given throughput cannot be calculated exactly in an analytical way. Therefore, the performance of a simulation study is necessary. Implementing the model in a simulation environment requires

time and effort. Basic algorithms are needed in order to obtain an executable simulation model.

In this work, we describe a method for the simulation of automated guided vehicle systems for the measurement of the throughput as a function of the number of vehicles. We model an automated guided vehicle system in order to transfer it into a simulation environment. We describe algorithms concerning the routing and deadlock-handling, that will be integrated into the model and could be used on different layouts. As a result we obtain an executable simulation model that can be used for the detailed analysis of individual automated guided vehicle systems.

[Keywords: AGV, MRFS, Throughput Analysis, Simulation, Time-Window Routing]

1 AUSGANGSSITUATION

Fahrerlose Transportsysteme (FTS) sind meist innerbetriebliche, flurgebundene Fördersysteme mit automatisch gesteuerten Fahrzeugen, deren Aufgabe der Transport von Gütern ist. Ein FTS besteht im Wesentlichen aus einem oder mehreren Fahrerlosen Transport Fahrzeugen (FTF), einer Leitsteuerung und einer Einrichtung zur Standortbestimmung und Datenübertragung [VDI-2510]. FTS werden unter anderem zur Produktionsversorgung und in automatisierten Lagersystemen eingesetzt. Ein Anwendungsfall von FTS ist die automatisierte „Ware-zur-Person“-Kommissionierung. Die FTF, welche in diesem Kontext auch als Kommissionierroboter bezeichnet werden, lagern die Behälter oder Regale, welche die zu kommissionierenden Artikel beinhalten, aus und transportieren diese in die entsprechende Kommissionierzone, wo die Artikel entnommen werden. Diese Systeme werden in der Literatur als mobile robotic fulfillment systems (MRFS) bezeichnet [Lam-2016].

Eine Eigenschaft von MRFS ist, dass jedes Fahrzeug jeden Lager- und Übergabeplatz im System anfahren und somit jeden Ein- und Auslagerauftrag ausführen kann. Vorteile, welche sich gegenüber herkömmlichen automatisierten Kommissioniersystemen ergeben, sind:

- das Layout kann variabel gestaltet werden,
- durch das Hinzufügen oder Entnehmen von Fahrzeugen kann der Systemdurchsatz skaliert werden,
- durch Redundanzen entsteht eine höhere Verfügbarkeit des Gesamtsystems,
- eine geforderte Sequenz kann bereits im Lager hergestellt werden, sodass die Sequenzierung in der Vorzone entfällt.

Jedoch erfordern solche Systeme komplexe Steuerungsstrategien, um einen robusten und effizienten Betrieb zu ermöglichen. Jedes Fahrzeug kann jeden Auftrag ausführen und mehrere Fahrzeuge bewegen sich im gleichen System. Somit müssen Strategien für die Koordination der Fahrzeuge untereinander eingesetzt werden. Dabei handelt es sich um Strategien für die Zuweisung von Aufträgen an Fahrzeuge, die Wegfindung und das Deadlock-Handling zur Vermeidung wechselseitiger Blockaden zwischen den Fahrzeugen.

2 PROBLEMSTELLUNG UND ZIELSETZUNG

Die Leistungsbestimmung von FTS ist eine Herausforderung. Von Interesse ist beispielsweise die Grenzleistung eines Systems oder die Anzahl der Fahrzeuge, die notwendig ist, um einen geforderten Durchsatz zu erzielen. Diese Fragestellungen müssen bereits in einer frühen Planungsphase oder sogar schon in der Angebotsphase beantwortet werden, da die Anzahl der Fahrzeuge für die Wirtschaftlichkeit der Systeme ausschlaggebend ist [Sch-1989]. Analytische Modelle zur Bestimmung der notwendigen Anzahl an Fahrzeugen sind ungenau und bieten nur eine Näherungslösung. Dies ist zum einen darauf zurückzuführen, dass die Leerfahrten, die zwischen zwei Transportaufträgen anfallen, von den gewählten Dispositionsstrategien abhängen und analytisch nicht zu ermitteln sind. Zum anderen können die Blockierungseffekte zwischen den Fahrzeugen nur geschätzt werden. [Le-2006; Gan-1998]

In einem komplexen System kann die notwendige Anzahl an Fahrzeugen daher wirtschaftlich nur durch die Durchführung einer Simulationsstudie erfolgen, auch wenn der Aufwand dafür erheblich ist [Le-2006; Vis-2006; Sch-1989; VDI-2710]. Der größte Aufwand bei der Durchführung einer Simulationsstudie zur Bestimmung der Anzahl der Fahrzeuge entsteht durch die Modellerstellung. Das Layout des betrachteten Systems muss in eine Simulationsumgebung überführt werden. Darüber hinaus müssen

Routing und Deadlock-Handling-Strategien bekannt sein und in das Modell integriert werden, um ein ablauffähiges Simulationsmodell zu erhalten.

Für den Umgang mit Deadlocks werden drei generische Ansätze unterschieden: die Prävention, das Erkennen und Auflösen sowie die Vermeidung von Deadlocks [Liu-2001]. Die Deadlock-Prävention stellt durch Anwendung einfacher, statischer Regeln sicher, dass im System keine Deadlocks entstehen können. Jedoch führt die Deadlock-Prävention zu einer geringen Ressourcenauslastung. Das Erkennen und Auflösen von Deadlocks ist nur bei einer geringen Anzahl an Fahrzeugen ein praktikabler Ansatz, da es bei einer hohen Fahrzeugdichte zu verketteten Deadlocks kommen kann, die durch allgemeingültige Regeln nicht mehr aufgelöst werden können. Bei der Deadlock-Vermeidung werden die Ressourcen dynamisch auf Basis des aktuellen Systemzustandes derart zugewiesen, dass keine Deadlocks entstehen können. Dieser Ansatz erfordert zur Umsetzung komplexe Strategien, ermöglicht aber eine hohe Ressourcennutzung.

Eine Möglichkeit, die Vermeidung von Deadlocks umzusetzen, ist das zeitfensterbasierte Routing. Das Layout wird dafür durch einen Graphen abstrahiert. Für jeden Knoten wird eine Liste mit reservierten und freien Zeitfenstern gespeichert. Die Berechnung einer neuer Route erfolgt zeitoptimal durch die freien Zeitfenster von Knoten zu Knoten, bis das Ziel erreicht wird. [Kim-1991; Mor-2009]

Die Erstellung eines Simulationsmodells zur Bestimmung der notwendigen Anzahl an Fahrzeugen ist zeitintensiv, erfordert Fachkenntnisse hinsichtlich der Simulationsumgebung sowie hinsichtlich der Steuerungsstrategien, die notwendig sind, um überhaupt ein ablauffähiges Simulationsmodell zu erhalten. Ist das Modell erstellt, erlaubt es nicht nur die Bestimmung der Leistung in Abhängigkeit der Anzahl der eingesetzten Fahrzeuge, sondern auch eine detaillierte Analyse des gesamten Systems. Eine Simulation eines FTS kann beispielsweise zur Optimierung des Layouts, zur Pufferdimensionierung, zur Bestimmung der Ressourcennutzung, sowie zum Vergleich unterschiedlicher Auftragsvergabe-strategien herangezogen werden.

Das Ziel des Beitrages besteht in der Entwicklung einer Methode zur Simulation von FTS, insbesondere von MRFS, um eine aufwandsarme Leistungsbestimmung zu ermöglichen. Betrachtet werden Systeme, in denen

- Fahrzeuge verfahren, welche sich auf einem vorgegebenen Wegenetz bewegen,
- einzelne Layoutabschnitte sowohl als uni- als auch als bidirektional befahrbar definiert werden können,
- die Fahrzeuge, wie in MRFS üblich, in einem rechtwinkligen Raster verfahren.

Die Methode soll eine Abbildung des Layouts in einer Simulationsumgebung umfassen. In dieser sollen Steuerungsstrategien für das Routing und Deadlock-Handling hinterlegt werden, die ein ablauffähiges Simulationsmodell sicherstellen und einen robusten und effizienten Betrieb des betrachteten Systems ermöglichen. Dafür wird auf den Ansatz des zeitfensterbasierten Routings zurückgegriffen.

In den folgenden Kapiteln wird detailliert auf die Modellierung und die Steuerungsstrategien eingegangen. Abschließend wird der Einsatz der entwickelten Methode an einem Anwendungsbeispiel gezeigt. In diesem wird der Durchsatzverlust ermittelt werden, welcher sich durch eine Bereitstellung in Sequenz gegenüber einer chaotischen Bereitstellung einstellt.

3 MODELLIERUNG VON FTS

In diesem Kapitel wird auf die Modellierung von FTS eingegangen, die es ermöglicht, das zeitfensterbasierte Routing anzuwenden. Zunächst wird die Modellierung des zugrundeliegenden Wegenetzes dargelegt. Anschließend wird die Modellierung der Bewegung der Fahrzeuge beschrieben.

3.1 MODELLIERUNG DES LAYOUTS

Simulationsumgebungen stellen zur Simulation von Produktions- und Logistiksystemen eine Reihe vordefinierter Bausteine zur Verfügung, aus denen individuelle Simulationsmodelle unterschiedlichen Abstraktionsgrades aufgebaut werden können. Diese Bausteine werden per Drag and Drop oder automatisiert in das Modell eingesetzt und über Kanten miteinander verbunden, um den Materialfluss abzubilden. Betrachtet man die Bausteine als Knoten, entsteht ein Graph, welcher das Materialflusssystem repräsentiert.

Der Grundgedanke der Überführung des Layouts eines FTS in eine Simulationsumgebung besteht darin, aus diesem Graphen eine formale Beschreibung des Layouts in Form von Adjazenzlisten zu generieren und in der Simulationsumgebung zu hinterlegen, um das zeitfensterbasierte Routing anzuwenden.

Das Wegenetz des FTS wird durch einen gerichteten Graphen $G = (V, E)$ beschrieben. Die Knoten V repräsentieren einzelne Layoutabschnitte, auf denen sich die Fahrzeuge bewegen. Die Kanten E haben keine Gewichtung und dienen lediglich der Information, welche Nachbarknoten von einem Knoten aus unmittelbar erreicht werden können.

Für die Unterteilung des Wegenetzes in einzelne Knoten, gilt:

- Jeder Knoten kann ein Fahrzeug vollständig aufnehmen.

- Auf jedem Knoten kann sich zu jedem Zeitpunkt nur ein einziges Fahrzeug befinden.

Die erste Bedingung verhindert, dass ein Fahrzeug mehr als zwei Knoten gleichzeitig belegt. Damit kann die Komplexität des zeitfensterbasierten Routings beschränkt werden. Die zweite Bedingung stellt sicher, dass keine Koordination mehrerer Fahrzeuge auf einem Knoten notwendig ist.

Die Fahrzeuge bewegen sich über die Knoten entlang der X-Achse und entlang der Y-Achse in einem rechtwinkligen Wegenetz. Die Ausrichtung A bezeichnet allgemein die Orientierung entlang dieser beiden Achsen.

A : Ausrichtung mit $A \in \{X, Y\}$.

Bezogen auf die Fahrzeuge gibt die Ausrichtung A an, entlang welcher der beiden Achsen ein Fahrzeug derzeit verfahren kann. Die Fahrzeuge können in jeder Ausrichtung vorwärts und rückwärts verfahren. Bezogen auf die Knoten gibt die Ausrichtung an, entlang welcher der beiden Achsen eine Bewegung über die Knoten möglich ist.

Über das gesamte Layout wird ein Raster gelegt und den einzelnen Knoten die jeweiligen Koordinaten im Raster zugewiesen. Anhand paarweiser Vergleiche der Koordinaten kann die Lage der Knoten zueinander bestimmt werden und für die Generierung der Adjazenzlisten herangezogen werden.

Einem Knoten $r_i \in V$ sind die folgenden Attribute zugeordnet:

$(X, Y)_{r_i}$: Koordinaten des Knoten in der Rasterung.

R_{r_i} : Quadrupel mit Booleschen Variablen, welches die Informationen enthält, in welcher Fahrtrichtung der Knoten verlassen werden kann (N, O, S, W), um die Fahrtrichtung entlang einzelner Segmente zu parametrisieren.

A_{r_i} : Menge der Ausrichtungen, welche angibt, entlang welcher Achsen ein Knoten befahren werden kann. Knoten können entlang der X-Achse oder entlang der Y-Achse befahren werden. Knoten, welche eine Kreuzung repräsentieren und einen Richtungswechsel zulassen, können sowohl entlang der X- als auch entlang der Y-Achse befahren werden. $A_{r_i} \in \{\{X\}, \{Y\}, \{X, Y\}\}$.

$L_{r_i}^A$: Ausdehnung des Knotens in Ausrichtung $A \in A_{r_i}$.

Abbildung 1 zeigt die Modellierung eines beispielhaften Layouts. An den Stirnseiten sind zwei Quergassen angeordnet, welche in beide Richtungen befahren werden können. Die Lagergassen hingegen können nur jeweils in eine Richtung befahren werden.

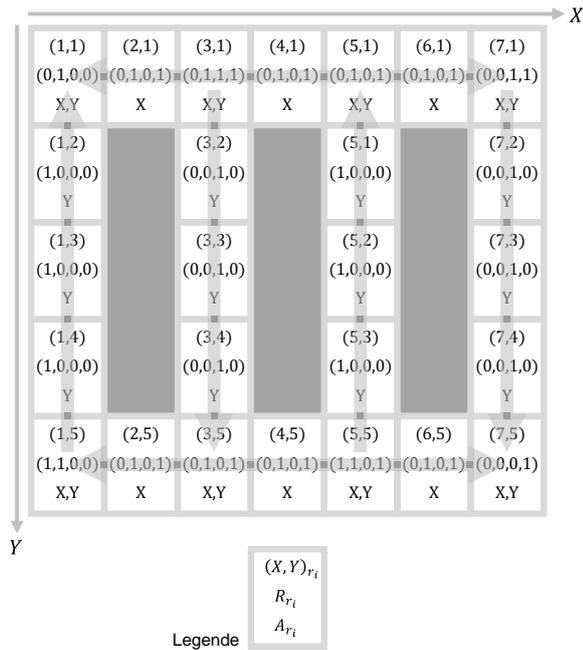


Abbildung 1. Modellierung eines Layouts mit unidirektionalen Lagergassen und bidirektionalen Quergassen durch entsprechende Belegung der Attribute der einzelnen Knoten.

Das Layout wird in der Simulationsumgebung durch zwei Adjazenzlisten formalisiert. In der ersten Adjazenzliste sind die Knoten unabhängig von ihrer Ausrichtung abgelegt. Über die Kanten im Simulationsmodell sowie die Information, in welcher Ausrichtung ein Knoten befahrbar ist und in welcher Fahrtrichtung er verlassen werden kann, ist es möglich, über den paarweisen Vergleich der Knotenkoordinaten die Liste der Nachfolger samt der Entfernung sowie der Fahrzeit von Knotenmittelpunkt zu Knotenmittelpunkt automatisiert zu generieren.

Da die minimalen Fahrzeiten zwischen zwei Knoten von der aktuellen Ausrichtung des Fahrzeuges auf dem Start-Knoten abhängen, wird eine zweite Adjazenzliste benötigt, welche die Ausrichtung der Knoten berücksichtigt. In dieser Adjazenzliste wird für jeden Knoten r_i und für jede Ausrichtung $A \in A_{r_i}$, in der der Knoten befahren werden kann, ein Eintrag r_i^A erzeugt. Die Nachfolger, welche in der Liste abgelegt werden, können entweder über eine reine Fahrbewegung oder über einen Wechsel der aktuellen Ausrichtung erreicht werden.

Tabelle 1. Adjazenzliste eines Kreuzungsknotens r_i .

Knoten mit Ausrichtung	Nachfolger	Fahrzeit
r_i^X	r_i^Y	t_{RW}
	r_j^X	$\frac{d(r_i, r_j)}{v_X}$
r_i^Y	r_i^X	t_{RW}
	r_h^Y	$\frac{d(r_i, r_h)}{v_Y}$

Mit:

t_{RW} : Richtungswechselzeit, die ein Fahrzeug benötigt, um die Ausrichtung zu wechseln.

$d(r_i, r_j)$: Entfernung der Mittelpunkte der Knoten r_i und r_j .

v_A : Geschwindigkeit in Ausrichtung A.

Anhand dieser Adjazenzliste wird mittels des Dijkstra-Algorithmus eine minimale Fahrzeitenmatrix zwischen allen Knoten und Ausrichtungen aufgestellt, welche für das zeitfensterbasierte Routing benötigt wird.

3.2 MODELLIERUNG DER BEWEGUNG DER FAHRZEUGE ÜBER DIE KNOTEN

Hinsichtlich der Art und Weise, wie die Fahrzeuge einen Wechsel der Ausrichtung durchführen, lassen sich zwei Kategorien unterscheiden. Entweder drehen sie sich um 90 Grad um die eigene Achse oder verfügen über einen Mechanismus zum seitlichen Verfahren, wie beispielsweise ein zusätzliches Seitenfahrwerk oder schwenkbare Räder (Abbildung 2).

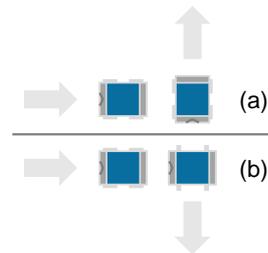


Abbildung 2. Um die Fahrtrichtung um 90 Grad zu wechseln führt das Fahrzeug eine Drehung um 90 Grad durch (a) oder behält die Orientierung bei, in dem es beispielsweise lediglich die Räder um 90 Grad schwenkt (b).

Je nach Kategorie, ändert sich die Länge des Fahrzeuges in Fahrtrichtung, wenn ein Wechsel der Ausrichtung vollzogen wird.

L_F^A : Länge des Fahrzeuges in Fahrtrichtung, wenn sich das Fahrzeug in Ausrichtung A bewegt.

Um die Bewegung eines Fahrzeuges über die Knoten zu beschreiben werden die folgenden Zeitstempel definiert:

T_{ES, r_i} : Start des Eintritts in den Knoten r_i .

T_{EE, r_i} : Ende des Eintritts in den Knoten r_i .

T_{An, r_i} : Ankunft, mittig positioniert, auf dem Knoten r_i .

T_{Ab, r_i} : Abfahrt, mittig positioniert, auf dem Knoten r_i .

T_{AS, r_i} : Start des Austritts aus dem Knoten r_i .

T_{AE, r_i} : Ende des Austritts aus dem Knoten r_i .

Sowie die folgenden Zeitdauern:

t_T^A : Transferzeit. Zeitdauer des Eintritts sowie des Austritts in bzw. aus einen Knoten in Ausrichtung A. Für die Dauer des Ein- und Austritts belegt ein FTF zwei benachbarte Knoten.

t_{P,r_i}^A : Positionierungszeit. Zeitdauer, die ein FTF benötigt, um in Ausrichtung A nach dem vollständigen Eintritt die Ankunftsposition (mittig positioniert) zu erreichen bzw. um von der Ankunftsposition den Rand des Knotens zu erreichen.

t_{W,r_i} : Wartezeit auf dem Knoten r_i (bis das nächste reservierte Zeitfenster auf dem nächsten Knoten beginnt).

Die Transfer- und Positionierungszeiten ergeben sich zu:

$$t_T^A = \frac{L_F^A}{v_A}$$

$$t_{P,r_i}^A = \frac{(L_{r_i}^A - L_F^A)/2}{v_A}$$

Darüber hinaus definieren wir für jeden Knoten r_i und jede Ausrichtung $A \in A_{r_i}$ eine minimale Räumungszeit.

t_{minR,r_i}^A : Minimale Räumungszeit. Minimale Zeitdauer, die ein FTF benötigt, um einen Knoten vollständig zu verlassen, wenn es sich in Ausrichtung A in Ankunftsposition auf dem Knoten r_i befindet.

Für Knoten, welche nur in einer Ausrichtung befahren werden können, gilt:

$$t_{minR,r_i}^A = t_{P,r_i}^A + t_T^A$$

In Abhängigkeit der Ausdehnung eines Knotens sowie der Richtungswechselzeit kann ein FTF einen Knoten früher räumen, wenn es einen Richtungswechsel auf dem Knoten durchführt und diesen anschließend verlässt. Daher gilt für Kreuzungsknoten:

$$t_{minR,r_i}^A = \min\{t_{P,r_i}^A + t_T^A, t_{RW} + t_{P,r_i}^{A'} + t_T^{A'}\}$$

Wobei A die Ausrichtung angibt, in welcher die Ankunft auf dem Kreuzungsknoten erfolgt und A' die Ausrichtung, in welcher der Knoten im Fall eines Richtungswechsels verlassen wird, mit:

$$A' = \begin{cases} X, & \text{für } A = Y \\ Y, & \text{für } A = X \end{cases}$$

Die Zusammenhänge der definierten Zeitstempel und Zeitdauern sind für eine lineare Bewegung der Abbildung 3 sowie für einen Richtungswechsel der Abbildung 4 zu entnehmen.

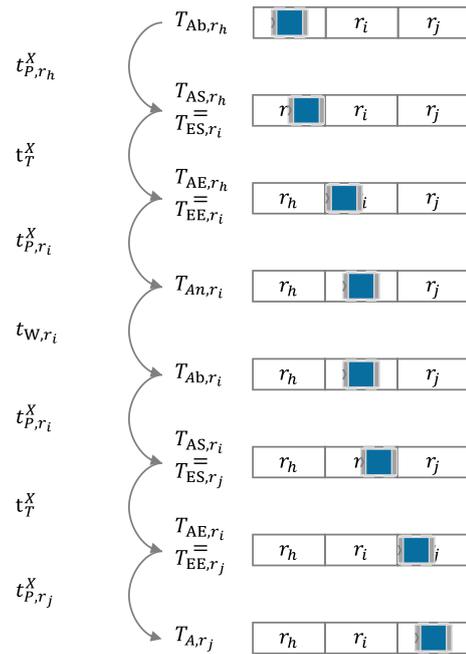


Abbildung 3. Bewegung eines Fahrzeugs über die Knoten ohne einen Richtungswechsel.

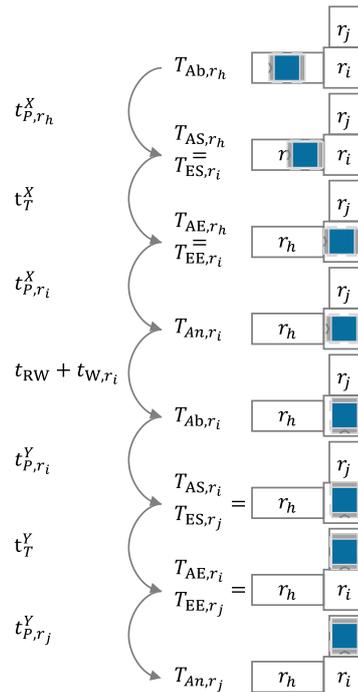


Abbildung 4. Bewegung eines Fahrzeugs über die Knoten mit einem Richtungswechsel auf einer Kreuzung.

Mit Hilfe der in diesem Kapitel beschriebene Modellierung lassen sich beliebige Layouts eines rechtwinkligen Wegenetzes manuell oder auch automatisiert erstellen sowie die Bewegung der FTF über das Wegenetz beschreiben. Um nach der Erstellung des Layouts ein ablauffähiges Simulationsmodell zu erhalten, werden in das Modell Steuerungsstrategien integriert, welche im folgenden Kapitel beschrieben werden.

4 STEUERUNGSSTRATEGIEN

In das Modell werden Strategien zur Routenfindung und zum Deadlock-Handling integriert. Für die Berechnung deadlockfreier Routen wird auf das Konzept des zeitfensterbasierten Routings nach ter Mors [Mor-2009] zurückgegriffen, welches die Berechnung schnellster, verklemmungsfreier Routen ermöglicht. Dieses Konzept wird dahingehend erweitert, dass

- Kreuzungen, auf denen ein Richtungswechsel um 90 Grad durchgeführt werden, in das Layout integriert werden,
- Ziele nicht als Knoten, sondern als Haltepositionen, welche auf einem oder zwei benachbarten Knoten liegen, definiert werden können,
- eine definierte Haltedauer an jedem Ziel eingeplant werden kann,
- eine Sequenzierung ermöglicht wird, welche zum Zeitpunkt des Routings geplant wird.

Die Bestimmung der Deadlock-freien Route wird in Kapitel 4.1 beschrieben, die Ausführung der berechneten Route in Kapitel 4.2.

4.1 ZEITFENSTERBASIERTES ROUTING

Für jeden Knoten im Layout existiert eine Liste mit reservierten und freien Zeitfenstern (Abbildung 5).

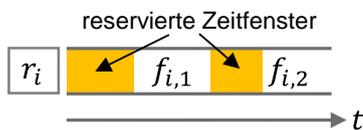


Abbildung 5. Freie Zeitfenster $f_{i,l}$ auf dem Knoten r_i .

Ein freies Zeitfenster auf einem Knoten ist durch das Ende des vorhergehenden reservierten Zeitfensters sowie den Beginn des nachfolgenden reservierten Zeitfensters bestimmt:

$$f_{i,l} = [\text{Start}_{i,l}, \text{Ende}_{i,l}]$$

$\text{Start}_{i,l}$: Beginn des l -ten freien Zeitfensters auf dem Knoten r_i .

$\text{Ende}_{i,l}$: Ende des l -ten freien Zeitfensters auf dem Knoten r_i .

Der zeitfensterbasierte Routingalgorithmus sucht nach einer Route für ein FTF zwischen einem Startort und einem Zielort von Knoten zu Knoten durch freie Zeitfenster, welche den Knoten zugeordnet sind (Abbildung 6). Das Ergebnis ist eine Liste mit Knoten, welche auf dem Weg vom Start zum Ziel befahren werden, sowie den Zeitintervallen, zu denen das FTF diese Knoten belegt.

In dem Beispiel in Abbildung 6 wird der Knoten r_4 auf dem Weg vom Start zum Ziel zweimal befahren, jedoch in zwei unterschiedlichen freien Zeitfenstern. Daher werden die zuvor eingeführten Zeitstempel zur Beschreibung der Bewegung eines FTFs über die Knoten im Folgenden nicht nur einem Knoten zugeordnet, sondern auch einem freien Zeitfenster, in welchem der Knoten von dem FTF befahren wird.

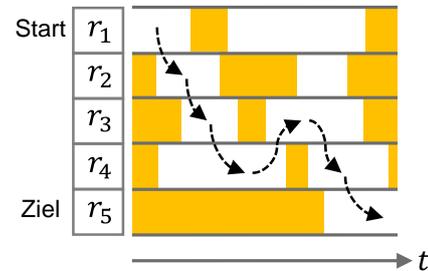


Abbildung 6. Route durch freie Zeitfenster von einem Startort auf dem Knoten r_1 zu einem Zielort auf dem Knoten r_5 .

In vielen Anwendungsfällen stehen bereits vor Beginn einer Fahrt mehrere Ziele fest, welche von einem FTF in einer bestimmten Reihenfolge angefahren werden müssen – wie beispielsweise bei der Ausführung eines Doppelspiels. Die Eingabe in den zeitfensterbasierten Routingalgorithmus umfasst daher neben dem Startknoten, der Ausrichtung des FTFs auf dem Startknoten und der Startzeit eine beliebige lange Liste mit Zielen Z_1, \dots, Z_z , welche

- in einer vorgegebenen Reihenfolge (z. B. Ziel Z_{n-1} vor Ziel Z_n) angefahren werden,
- derart angefahren werden, dass eine Aktion mit einer vorgegebenen Verweildauer an jedem dieser Ziele ausführbar ist (z. B. ein Lagerspiel),
- derart angefahren werden, dass an den Zielen eine optional vorgegebene Sequenz an Fahrzeugen eingehalten wird (z. B. FTF mit Sequenznummer 1 vor FTF mit Sequenznummer 2).

Das letzte Ziel Z_z in der übergebenen Liste der anzufahrenden Ziele bezeichnen wir auch als finales Ziel, am welchem die Fahrt des FTF endet. Die vorgegebene Verweildauer zur Durchführung einer Aktion an einem Ziel definieren wir als $t_{v,n}$:

$t_{v,n}$: Vorgegebene Verweildauer am n -ten Ziel in der Liste der übergebenen Ziele, bevor die Fahrt fortgesetzt werden kann.

Der zeitfensterbasierte Routingalgorithmus errechnet die Route zu einer gegebenen Startzeit vom aktuellen Standort von Ziel zu Ziel derart, dass alle gegebenen Kriterien erfüllt werden und die schnellste Route gefunden wird, welche das finale Ziel in der übergebenen Liste zum frühestmöglichen Zeitpunkt unter Berücksichtigung bestehender Reservierungen erreicht.

Ein Ziel entspricht nicht zwingend einem einzigen Knoten im Layout. Vielmehr ist ein Ziel durch eine Halteposition definiert, welche ein FTF einnimmt, um das Ziel zu erreichen. Die Halteposition zum Erreichen eines Ziels befindet sich entweder ausschließlich auf einem Knoten, oder aber auf zwei benachbarten Knoten. In dem Beispiel in Abbildung 7 muss sich ein FTF mittig vor einem Lagerplatz positionieren, um auf diesen zuzugreifen. Für den Zugriff auf den Lagerplatz 2 belegt ein FTF einen einzigen Knoten, während hingegen für den Zugriff auf den Lagerplatz 9 zwei Knoten durch das FTF belegt sind.

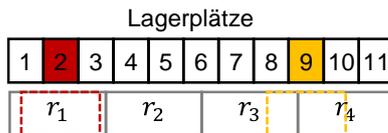


Abbildung 7. Haltepositionen für den Zugriff auf unterschiedliche Lagerplätze.

Wir fassen alle Knoten, welche beim Erreichen des Ziels Z_n durch das FTF belegt werden, in der Zielknotenmenge (ZKM) zusammen.

ZKM_n : Zielknotenmenge des Ziels Z_n , welche alle Knoten beinhaltet, die beim Erreichen des n -ten Ziels durch das FTF belegt werden.

Tabelle 2 enthält eine beispielhafte Liste mit drei Zielen, welche für die Ausführung eines Doppelspiels ausgehend vom aktuellen Standort des FTF angefahren werden sollen. Aufgeführt sind neben den Zielen die ZKM der jeweiligen Ziele, die Verweildauern sowie eine Sequenznummer für das finale Ziel. Für den Zugriff auf das erste Ziel – den Einlagerplatz – belegt das FTF nur einen einzigen Knoten, während hingegen die ZKM des zweiten Ziels – der Auslagerplatz – zwei Knoten enthält (vgl. Abbildung 7). Das finale Ziel der Fahrt stellt ein Pickplatz dar. Der auszulagernde Artikel ist mit einer Sequenznummer versehen und soll dieser entsprechend an dem Pickplatz bereitgestellt werden. Da nicht bekannt ist, wann das FTF den Pickplatz verlässt, wird zunächst von einer unendlichen Verweildauer ausgegangen.

Tabelle 2. Eingabe in den Routingalgorithmus: Liste mit drei Zielen für die Ausführung eines Doppelspiels.

n	Ziel Z_n	ZKM_n	$t_{v,n}$	Seq-Nr. S_n
1	Einlagerplatz 2	$\{r_1\}$	$t_{\text{Lagerspiel}}$	–
2	Auslagerplatz 9	$\{r_3, r_4\}$	$t_{\text{Lagerspiel}}$	–
3	Pickplatz A	$\{r_p\}$	∞	17

Für das Routing der Fahrzeuge treffen wir die folgenden Annahmen:

- Die ZKM des finalen Ziels Z_z , besteht immer aus genau einem Knoten. Dieser Knoten repräsentiert beispielsweise einen Übergabeplatz oder einen Pufferplatz.
- Die Ausrichtung, in welcher ein Ziel zu erreichen ist, ist nicht vorgegeben. Stimmen die Halteposition zum Erreichen eines Ziels und die Ankunftsposition auf einem Kreuzungsknoten überein, kann ein FTF ein Ziel in beiden Ausrichtungen erreichen.
- Befindet sich die Halteposition eines Ziels nur auf einem Knoten und stimmt diese nicht mit der Ankunftsposition überein, so muss das FTF bevor es nach der Verweildauer am Ziel seine Fahrt fortsetzt, zunächst die Ankunftsposition auf dem Knoten einnehmen.

Der Kern des Algorithmus besteht in der Untersuchung, welche weiteren freien Zeitfenster aus einem freien Zeitfenster heraus erreichbar sind (Abbildung 8).

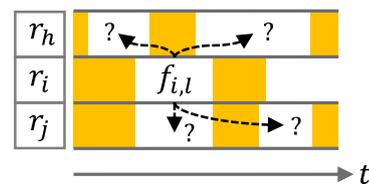


Abbildung 8. Welche freien Zeitfenster auf allen Nachbarknoten des Knotens r_i sind aus dem aktuellen Zeitfenster $f_{i,l}$ heraus erreichbar?

Ist ein freies Zeitfenster erreichbar, kann der Knoten, dem dieses Zeitfenster zugeordnet ist, in eben diesem Zeitfenster von dem FTF, ausgehend von der Startposition und der gegebenen Startzeit, angefahren werden; das freie Zeitfenster ist für das FTF erreichbar.

Ter Mors erstellt zu Beginn des Routings den sogenannten Zeitfenstergraphen. Dabei handelt es sich um einen Graphen, in welchem die Knoten freie Zeitfenster und die Kanten zwischen den Knoten die Erreichbarkeit der freien Zeitfenster untereinander repräsentieren. Die Suche nach der schnellsten Route entspricht dann der Suche des kürzesten Pfades in diesem Graphen. Dieses Konzept lässt sich bei der Berücksichtigung von Richtungswechseln auf Kreuzungen nicht anwenden. Die Erreichbarkeit eines freien Zeitfensters hängt auch von der aktuellen Ausrichtung ab und lässt sich somit nicht statisch in einem Graphen hinterlegen. Je nach aktueller Ausrichtung, in der sich ein Fahrzeug auf einem Knoten befindet, bedarf es vor dem Eintritt in den Nachbarknoten eines Wechsels der Ausrichtung, welcher zusätzlich Zeit erfordert. Anstatt den Zeitfenstergraphen für jeden Routingprozess vollständig zu generieren, prüfen wir schrittweise, welche freien Zeitfenster mit der aktuellen Ausrichtung aus dem aktuellen Zeitfenster heraus erreichbar sind. Der Graph wird also nur lokal

berechnet, was insbesondere bei großen Layouts auch einen Rechenvorteil mit sich bringt.

Der Algorithmus unterscheidet drei unterschiedliche Mengen an Zeitfenstern: Die Menge der

- *freien Zeitfenster*, welche alle freien Zeitfenster auf allen Knoten enthält,
- *erreichbaren Zeitfenster*, welche alle Zeitfenster enthält, die sich im Verlauf des Routings als erreichbar herausgestellt haben und für die Untersuchung in der nächsten Iteration ausgewählt werden können,
- *untersuchten Zeitfenster*, für die die Untersuchung der Erreichbarkeit aller freien Zeitfenster auf allen Nachbarknoten bereits erfolgt ist.

Ist ein freies Zeitfenster $f_{i,l}$ erreichbar und wird dieses in die Menge der erreichbaren Zeitfenster aufgenommen, werden folgende Merkmale gespeichert:

- $ID_{f_{i,l}}$: Eindeutige ID, welche zur Unterscheidung notwendig ist, da freie Zeitfenster während eines Routings mehrmals erreicht und in die Menge der erreichbaren Zeitfenster aufgenommen werden können.
- $T_{An,f_{i,l}}$: Ankunftszeitpunkt in dem freien Zeitfenster $f_{i,l}$ auf dem Knoten r_i .
- $A_{f_{i,l}}$: Ausrichtung, in welcher die Ankunft in dem freien Zeitfenster $f_{i,l}$ auf dem Knoten r_i erfolgt.
- $D_{f_{i,l}}$: Kürzeste Gesamtfahrzeit vom Startknoten zum finalen Ziel Z_n in der übergebenen Liste der Ziele durch das aufgenommene Zeitfenster. Die Fahrzeit setzt sich aus der Fahrzeit bis zur Ankunft $T_{An,f_{i,l}}$ im freien Zeitfenster $f_{i,l}$ auf dem Knoten r_i und den minimalen Fahrzeiten zu den restlichen Zielen zuzüglich der Verweildauern an diesen Zielen zusammen.
- $n_{f_{i,l}}$: Zähler, welcher angibt, zu welchem Ziel aus der Liste der übergebenen Ziele von dem Zeitfenster $f_{i,l}$ aus geroutet wird. Somit ist bekannt, welche Ziele $Z_1, \dots, Z_{n_{f_{i,l}}-1}$ vor der Ankunft im freien Zeitfensters $f_{i,l}$ bereits erreicht worden sind. Mit $n_{f_{i,l}} = 1, \dots, z + 1$.
- $B_{f_{i,l}}$: Zeiger (Backpointer), welcher auf das freie Zeitfenster zeigt, aus welchem das neu aufgenommene freie Zeitfenster erreicht wird, um nach Abschluss des Routings die Route konstruieren zu können.

Der Algorithmus besteht im Wesentlichen aus

0. einer Initialisierung,
1. der Auswahl des freien Zeitfensters für die nächste Iteration,
2. der Prüfung, ob das gewählte Zeitfenster alle Bedingungen erfüllt, falls ein Ziel erreicht wird,
3. der Prüfung der Erreichbarkeit aller freien Zeitfenster auf allen Nachbarknoten aus dem gewählten Zeitfenster heraus,
4. einer Aufbereitung der Route, nach dem das finale Ziel erreicht worden ist.

Die Abbildung 9 veranschaulicht das Vorgehen sowie den Zusammenhang der unterschiedlichen Mengen an Zeitfenstern.

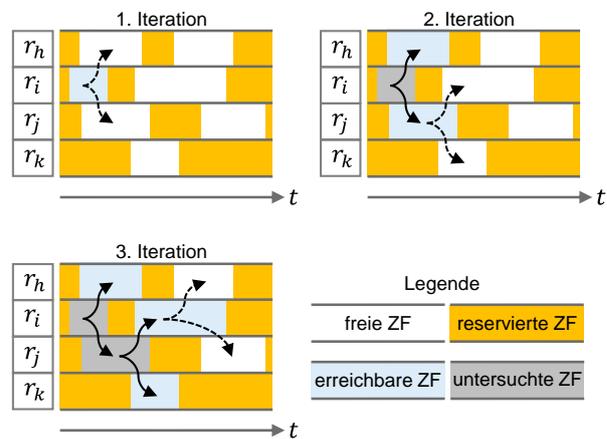


Abbildung 9. Drei Iterationen des zeitfensterbasierten Routingalgorithmus.

Ausgehend von einem erreichbaren Zeitfenster wird geprüft, welche weiteren freien Zeitfenster auf den Nachbarknoten erreichbar sind. Diese erreichbaren Zeitfenster werden der Menge der erreichbaren Zeitfenster hinzugefügt und können für die Untersuchung in einer nachfolgenden Iteration ausgewählt werden. Ist die Untersuchung der Erreichbarkeit aller freien Zeitfenster auf allen Nachbarknoten erfolgt, wird das Zeitfenster der Menge der untersuchten Zeitfenster hinzugefügt.

Der Ablauf des Algorithmus ist in der Abbildung 10 dargestellt und wird im Folgenden detailliert erläutert.

Zu Beginn des Routings erfolgt eine Initialisierung (Abbildung 10, a), in der das aktuelle Zeitfenster, aus welchem heraus das Routing gestartet wird, der Menge der erreichbaren Zeitfenster hinzugefügt wird. Dabei wird der Ankunftszeitpunkt in diesem Zeitfenster mit der frühestmöglichen Abfahrtszeit belegt.

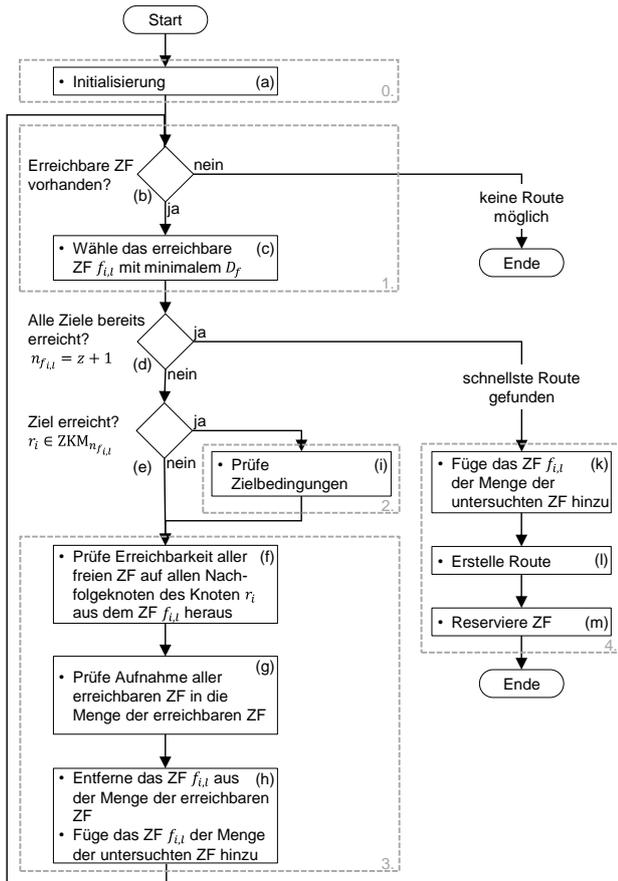


Abbildung 10. Zeitfensterbasiertes Routing

In jeder Iteration wird zunächst geprüft, ob noch Zeitfenster in der Menge der erreichbaren Zeitfenster vorhanden sind (Abbildung 10, b). Ist diese Menge leer, so kann keine Route gefunden werden. Dieser Fall kann nur dann auftreten, wenn FTF einen Knoten für eine unbestimmte Zeit reservieren können. Dieses Vorgehen kann beispielsweise angewendet werden, wenn ein untätiges FTF gepuffert werden soll und nicht bekannt ist, wann es den Pufferplatz wieder verlässt. Soll nun ein weiteres FTF auf diesen Pufferplatz geroutet werden, kann der Zielknoten nicht erreicht werden und der Algorithmus gibt zurück, dass keine Route möglich ist.

Ist in der Menge der erreichbaren Zeitfenster noch mindestens ein weiteres Zeitfenster vorhanden, so wird aus der Menge der erreichbaren Zeitfenster dasjenige Zeitfenster $f_{i,l}$ gewählt, welches derzeit die kürzeste Gesamtfahrzeit $D_{f_{i,l}}$ zum finalen Ziel ermöglicht (Abbildung 10, c).

Wurden vor der Ankunft im aktuellen Zeitfenster $f_{i,l}$ noch nicht alle übergebenen Ziele Z_1, \dots, Z_n erreicht (Abbildung 10, d), gilt also

$$n_{f_{i,l}} \leq z$$

und ist das gewählte Zeitfenster nicht einem Knoten der ZKM des aktuellen Ziels $Z_{f_{i,l}}$ zugeordnet (Abbildung 10, e), d. h.

$$r_i \notin ZKM_{n_{f_{i,l}}}$$

so erfolgt die Untersuchung, welche freien Zeitfenster auf allen Knoten in der Adjazenzliste des aktuellen Knoten r_i aus dem aktuellen Zeitfenster $f_{i,l}$ mit der gegebenen Ankunftszeit $T_{An,f_{i,l}}$ und Ausrichtung $A_{f_{i,l}}$ erreicht werden können (Abbildung 10, f). Um die Erreichbarkeit eines freien Zeitfensters $f_{j,m}$ aus dem aktuellen Zeitfenster $f_{i,l}$ zu prüfen, wird wie in der Abbildung 11 vorgegangen.

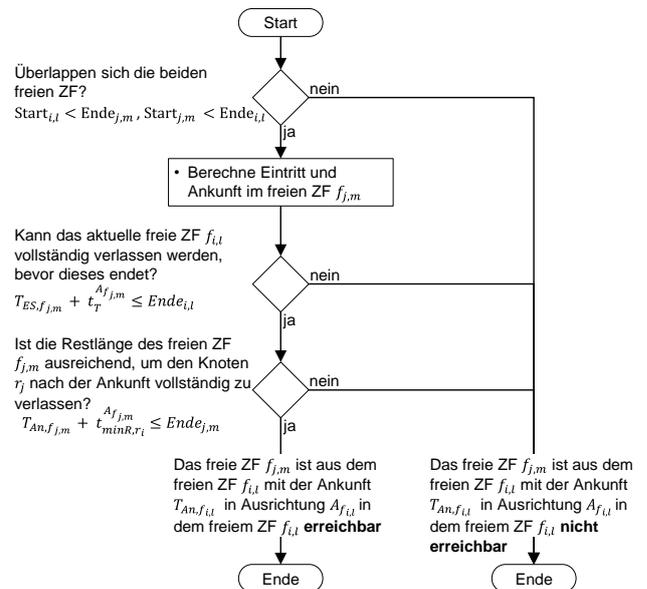


Abbildung 11. Ist das freie Zeitfenster $f_{j,m}$ aus dem gewählten freien Zeitfenster $f_{i,l}$ erreichbar?

Überlappen sich die beiden freien Zeitfenster $f_{i,l}$ und $f_{j,m}$, gilt also

$$Start_{j,m} < Ende_{i,l} \text{ und } Start_{i,l} < Ende_{j,m}$$

kann der Eintritt in das freie Zeitfenster $f_{j,m}$ berechnet werden:

$$T_{ES,f_{j,m}} = \begin{cases} \max \{ T_{An,f_{i,l}} + t_{p,r_i}^{A_{f_{i,l}}}, Start_{j,m} \} & , \text{ ohne RW} \\ \max \{ T_{An,f_{i,l}} + t_{RW} + t_{p,r_i}^{A_{f_{i,l}}}, Start_{j,m} \} & , \text{ mit RW} \end{cases}$$

Mit:

$$A'_{f_{i,l}} = \begin{cases} X, \text{ für } A_{f_{i,l}} = Y \\ Y, \text{ für } A_{f_{i,l}} = X \end{cases}$$

Der Eintritt in das Zielzeitfenster kann frühestens erfolgen, wenn das FTF den aktuellen Knoten passiert hat. Außerdem kann der Eintritt in das Zielzeitfenster frühestens erfolgen, wenn das freie Zeitfenster begonnen hat. Gegebenenfalls muss ein Richtungswechsel vollzogen werden, um in das freie Zeitfenster einzutreten. Ein Richtungswechsel ist nur auf einer Kreuzung möglich und ist genau dann notwendig, wenn der nachfolgende Knoten r_j nicht über die aktuelle Ausrichtung $A_{f_{i,l}}$ im freien Zeitfensters $f_{i,l}$ auf dem aktuellen Knoten r_i zu erreichen ist. Falls ein Richtungswechsel notwendig ist, ändert sich die Ausrichtung der Ankunft im freien Zeitfenster $f_{j,m}$ gegenüber der Ausrichtung im aktuellen Zeitfenster $f_{i,l}$. Anderenfalls wird diese übernommen:

$$A_{f_{j,m}} = \begin{cases} A_{f_{i,l}} & , \text{ ohne RW} \\ A'_{f_{i,l}} & , \text{ mit RW} \end{cases}$$

Die Ankunft im freien Zeitfenster $f_{j,m}$ ergibt sich durch die Addition der Transferzeit und der Positionierungszeit auf dem Knoten r_j :

$$T_{An,f_{j,m}} = T_{ES,f_{j,m}} + t_T^{A_{f_{j,m}}} + t_{P,r_j}^{A_{f_{j,m}}}$$

Nach der Berechnung der Ankunft ist abschließend zu prüfen, ob das aktuelle freie Zeitfenster $f_{i,l}$ vollständig verlassen wird, bevor dieses endet:

$$T_{ES,f_{j,m}} + t_T^{A_{f_{j,m}}} \leq \text{Ende}_{i,l}$$

und ob die Restlänge des freien Zeitfensters $f_{j,m}$ ausreicht, um den Knoten r_j nach dem Erreichen der Ankunftsposition vollständig zu verlassen:

$$T_{An,f_{j,m}} + t_{\min R,r_i}^{A_{f_{j,m}}} \leq \text{Ende}_{j,m}$$

Die Abbildungen 12 und 13 veranschaulichen die Bedingungen für die Erreichbarkeit eines freien Zeitfensters $f_{j,m}$ aus dem aktuellen Zeitfenster $f_{i,l}$ heraus, ohne dass dabei ein Richtungswechsel notwendig ist.

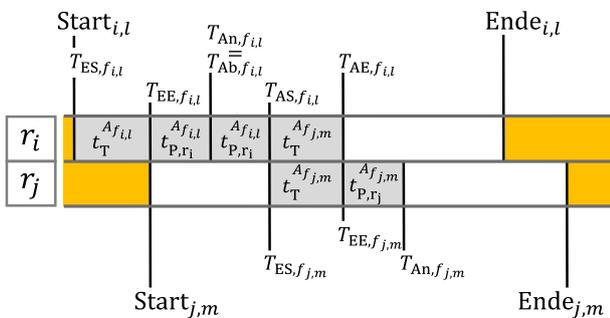


Abbildung 12. Das freie Zeitfenster $f_{j,m}$ auf dem Knoten r_j ist erreichbar, der Knoten r_i kann direkt durchfahren werden.

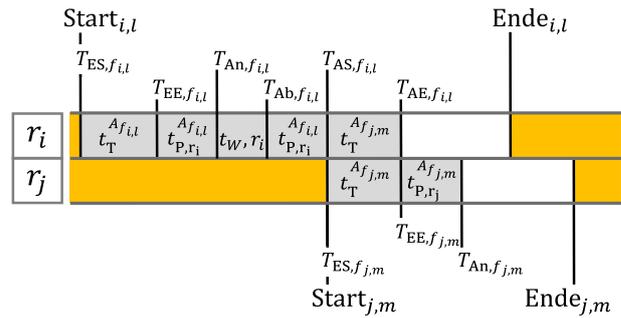


Abbildung 13. Das freie Zeitfenster $f_{j,m}$ auf dem Knoten r_j ist erreichbar, jedoch muss die Zeitdauer t_{W,r_i} auf dem Knoten r_i gewartet werden, bis das freie Zeitfenster $f_{j,m}$ beginnt und die Einfahrt in den Knoten gestartet werden kann.

Sind alle Bedingungen erfüllt, wird geprüft, ob das erreichbare Zeitfenster $f_{j,m}$ in die Menge der erreichbaren Zeitfenster aufgenommen wird (Abbildung 9, g). Ein erreichbares Zeitfenster wird genau dann zur Untersuchung in die Menge der erreichbaren Zeitfenster aufgenommen, wenn:

- es mit gleichem Zähler $n_{f_{j,m}}$ weder in der Menge der erreichbaren Zeitfenster, noch in der Menge der untersuchten Zeitfenster vorhanden ist,
- es mit gleichem Zähler $n_{f_{j,m}}$ bereits in der Menge der erreichbaren Zeitfenster vorhanden ist, aber eine kürzere mögliche Gesamtfahrzeit $D_{f_{j,m}}$ zum Ziel aufweist, als das bereits vorhandene Zeitfenster.

Abschließend ist die Information zu übergeben, zu welchem Ziel aus dem aufgenommenen Zeitfenster $f_{j,m}$ geroutet wird:

$$n_{f_{j,m}} = n_{f_{i,l}}$$

sowie der Zeiger des erreichten Zeitfensters $f_{j,m}$ auf das aktuelle Zeitfenster $f_{i,l}$ zu setzen, um nach Erreichen des finalen Ziels die Route zu konstruieren:

$$B_{f_{j,m}} = f_{i,l}$$

Wurden auf allen Nachfolgeknoten alle freien Zeitfenster untersucht, wird das aktuelle Zeitfenster $f_{i,l}$ der Menge der erreichbaren Zeitfenster entnommen und der Menge der untersuchten Zeitfenster hinzugefügt (Abbildung 10, h).

Wird hingegen für die nächste Iteration des Algorithmus ein Zeitfenster $f_{i,l}$ gewählt, welches einem Knoten der ZKM des aktuellen Ziels zugeordnet ist (Abbildung 10, e), gilt also

$$r_i \in \text{ZKM}_{n_{f_{i,l}}}$$

wird ein Ziel $Z_{n_{f_{i,l}}}$ im Routing erreicht und die Zielbedingungen sind zu prüfen (Abbildung 10, i).

Für die Prüfung der Zielbedingungen ist eine Fallunterscheidung zu treffen. Im ersten Fall enthält die ZKM des Ziels $Z_{n_{f_{i,l}}}$ nur einen Knoten. Dann wird die Ankunftszeit errechnet, zu welcher das FTF die Ankunftsposition einnimmt, nachdem die Verweildauer am Ziel verstrichen ist. Stimmen die Halteposition des Ziels $Z_{n_{f_{i,l}}}$ und die Ankunftsposition überein, ergibt sich die Ankunftszeit zu

$$T_{An,f'_{i,l}} = T_{An,f_{i,l}} + t_{v,n_{f_{i,l}}}$$

Anderenfalls muss eine Fahrbewegung berücksichtigt werden, um nach dem Erreichen der Halteposition die Ankunftsposition zu erreichen, falls diese vorher passiert wurde. Nun wird überprüft, ob der aktuelle Knoten r_i vollständig verlassen werden kann, bevor das Zeitfenster $f_{i,l}$ endet:

$$T_{An,f'_{i,l}} + t_{minR,r_i}^{A_{f_{i,l}}} \leq Ende_{i,l}$$

Darüber hinaus wird auch geprüft, ob eine gegebene Sequenzanforderung verletzt wird. Wird ein Ziel erreicht und hat ein FTF mit einer vorhergehenden Sequenznummer ein späteres Zeitfenster auf diesem Knoten reserviert, erreicht das aktuelle FTF dieses Ziel zu früh und die Sequenz ist verletzt. In diesem Fall kann das Ziel in dem freien Zeitfenster $f_{i,l}$ nicht erreicht werden. Erst ein freies Zeitfenster auf dem Knoten r_i , welches hinter dem reservierten Zeitfenster für das FTF mit der vorhergehenden Sequenznummer liegt, ermöglicht das Erreichen des Ziels (Abbildung 14).

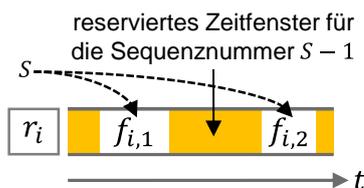


Abbildung 14. Wird die Sequenznummer S auf den Zielknoten r_i geroutet, dann verletzt das freie Zeitfenster $f_{i,1}$ die Sequenzanforderung, während hingegen das freie Zeitfenster $f_{i,2}$ die geforderte Sequenz einhält.

Sind alle Zielbedingungen erfüllt, wird ein Duplikat $f'_{i,l}$ des Zeitfensters $f_{i,l}$ in der Menge der erreichbaren Zeitfenster erstellt, wobei eine eindeutige ID zugewiesen wird und der aktualisierte Ankunftszeitpunkt $T_{An,f'_{i,l}}$ gespeichert wird. Da ein Ziel erreicht wurde, wird von dem neu aufgenommenen Zeitfenster in Richtung des nächsten Ziels $Z_{n_{f_{i,l}+1}}$ in der Liste der übergebenen Ziele geroutet:

$$n_{f'_{i,l}} = n_{f_{i,l}} + 1$$

Wird mindestens eine der Zielbedingungen verletzt, kann in dem aktuellen Zeitfenster das Ziel nicht erreicht werden. Der Routingalgorithmus läuft weiter, bis ein späteres Zeitfenster auf dem Zielknoten gefunden wird, welches alle Zielbedingungen erfüllt (siehe Abbildung 15)

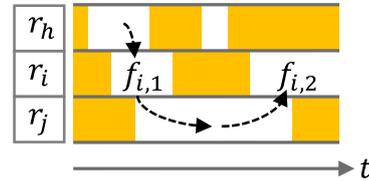


Abbildung 15. Das Zeitfenster $f_{i,1}$ auf dem Zielknoten r_i verletzt die Zielbedingungen, wird aber dennoch durchfahren, um das spätere Zeitfenster $f_{i,2}$ auf dem Zielknoten zu erreichen, welches alle Zielbedingungen erfüllt.

Werden für das Erreichen des Ziels hingegen zwei Knoten benötigt, wird die Erreichbarkeit aller freien Zeitfenster auf dem weiteren Knoten r_j der ZKM gemäß dem allgemeinen Vorgehen (Abbildung 10, f) geprüft, wobei nun die vorgegebene Verweildauer $t_{v,n_{f_{i,l}}}$ am Ziel $Z_{f_{i,l}}$ zu berücksichtigen ist. Das gilt für die Berechnung der Ankunft in einem freien Zeitfenster $f_{j,m}$:

$$T_{An,f_{j,m}} = T_{ES,f_{j,m}} + t_T^{A_{f_{j,m}}} + t_{v,n_{f_{i,l}}} + t_{p,r_j}^{A_{f_{j,m}}}$$

sowie für die Überprüfung, ob das aktuelle freie Zeitfenster vollständig verlassen wird, bevor dieses endet:

$$T_{ES,f_{j,m}} + t_T^{A_{f_{j,m}}} + t_{v,n_{f_{i,l}}} \leq Ende_{i,l}$$

Wird eine vorliegende Sequenzbedingung durch das Erreichen des Ziels in den Zeitfenstern $f_{i,l}$ auf dem Knoten r_i und $f_{j,m}$ auf dem Knoten r_j nicht verletzt, wird das erreichbare Zeitfenster $f_{j,m}$ auf dem weiteren Knoten r_j der ZKM ebenfalls in die Menge der erreichbaren Zeitfenster aufgenommen, wobei der Zähler $n_{f_{j,m}}$ aktualisiert wird:

$$n_{f_{j,m}} = n_{f_{i,l}} + 1$$

Wird für die nächste Iteration des Algorithmus ein Zeitfenster $f_{i,l}$ gewählt, für welches gilt:

$$n_{f_{i,l}} = z + 1$$

so wurden alle übergebenen Ziele erreicht und die schnellste Route ist gefunden (Abbildung 10, d). Das Zeitfenster ist nun der Menge der untersuchten Zeitfenster hinzuzufügen (Abbildung 10, k) und über die Rückverfolgung der Zeiger B kann die Route erstellt werden (Abbildung 10, l). Dafür müssen die Abfahrtszeiten von den einzelnen Knoten errechnet werden, diese ergeben sich für zwei aufeinanderfolgende Knoten r_i und r_j der Route zu:

$$T_{Ab,f_{i,l}} = T_{An,f_{j,m}} - t_{P,r_j}^{Af_{j,m}} - t_T^{Af_{j,m}} - t_{P,r_i}^{Af_{j,m}}$$

Die Wartezeiten auf den einzelnen Knoten ergeben sich aus den Differenzen der An- und Abfahrtszeiten. Die benötigten Zeitfenster τ_i auf allen Knoten τ_i

$$\tau_i = [T_{ES,f_{i,l}}, T_{AE,f_{i,l}}]$$

der errechneten Route werden reserviert (Abbildung 10, m). Die Ausführung einer errechneten Route wird im folgenden Abschnitt beschrieben.

4.2 AUSFÜHRUNG DER ROUTE

Da zur Berechnung der Route weder Beschleunigung noch Verzögerungen der Fahrzeuge berücksichtigt werden und es zu unvorhergesehen Verspätungen der Fahrzeuge kommen kann, wird die Ausführung der Route von den errechneten Zeitfenstern entkoppelt. Maza und Castagna [Maz-2005] haben gezeigt, dass die Abwesenheit von Deadlocks garantiert werden kann, solange die Reihenfolge auf den Knoten eingehalten wird, welche sich durch die Reservierung der Zeitfenster ergibt.

Das Ergebnis des Routings ist eine Liste, welche die Knoten, sowie die Zeitintervalle, zu welchen diese Knoten von dem FTF belegt sind, enthält. Es ist jedoch nicht nur bekannt, welche Knoten ein FTF für seine Fahrt zu welchen Zeitintervallen reserviert hat, sondern auch, welche FTF einen bestimmten Knoten reserviert haben, insbesondere die Reihenfolge, in welcher die Zeitintervalle auf diesem Knoten liegen. Diese Eigenschaft haben wir uns bereits für die Sequenzierung zu Nutze gemacht und nutzen sie nun auch, um die FTF auch im Fall von Verspätungen verklemmungsfrei an ihr Ziel zu bringen.

Um die Reihenfolge, in der die FTF ihre Zeitfenster auf einem Knoten reserviert haben, bei der Fahrt einzuhalten, führen wir das Konzept der Beanspruchung von Knoten ein:

- Ein FTF darf nur dann einen Knoten befahren, wenn es diesen zuvor beansprucht hat.
- Ein FTF darf genau dann einen Knoten beanspruchen, wenn es das erste Zeitfenster auf diesem Knoten reserviert hat.

Vor Beginn einer jeden Fahrt, wird nun die maximal mögliche Anzahl an Knoten beansprucht, bis ein nächstes Ziel erreicht wird. Die Menge der beanspruchten Knoten wird als Segment bezeichnet. Das Vorgehen zum Beanspruchen von Knoten des nächsten Segmentes ist in der Abbildung 16 dargestellt.

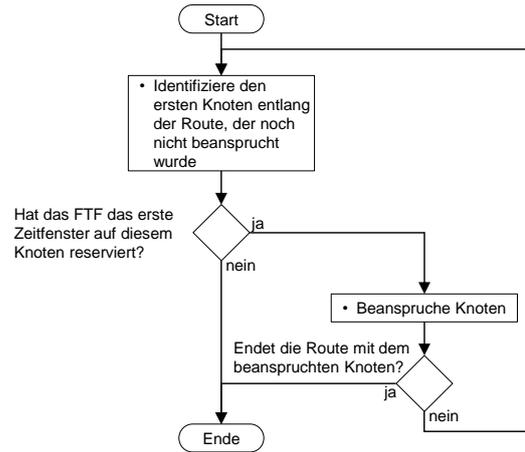


Abbildung 16. Segment prüfen: Vorgehen, um die nächsten Knoten für die Fahrt zu beanspruchen.

An dieser Stelle ist anzumerken, dass nach dem erfolgreichen Routing durchaus früheste Zeitfenster auf Knoten reserviert werden, welche bereits beansprucht worden sind. In diesem Fall muss die Beanspruchung des betreffenden Knotens rückgängig gemacht werden, da nun das neu geroutete FTF diesen Knoten als erstes passieren muss.

Konnte mindestens ein weiterer Knoten beansprucht werden, kann die Fahrt über die Knoten beginnen. Die Ausführung der Fahrt erfolgt gemäß dem Ablauf in Abbildung 17.

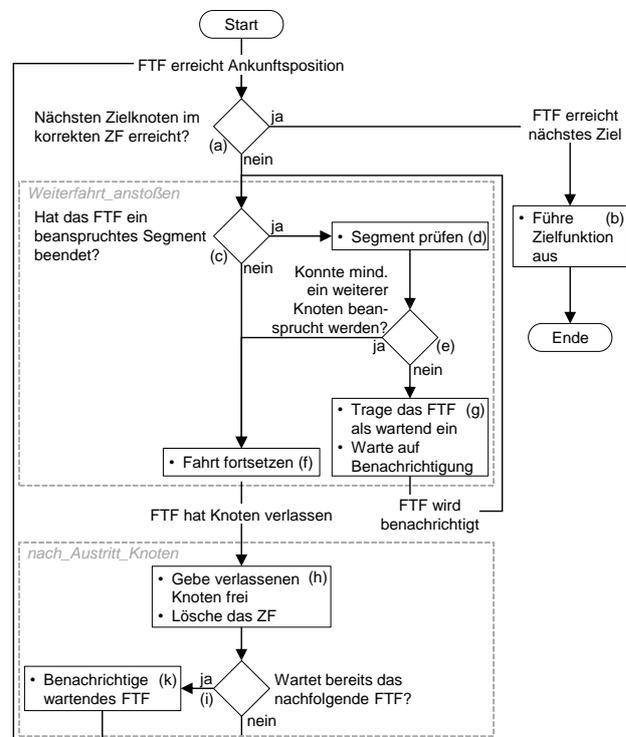


Abbildung 17. Ausführung einer berechneten Route.

Wenn ein FTF die Ankunftsposition eines Knoten erreicht, wird geprüft, ob es sein nächstes Ziel im vorgesehenen Zeitfenster erreicht (Abbildung 17, a). In diesem Fall kann die entsprechende Zielfunktion ausgeführt werden, welche z. B. eine Ein- oder Auslagerung darstellen kann (Abbildung 17, b).

Anderenfalls wird geprüft, ob das FTF ein beanspruchtes Segment beendet hat (Abbildung 17, c). In diesem Fall wird versucht, weitere Knoten entlang der Route zu beanspruchen (Abbildung 17, d). Konnte mindestens ein weiterer Knoten beansprucht werden (Abbildung 17, e), wird die Fahrt fortgesetzt (Abbildung 17, f). Anderenfalls trägt sich das FTF als wartendes FTF ein und wartet auf die Freigabe des vorhergehenden FTF (Abbildung 17, g).

Sobald ein FTF einen Knoten vollständig verlassen hat, wird das entsprechende Zeitfenster auf dem Knoten gelöscht und der Knoten wieder freigegeben (Abbildung 17, h). Abschließend wird geprüft, ob sich das nachfolgende FTF bereits als wartendes FTF eingetragen hat und auf dem Nachbarknoten wartet (Abbildung 17, i). In diesem Fall wird die Weiterfahrt des wartenden FTF angestoßen, welches nun mindestens einen weiteren Knoten beanspruchen und seine Fahrt fortsetzen kann (Abbildung 17, k).

5 ANWENDUNGSBEISPIEL SEQUENZIERUNG MIT SHUTTLE-SYSTEMEN

In diesem Kapitel beschreiben wir ein Anwendungsbeispiel der beschriebenen Methode zur Simulation eines FTS. Wir betrachten ein Shuttle-System mit gassen- und ebenen-wechselnden Fahrzeugen und untersuchen die Durchsatzverluste, welche durch eine Auslagerung in Sequenz entstehen. Dafür nutzen wir ein Simulationsmodell, welches in der Simulationsumgebung *Plant Simulation* erstellt wurde. Zunächst definieren wir das betrachtete System. Anschließend wird die Simulation vorgestellt.

5.1 SYSTEMBESCHREIBUNG

Shuttle-Systeme lassen sich beispielsweise anhand der Bewegungsachsen, entlang welcher sich die Shuttle-Fahrzeuge bewegen können, klassifizieren. Während Systeme mit gassen- und ebenen-gebundenen Shuttle-Fahrzeugen den höchsten Durchsatz ermöglichen, bieten Systeme mit Shuttle-Fahrzeugen, welche sowohl die Lagerebene mittels eines Shuttle-Liftes als auch die Lagergasse über Quergasse wechseln können, eine erhöhte Flexibilität. Der erzielbare Durchsatz kann durch die Anzahl der eingesetzten Shuttle-Fahrzeuge skaliert werden.

Jede Ebene eines Shuttle-Systems mit gassen- und ebenen-wechselnden Shuttle-Fahrzeugen kann als ein MRFS aufgefasst werden, welches über einen Lift mit anderen Ebenen sowie der Ein- und Auslagerebene (E/A-Ebene) verbunden ist (siehe Abbildung 18).

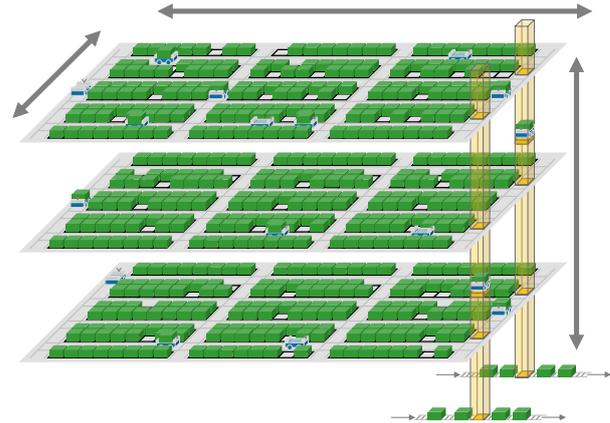


Abbildung 18. Shuttle-Lager mit gassen- und ebenen-wechselnden Fahrzeugen.

Jedes Shuttle-Fahrzeug kann jeden Lagerplatz im gesamten System erreichen. Trifft ein Shuttle-Fahrzeug auf einer Lagerebene ein, erfolgt das Routing auf dieser Lagerebene ausgehend vom Lift zum Einlagerplatz, zum Auslagerplatz und anschließend wieder zu dem Lift, über den die Ebene verlassen wird.

Die Modellierung einer Lagerebene erfolgt gemäß Kapitel 3.1, wobei unterschiedliche Klassen an Knoten benutzt werden, um zwischen Pufferplätzen, Kreuzungen, Quergassen und Lagergassen zu unterscheiden. Die exakte Modellierung kann [Lie-2017] entnommen werden. Wir nehmen an, dass jeder Bereitstellort durch genau einen Lift versorgt wird. Die Sequenzierung erfolgt zweistufig. Auf jeder Lagerebene erfolgt eine ebenen-bezogene Sequenzierung, bei der die Lagereinheiten nach aufsteigender Sequenznummer an den Liften bereitgestellt werden. Die finale Sequenz wird durch die Lifte hergestellt, welche die Shuttle-Fahrzeuge lückenlos nach aufsteigender Sequenznummer von den Lagerebenen abziehen (siehe Abbildung 19).

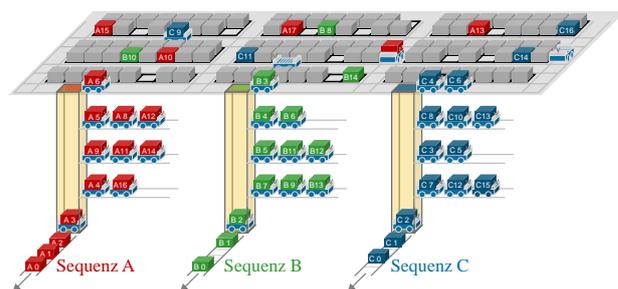


Abbildung 19. Jeder Lift bedient einen unabhängigen Sequenzstrom.

Für die Untersuchung betrachten wir ein Lager mit 24.000 Lagerplätzen, welche sich auf acht Ebenen und 15 Lagergassen verteilen. Auf jeder Ebene sind drei Quergassen eingezogen, jeweils an der Stirnseite sowie mittig im Lager. Drei Lifte verbinden die Ebenen mit der EA-Ebene.

Abbildung 20 zeigt einen Ausschnitt einer der acht Lagerebenen im Simulationsmodell. Die eingefärbten Knoten am unteren Rand stellen die Lifte dar. Die Shuttle-Fahrzeuge warten mit den auszulagernden Lagereinheiten in aufsteigender Sequenznummernreihenfolge vor den Liften. Fehlende Sequenznummern werden von anderen Ebenen ausgelagert.

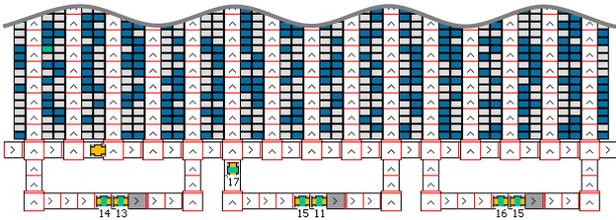


Abbildung 20. Ausschnitt einer Lagerebene im Simulationsmodell.

Die Shuttle-Fahrzeuge führen ausschließlich Doppelspiele aus. Nach Übergabe der ausgelagerten Lagereinheit auf der EA-Ebene wird eine einzulagernde Lagereinheit übergeben, sowie ein neuer Auslagerauftrag zugewiesen. Dabei handelt es sich um den ältesten Auslagerauftrag im System, welcher noch nicht zugewiesen wurde. Damit bedient jedes Shuttle-Fahrzeug jeden Lift und ist nicht etwa einem Lift fest zugeordnet. Für die Durchsatzanalyse variieren wir die Anzahl der Shuttle-Fahrzeuge von 2 bis 30 mit einer Schrittweite von 2 Shuttle-Fahrzeugen. Bei einem ersten Durchlauf erfolgt die Auslagerung chaotisch, bei einem zweiten Durchlauf hingegen ist die Auslagerung in einer fortlaufenden Sequenz erforderlich. Somit ergeben sich 30 Experimente, welche mit fünf Replikationen je Experiment und einer Simulationszeit von zehn Stunden durchgeführt werden.

5.2 ERGEBNISSE

Der erzielte Durchsatz in Abhängigkeit der Anzahl der eingesetzten Shuttle-Fahrzeuge ist der Abbildung 21 zu entnehmen.

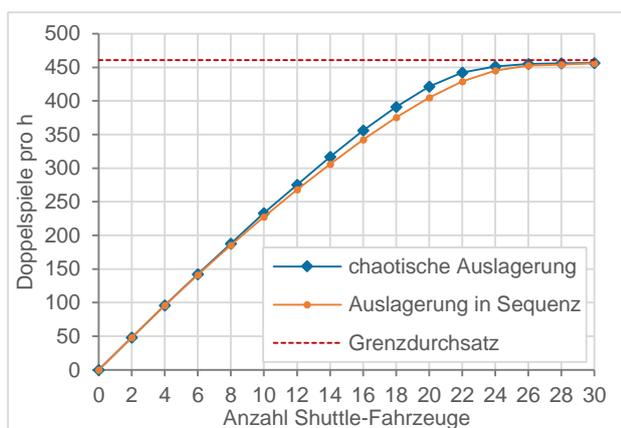


Abbildung 21. Erzielter Durchsatz in Abhängigkeit der Anzahl eingesetzter Shuttle-Fahrzeuge.

Mit zunehmender Anzahl an Shuttle-Fahrzeugen steigt der Durchsatz zunächst nahezu linear an. Die erzielte Durchsatzsteigerung je zusätzlich eingesetztem Fahrzeug nimmt bei steigender Anzahl an Shuttle-Fahrzeugen jedoch immer weiter ab. Bei ca. 26 Shuttle-Fahrzeugen wird der Grenzdurchsatz des Systems erreicht. Die Lifte arbeiten ununterbrochen, zusätzliche Fahrzeuge führen nicht mehr zu einer Steigerung des Durchsatzes.

Abbildung 22 veranschaulicht den Durchsatzverlust einer Auslagerung in Sequenz gegenüber einer chaotischen Auslagerung.

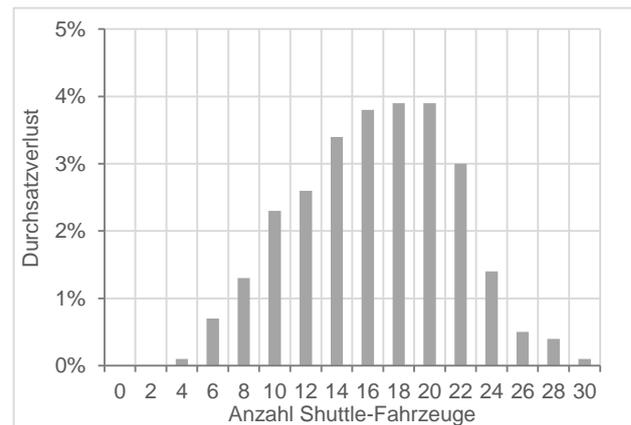


Abbildung 22. Durchsatzverlust, welcher durch die Auslagerung in Sequenz gegenüber einer chaotischen Auslagerung entsteht.

Bei einer geringen Anzahl an Shuttle-Fahrzeugen ist der Durchsatzverlust durch die Sequenzierung gering. Dieser nimmt mit steigender Anzahl an Shuttle-Fahrzeugen zu und erreicht bei 20 Shuttle-Fahrzeugen seinen Höhepunkt. Anschließend fällt der Durchsatzverlust wieder. Die Lifte stellen nun eindeutig den Engpass dar und die Sequenzierung auf den einzelnen Lagerebenen bleibt ohne Auswirkungen auf den Durchsatz, da die Fahrzeuge ohnehin auf die Lifte warten müssen. Der Durchsatzverlust setzt sich aus zwei Komponenten zusammen. Zum einen nehmen die Spielzeiten auf einer Lagerebene zu, falls spätere Zeitfenster beim Lift angefahren werden müssen. Zum anderen muss ein Lift immer auf die nächste Sequenznummer warten, auch wenn bereits weitere Shuttle-Fahrzeuge mit höheren Sequenznummern auf anderen Ebenen warten.

Anhand der Ergebnisse wird deutlich, dass sich durch die Auslagerung in Sequenz ein Durchsatzverlust in Abhängigkeit der Anzahl an Shuttle-Fahrzeuge einstellt. Dieser Durchsatzverlust fällt jedoch mit weniger als vier Prozent gegenüber einer chaotischen Auslagerung gering aus und lässt sich durch zusätzliche Shuttle-Fahrzeuge kompensieren. Durch das Anwenden der in diesem Beitrag beschriebenen Methode kann somit gezeigt werden, dass bei Shuttle-Systemen mit gassen- und ebenen-wechselnden Shuttle-Fahrzeugen auf eine aufwändige Sequenzierung in der Lagervorzone verzichtet werden kann.

6 ZUSAMMENFASSUNG UND AUSBLICK

Die Leistungsbestimmung von FTS ist eine Herausforderung und kann präzise nur durch die Durchführung einer Simulationsstudie erfolgen. Um den Aufwand dafür zu reduzieren, haben wir eine Methode vorgestellt, um FTS zu modellieren, damit diese in eine Simulationsumgebung überführt werden können. Darüber hinaus haben wir Steuerungsstrategien beschrieben, welche sich in das Simulationsmodell hinterlegen lassen, um ein ablauffähiges Modell zu erhalten. Dieses Modell kann anschließend für die individuelle Systemanalyse herangezogen werden.

Die beschriebenen Ansätze zur Modellierung sowie zur Steuerung von FTS werden in einen Bausteinkasten integriert, welcher zur Simulation unterschiedlichster FTS herangezogen werden kann. In diesen Bausteinkasten werden in einem nächsten Schritt Ausfallstrategien hinterlegt, welche einen robusten und effizienten Betrieb des FTS ermöglichen sollen, auch wenn einzelne FTF ausfallen. Damit lassen sich Verfügbarkeiten für die Fahrzeuge hinterlegen, um noch präzisere Aussagen über den Durchsatz und das Systemverhalten zu erhalten.

Darüber hinaus sollen einheitliche Auswertungsfunktionen integriert werden, wie beispielsweise ein Sankey-Diagramm, welches für die Engpassanalyse im Layout herangezogen werden kann.

Der beschriebene Ansatz zur Modellierung und Anwendung des zeitfensterbasierten Routings lässt sich auf kurvengängige Fahrzeuge übertragen, welche im Rahmen der Arbeit nicht berücksichtigt werden. Dafür müsste in Zeitfenstern, welche auf Knoten liegen, die eine Kurve darstellen, die Ausrichtung der FTF geändert werden.

Darüber hinaus kann eine weitere Ausrichtung entlang der z-Achse eingeführt werden, um Systeme mit dreidimensional verfahrenen Fahrzeugen, beziehungsweise Robotern zu simulieren.

LITERATUR

- [Gan-1998] Ganesharajah, T.; Hall, N. G.; Sriskandarajah, C.: Design and operational issues in AGV-served manufacturing systems. In: *Annals of Operations Research*, Jg. 76 (1998) Nr. 0, S. 109–154.
- [Kim-1991] Kim, C. W.; Tanchoco, J.: Conflict-free shortest-time bidirectional AGV routing. In: *International Journal of Production Research*, Jg. 29 (1991) Nr. 12, S. 2377–2391.

- [Lam-2016] Lamballais, T.; Roy, D., De Koster, M. B. M.: Estimating performance in a Robotic Mobile Fulfillment System. In: *European Journal of Operational Research*, Jg. 256 (2016) Nr. 3, S. 976–990.
- [Le--2006] Le-Anh, T.; De Koster, M. B. M.: A review of design and control of automated guided vehicle systems. In: *European Journal of Operational Research*, Jg. 171 (2006) Nr. 1, S. 1–23.
- [Lie-2017] Lienert, T.; Fottner, J.: No more Deadlocks - Applying the Time Window Routing Method to Shuttle Systems. In: *Proceedings of the 31st European Conference on Modelling and Simulation (2017)*
- [Liu-2001] Liu, F.-H.; Hung, P.-C.: Real-time deadlock-free control strategy for single multi-load automated guided vehicle on a job shop manufacturing system. In: *International Journal of Production Research*, Jg. 39 (2001) Nr. 7, S. 1323–1342.
- [Maz-2005] Maza, S.; Castagna, P.: A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles. In: *Computers in Industry*, Jg. 56 (2005), S. 719–733.
- [Mor-2009] ter Mors, A. W.: The world according to MARP – Multi-Agent Route Planning. Dissertation, Technische Universiteit Delft, Delft, 2009.
- [Sch-1989] Schmidt, F.: Komplexe Fahrerlose Transportsysteme – Fahrzeuganzahl, Investitionsaufwand, Wirtschaftlichkeit. Verlag TÜV Rheinland GmbH, Köln, 1989.
- [VDI-2510] Verein Deutscher Ingenieure: Fahrerlose Transportsysteme (FTS). VDI-Richtlinie Nr. 2510, 2005.
- [VDI-2710] Verein Deutscher Ingenieure: Einsatzgebiete der Simulation für Fahrerlose Transportfahrzeuge (FTS). VDI-Richtlinie Nr. 2710 Blatt 3, 2014.

[Vis-2006] Vis, I. F.: Survey of research in the design and control of automated guided vehicle systems. In: European Journal of Operational Research, Jg. 170 (2006) Nr. 3, S. 677–709.

Thomas Lienert, M.Sc., Research assistant at the Institute for Materials Handling, Material Flow, Logistics at the Technical University of Munich.

Prof. Dr.-Ing. Johannes Fottner, Professor and head of the Institute for Materials Handling, Material Flow, Logistics at the Technical University of Munich.

Address: Technische Universität München, Lehrstuhl fml,
Boltzmannstraße 15 85748 Garching, Germany,
Phone: +49 89 289 15939

E-Mail: lienert@fml.mw.tum.de